

湖南大学

HUNAN UNIVERSITY



毕业论文

含图结构的自适应

Lasso-COX 财务危机预

警研究

论文题目：

学生姓名：

张笑竹

学生学号：

201618070114

专业班级：

统计学 2016 级 01 班

学院名称：

金融与统计学院

指导老师：

王小燕

学院院长：

潘敏

2020 年 6 月 5 日

湖南大学

毕业论文原创性声明

本人郑重声明：所呈交的论文是本人在导师的指导下独立进行研究所取得的成果。除了文中特别加以标注引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写的成果作品。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律后果由本人承担。

学生签名：张策竹

日期：2020 年 6 月 5 日

毕业论文版权使用授权书

本毕业论文作者完全了解学校有关保留、使用论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权湖南大学可以将本论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本论文。

本论文属于

1、保 密 ☐，在_____年解密后适用本授权书。

2、不保密 ☒。

（请在以上相应方框内打“√”）

学 生 签名：张策竹

日期：2020 年 6 月 5 日

导 师 签名：[Signature]

日期：2020 年 6 月 5 日

含图结构的自适应 Lasso-COX 财务危机预警研究

摘 要

含图结构的自适应 Lasso 方法,是惩罚函数回归分析领域中的一个重要跟进性话题。本文利用图结构表示协变量之间的复杂网络关系,并将这一结构加入惩罚函数,与自适应 Lasso 方法相结合,同时实现变量的筛选和估计。尽管从“含图结构 Lasso 方法”到“含图结构自适应 Lasso 方法”的过渡,是惩罚函数的一个相对微小的结构变化,但是当新的方法应用于 Cox 比例风险模型时,这一变化极大地改善了估计参数的统计性质。具体而言,新方法的估计量具有 Oracle 性质;因此,当样本量较大时,这一方法的估计、预测性能更好,且它的显著性检验也相对方便。

针对这一新的方法,本文进行了全面而系统的研究,具体内容包括模型构建、求解算法(坐标下降法)、理论性质(估计量方差、组效应、渐进性质)、显著性检验、模拟和实证分析。其中,模拟部分将新方法与其他的方法(Lasso 和 Elastic Net)进行了对比,通过 5 个评价指标,在实践中说明了新方法的优良性质。

最后,实证分析部分将新方法应用于财务危机预警。建立的预警系统不仅能够挑选重要的财务指标,而且能够通过调节参数的动态变化,识别指标网络的中心点。利用筛选得到的指标体系,可以对各家上市公司在不同时点被 ST 的概率进行预测,从而监控财务风险,识别财务危机。

关键词: 财务预警; Cox 风险比例模型; 自适应 Lasso; 网络结构; Oracle 性质

On the Graph-incorporated Adaptive Lasso for Cox's Proportional Hazards Model with an Application to the Financial Crisis Warning System

Abstract

I propose a new procedure, the graph-incorporated adaptive Lasso for Cox's proportional hazards model, which is an important follow-up topic in regularized regression analysis. I incorporate a graph structure, which represents the complex networks among covariates, into the penalty function of adaptive Lasso. The transition from the non-adaptive graph-incorporated Lasso to the adaptive one, with its application to the Cox's model, though a minor change in structure, profoundly improves the statistical properties of estimators. Specifically, estimators of the proposed procedure enjoy the Oracle property, easier significance tests, and a better prediction performance when the sample size is large.

I elucidate the proposed procedure thoroughly and systematically by describing the penalized function, the coordinate descent algorithm, theoretical properties (variance of estimators, grouping effect, asymptotic properties) and significance tests. In the simulation study, I compare the proposed procedure with Lasso and Elastic Net, and demonstrate its excellent properties via five assessment indices.

In the empirical study, I establish a warning system of the financial crisis by the proposed procedure. The system can not only select significant financial indices but also identify the central point of the covariate networks by adjusting tuning parameters. The Cox's model constructed using chosen indices is able to estimate the probability of listed companies being "specially treated" as the time elapsed, so as to monitor financial risks and identify potential financial crisis.

Key words: financial crisis; Cox's proportional hazards model; adaptive Lasso; network structure; Oracle property

目 录

摘 要	I
Abstract	II
插图索引	V
附表索引	VI
一、绪论	1
(一) 研究背景	1
(二) 研究目的	1
(三) 研究现状	2
(四) 研究方法	3
(五) 研究内容	3
二、模型构建	4
(一) 风险比例模型	4
(二) 先验的图结构	5
1. 拉普拉斯矩阵	5
2. 图结构的构造	6
(三) 目标函数的构建	7
1. 惩罚函数	7
2. 目标函数	8
3. 惩罚函数的修正	8
三、坐标下降法	9
(一) 对数偏似然函数的泰勒展开	9
(二) 迭代公式	10
(三) 算法表述	11
(四) 解的路径和交叉检验	12
四、理论性质	13
(一) 估计量的方差	13
(二) 组效应	14
(三) 渐进性质	15
(四) 显著性检验	16
1. 全局检验	17
2. 局部检验	18
五、模拟	18
(一) 生成数据	19
(二) 指标测度	19

(三) 模拟结果	21
六、实证分析	23
(一) 数据设计与财务指标	23
(二) 估计结果	25
(三) 图结构分析	27
(四) 动态预测	31
(五) 结论与建议	32
七、讨论与展望	33
(一) 研究结论	33
(二) 不足和展望	34
参考文献	35
致谢	37
附录 A	38
(一) 定理 1 的证明	38
(二) 定理 2 的证明	39
(三) 定理 3 的证明	42
1. 稀疏性	42
2. 渐进正态性	42
附录 B	44
(一) 模拟代码	44
(二) 实证代码	51

插图索引

图 1	30 个财务指标的初始图结构	24
图 2	30 个财务指标的“阈值法”图结构	25
图 3	aGLasso 选择的子图结构	26
图 4	5 家企业的生存函数估计	32

附表索引

表 1	“阈值法”图结构的模拟结果	21
表 2	“协方差法”图结构的模拟结果	22
表 3	“相关系数法”图结构的模拟结果	22
表 4	财务指标	24
表 5	交叉检验结果	25
表 6	aGLasso 过程的估计量	27
表 7	不同组合的调整参数下图结构的变化.....	29
表 8	基准生存函数估计量.....	31

一、绪论

以 Lasso、SCAD、Elastic Net 等为代表的惩罚方法是二十年来回归分析和变量选择领域的革命性成果。结合惩罚方法的优良特点,本文额外考虑具有分组特性的协变量之间的图结构。此外,由于 Lasso 方法的估计量不具有 Oracle 性质,因此本文借助自适应 Lasso 方法的思想,对 L_1 范数惩罚部分进行权重修正,从而构建包含图结构的自适应 Lasso 方法。本文将这一方法应用于 Cox 风险比例模型 (Cox's Proportional Hazards Model),并从估计量的求解算法、理论性质、模拟比较等各个方面,进行进一步的研究。最后,利用自适应 Lasso-Cox 方法建立财务危机预警模型,进行实证分析。

(一) 研究背景

有效地识别财务危机前兆是上市公司应对风险的关键之一。2008 年全球金融危机蔓延,使得全球众多企业陷入资金链断裂的困境。2015 年中国股市大幅度异常波动,使得国内多家上市公司因资金周转困难,面临严重的财务危机。2019 年中美贸易战打响,众多从事外贸行业的、抗风险能力较弱的中小企业出现亏损,进退两难。2020 年开年,受到 COVID-19 疫情影响,中国几乎全面停工停产,美股十天四次熔断,世界迎来新一轮的金融动荡。因此,建立有效的财务危机预警模型,能够帮助企业提前识别自身的潜在风险,在猝不及防的外部危机到来之前,及时采取内部措施,加强危机应对能力。此外,债权人、投资者、监管者也可以通过该模型了解公司面临的风险大小,从而减少自身损失、保护自身利益、加强市场监管。财务危机预警模型对于维护市场稳定有着重要的意义。

(二) 研究目的

自 20 世纪以来有关财务危机预警方法的研究蓬勃发展。传统的二分类方法包括一元线性判别模型、多元线性判别模型、Logistic 模型等;但是他们都或多或少地存在模型包含信息不足、对协变量分布要求严格等问题,且不能根据时间变化对财务危机进行动态预测。BP 神经网络、支持向量机等方法实现了预测性能的显著提升,却仍然是一种静态方法。然而, Cox 模型对协变量的分布无特殊要求,且允许存在删失数据;它不仅能够给出对某一时点公司财务情形的判断,而且能够动态展示上市公司的财务状况,是非常强大的分析工具。

不同于一般的协变量集合,上市公司的财务指标通常以组出现,组内相关性较强,而组间相关性较弱。本文对一个好的财务危机预警模型提出如下要求:①能够对上市公

司财务指标之间的相关性和结构进行分析；②能够对财务指标进行筛选，构建具有代表性的协变量集合；③能够根据上市公司的财务状况，利用筛选得到的财务指标进行预测，估计每一个公司的生存函数；④预测结果的性质较好，精度较高。

本研究的目的是，就是以建立一个满足上述四条要求的财务危机预警模型为最终落脚点，推广在生存分析中的、具有图结构的、广义线性回归的惩罚方法，研究它的性质、求解算法，并将它与其他方法进行比较；从而给出该方法的一个系统、全面的阐述。

（三）研究现状

在生存分析中，研究生存时间与协变量关系的最经典方法是 Cox (1972) 的风险比例模型。为了解决高维 Cox 模型的求解、推断问题，二十年来一系列正则化方法迅猛发展。Tibshirani (1996) 率先提出应用于线性回归的 Lasso (Least Absolute Shrinkage and Selection Operator) 方法，这一方法能够同时实现回归变量的筛选和参数估计。随后，Tibshirani (1997) 将 Lasso 方法推广至 Cox 模型，并对估计算法和估计量方差进行了简述。然而，Fan 和 Li (2001) 认为 Lasso 估计量存在统计性质上的缺陷，因而另辟蹊径，提出了 SCAD (Smoothly Clipped Absolute Deviation) 方法，并且证明了它估计量的 Oracle 性质。此后，Fan 和 Li (2002) 将 SCAD 方法的应用范围进一步推广至 Cox 模型。自 Fan 和 Li 提出 SCAD 方法以来，正则化领域逐渐形成了一个统一的评价框架，对惩罚函数提出了无偏性、稀疏性和连续性的要求，并且以是否满足 Oracle 性质作为评价估计量的重要标准之一。

此后，在这一框架之内，出现了大量的针对不同惩罚函数及其估计量的研究。Zou (2006) 通过构造反例，证明 Lasso 估计量的不一致性；与此同时，通过对 Lasso 方法进行惩罚项权重的修正，提出自适应 Lasso 方法，并证明其估计量具有 Oracle 性质。随后，Zhang 和 Lu (2007) 将自适应 Lasso 方法推广至 Cox 模型，且这一推广依然保持了估计量的 Oracle 性质；此外，文献也对估计量的求解算法和估计量方差进行了说明。这些方法都能够得到更加稀疏的模型，这对于获得良好的预测性能、提高模型的解释性是至关重要的。

尽管如此，对于存在先验图结构的协变量集合，上述方法往往会忽视这种结构的影响。Zou 和 Hastie (2005) 提出了 Elastic Net 方法，将 Lasso 回归和岭回归相结合，并由 Wu 和 Lange (2008) 将其推广至高维 Cox 回归；这种方法能够考虑一些分组效应，却仍然没有利用任何先验的图结构。为此，Li 和 Li (2008, 2010) 考虑了网络结构限制的惩罚方法，利用拉普拉斯惩罚表示网络结构。随后，Sun 等 (2014) 则将这种方法从线性形式推广至 Cox 模型，却只能得到以上下界为限制的弱 Oracle 性质。此后，随着正则化研究的前沿突围至 Oracle 不等式以及估计量显著性检验的方向，针对（应用于 Cox 模型的）图结构这一特定形式的惩罚函数，相关文献就相对较少了。因此，寻找包含图结构的、其估计量具有 Oracle 性质的惩罚函数，仍然是一个值得探讨的话题。值得

注意的是,构造这样的惩罚函数,与估计量显著性检验等前沿领域并不矛盾,因为在估计量服从渐进正态分布的前提下,检验统计量的构造并不困难。

由于 Lasso 方法中 L_1 范数惩罚部分的不可导性,不能采取直接求导的方法估计系数。目前常用的方法是坐标下降法 (Coordinate Descend Algorithm)。Fu (1998) 对牛顿-拉弗森算法进行修正,提出针对 Lasso 的“射击”算法 (Shooting Algorithm), Friedman 等 (2007) 则在此基础上正式提出坐标下降法。随后, Friedman 等 (2010) 将针对线性回归的坐标下降法推广至广义线性模型,而 Simon 等 (2011) 则更进一步,将坐标下降法推广至 Cox 模型。

综上所述,在正则化方法发展的历程中,不同的惩罚函数总是先与最简单的线性模型相结合,以便简化问题的复杂性,突出惩罚函数的本质特征。随后,伴随着人们对某一惩罚函数的认识不断深入,它开始与广义线性模型、Cox 模型相结合。面对不具有 Oracle 性质的 Lasso 估计量,除了另辟蹊径 (例如 SCAD 方法),将 Lasso 过程修正为自适应 Lasso 过程显然是常见的思路。现阶段,包含先验图结构的 Lasso 惩罚方法的研究已经相对成熟。然而,在自适应 Lasso 的启发下,进一步地改进和研究仍有必要,从而为正则化家族发展出性质更加优良的方法。

(四) 研究方法

本文从 Lasso、自适应 Lasso、图结构这三个方面入手,采用理论、模拟和实证相结合的方法进行分析。在理论研究部分,参考已有的文献,进行模型构建和推导。在模拟研究部分,编写 R 语言代码生成随机数据,并执行各种惩罚方法的估计和推断过程。在实证研究部分,利用万德数据库获取上市公司的数据,利用 R 语言进行数据清洗和整理,执行本文所探讨的方法,并借助 Gephi 软件分析图结构,得到最终结论。

(五) 研究内容

为了达到上述研究目的,本文将研究内容安排如下:第一章是绪论部分,这部分主要介绍本研究的研究背景、研究目的、研究现状和研究方法;第二章是模型构建部分,这部分引入针对 Cox 比例风险模型的、含图结构的自适应 Lasso 方法,给出模型的具体形式;第三章是求解算法部分,这部分给出坐标下降算法的推导过程和最终形式;第四章是理论性质分析部分,这部分对估计量的方差进行估计,对网络结构的组效应进行论证,对估计量的 Oracle 性质进行说明,并给出估计量显著性检验的方法;第五章是模拟比较部分,通过随机生成的模拟数据,对比多种情况下模型性能的差异;第六章是实证分析,将模型应用于上市企业的财务危机预警,并给出相应的分析和结论。第七章给出简单的讨论和展望。附录中包括部分定理的证明,以及模拟和实证研究的代码。

二、模型构建

在这一章中，拟构建一个应用于 Cox 风险比例模型的惩罚方法，这一方法选取了正则化家族中最为经典的 Lasso 方法作为基本框架。在此基础上，进行图结构方面和自适应方面的拓展，最终给出模型的具体形式。

（一）风险比例模型

首先，介绍一些符号。定义 T 为失效时间（failure time）， C 为删失时间（censoring time）， $\tilde{T} = \min\{T, C\}$ 为删失失效时间（censored failure time）。定义 $\Delta = I(T \leq C)$ 为失效指标，其中 $I(\cdot)$ 为示性函数。定义

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix} = \begin{pmatrix} \mathbf{X}_{(1)}^T \\ \mathbf{X}_{(2)}^T \\ \vdots \\ \mathbf{X}_{(n)}^T \end{pmatrix} = (\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_p)$$

为协变量矩阵， T 和 C 相互条件独立于 \mathbf{X} ，且删失机制是无信息性的。观测值包含三个部分，为 $(\tilde{T}_i, \Delta_i, \mathbf{X}_{(i)})$ ， $i = 1, 2, \dots, n$ ，且观测值之间相互独立。

根据 Cox (1972)，Cox 风险比例模型的基本形式为：

$$\lambda_i(t|\mathbf{X}_{(i)}) = \lambda_0(t) \exp(\boldsymbol{\beta}^T \mathbf{X}_{(i)}) \quad (1)$$

其中， $\lambda_i(t|\mathbf{X}_{(i)})$ 为个体 i 的风险函数。 $\lambda_0(t)$ 为基准风险函数，其具体形式无需指定。 $\boldsymbol{\beta} = (\beta_1, \beta_2, \dots, \beta_p)^T$ 是未知的回归系数向量。为了估计 $\boldsymbol{\beta}$ ，考虑偏似然函数

$$L(\boldsymbol{\beta}) = \prod_{i=1}^n \left\{ \frac{\exp(\boldsymbol{\beta}^T \mathbf{X}_{(i)})}{\sum_{j=1}^n I(\tilde{T}_j \geq \tilde{T}_i) \exp(\boldsymbol{\beta}^T \mathbf{X}_{(j)})} \right\}^{\Delta_i}.$$

注意到，偏似然函数 $L(\boldsymbol{\beta})$ 中已经不包括 $\lambda_0(t)$ 。为了求解的方便，考虑对数偏似然函数

$$\ell(\boldsymbol{\beta}) = \sum_{i=1}^n \Delta_i \left[\boldsymbol{\beta}^T \mathbf{X}_{(i)} - \log \left\{ \sum_{j=1}^n I(\tilde{T}_j \geq \tilde{T}_i) \exp(\boldsymbol{\beta}^T \mathbf{X}_{(j)}) \right\} \right]. \quad (2)$$

式(2)的极大值点是所求 $\boldsymbol{\beta}$ 的一个估计量 $\hat{\boldsymbol{\beta}}$ ，尽管它与利用惩罚方法求得的估计量不同。

作为目标函数的一部分，式(2)的推导过程在以上论述中得以体现。此外，有关 Cox 风险比例模型，还需要说明以下几点：

(1) Cox 风险比例模型不仅包含失效观测的信息，而且包含删失观测的信息。在偏似然函数 $L(\boldsymbol{\beta})$ 中，每一项的分子代表了某个失效观测的协变量线性组合指数幂，而每一项的分母代表了所有在险观测的协变量线性组合指数幂之和；其中，在险观测既包括在险的失效观测，也包括在险的删失观测。因此，Cox 风险比例模型不会因为删失观测

的实际失效时间未知就将其忽略，从而能够充分利用所有观测的信息。

(2) Cox 风险比例模型是半参数模型。与其他对变量分布有严格要求的判别方法不同，Cox 模型对各个变量的分布无明确要求。具体而言，在估计风险函数时，基准风险函数 $\lambda_0(t)$ 无需服从某个已知分布，这是由于，对于两个协变量分别为 \mathbf{Z} 和 \mathbf{Z}^* 的个体而言，它们风险函数的比率为

$$\frac{\lambda_1(t|\mathbf{Z})}{\lambda_2(t|\mathbf{Z}^*)} = \frac{\lambda_0(t) \exp(\beta^T \mathbf{Z})}{\lambda_0(t) \exp(\beta^T \mathbf{Z}^*)} = \exp[\beta^T (\mathbf{Z} - \mathbf{Z}^*)].$$

其中，基准风险函数 $\lambda_0(t)$ 被消掉，得到的结果是一个常数；而我们关心的往往是这样的比率，而非 $\lambda_0(t)$ 本身。

以上说明了 Cox 模型的非参数性质，然而 Cox 模型也具有参数性质，它体现在以参数 β 为系数的协变量线性组合中；正因如此，它被视作是半参数的。事实上，后续的分析是完全围绕参数 β 的估计展开的。此外，需要说明的是线性组合 $\beta^T \mathbf{Z}$ 中并不包含截距项；即便 $\exp(\beta_0)$ 存在，也会被吸收进入 $\lambda_0(t)$ 。

(3) Cox 风险比例模型是动态模型。从式(1)可以看出，风险函数与时间 t 有关，随着时间的变化，基准风险和不同个体的风险都在动态发展。尽管在估计参数时无需关注 $\lambda_0(t)$ ，但是最终可以利用估计量 $\hat{\beta}$ 来估计 $\lambda_0(t)$ 和生存函数 $S_0(t)$ ，从而实现不同时点上对个体的风险预测。

(二) 先验的图结构

1. 拉普拉斯矩阵

图结构反映协变量之间的内在联系和分组。对于能够利用图结构度量的协变量集合，变量之间是高度相关的。施加先验的马尔科夫随机场 (Markov Random Field)，能够解释这种回归系数的依赖关系，并使得回归系数更加平滑。MRF 将 β 的先验联合分布，根据图结构分解为较低维度的分布。高斯 MRF 模型假定 β 的联合分布为

$$f(\beta) \propto \exp \left\{ -\frac{1}{2\sigma^2} \beta^T \mathbf{L} \beta \right\}.$$

受这一假定的启发，Li 和 Li (2008) 引入了基于网络结构约束的变量选择方法。

考虑图结构 $G = (V, E, W)$ ，其中， $V = \{1, 2, \dots, p\}$ 是代表 p 个协变量的点集合； $E = \{j \sim k\}$ 是图结构的边集合，对应着连接变量 \mathbf{X}_j 和 \mathbf{X}_k 的边； $W = \{w(j, k)\}$ 是图结构的权重集合，对应着边 $(j \sim k)$ 的权重。在实际应用中，边的权重往往用来测量顶点（变量）之间的不确定性，且令 $w(j, j) = 0$ （图结构中不包括自环）。此外，考虑顶点 j 的度

$$d_j = \sum_{k=1}^p w_{jk},$$

当 $d_j = 0$ 时 j 为孤立点。

根据 Chung (1997)，下面，定义图结构 G 的拉普拉斯矩阵 $\mathbf{L} = (l_{jk})$

$$l_{jk} = \begin{cases} 1, & \text{当 } j = k \text{ 且 } d_j \neq 0, \\ -w(j, k) / \sqrt{d_j d_k}, & \text{当 } j \text{ 和 } k \text{ 相邻}, \\ 0, & \text{其他情况}. \end{cases}$$

注意到, 矩阵 \mathbf{L} 是对称的, 因而可以写作 $\mathbf{L} = \mathbf{S}\mathbf{S}^T$; 其中, $\mathbf{S}_{p \times m}$ 的每一行代表顶点(协变量), 而每一列代表边($j \sim k$); 当边($j \sim k$)对应的行为协变量 j 时, 相应的元素 $\mathbf{S}_{j(j \sim k)} = \sqrt{w(j, k)/d_j}$, 当边($j \sim k$)对应的行为协变量 k 时, 相应的元素 $\mathbf{S}_{j(j \sim k)} = -\sqrt{w(j, k)/d_k}$, 其他情况下对应的元素为 0. 根据代数计算, 可以得到

$$\beta^T \mathbf{L} \beta = \sum_{j=1}^p \sum_{k=1}^p \frac{w_{jk}}{2} \left(\frac{\beta_j}{\sqrt{d_j}} - \frac{\beta_k}{\sqrt{d_k}} \right)^2.$$

此外, Chung (1997) 还指出, \mathbf{L} 的特征值能够反映许多图结构 G 的性质。

2. 图结构的构造

注意到, 在实际应用中, 尽管某些情况下存在既定的网络结构(如 Li 和 Li (2010) 中的基因表达分子模组), 但是在更多的情境下, 需要根据数据构造先验的图结构 G , 并利用邻接矩阵($w(j, k)$)表示。由于权重代表顶点(协变量)之间的联系程度, 所以应当从能够反映变量相关性的协方差或者相关系数出发, 考虑图结构的构造方法。

根据 Huang (2011), 本文考虑以下三种先验网络结构的构造方法。

(1) 阈值法

首先, 计算协变量 \mathbf{X} 的相关系数矩阵 $\text{Cor}(\mathbf{X}) = (r_{jk})$ 。然后, 为了寻找 r_{jk} 的阈值, 考虑费雪变换

$$z_{jk} = 0.5 \log \frac{1 + r_{jk}}{1 - r_{jk}},$$

若 $r_{jk} = 0$, 则 $\sqrt{n-3}r_{jk}$ 大致服从于标准正态分布。若对 $\sqrt{n-3}r_{jk}$ 设置阈值 c , 则 r_{jk} 对应的阈值为

$$r = \frac{\exp\left(\frac{2c}{\sqrt{n-3}}\right) - 1}{\exp\left(\frac{2c}{\sqrt{n-3}}\right) + 1}.$$

阈值 c 的确定, 可以利用 p-值的思想, 即若记取定的 p-值为 pv , 则 $c = \Phi^{-1}(1 - pv)$ 。

从而得到邻接矩阵为:

$$w(j, k) = \begin{cases} \text{sgn}(r_{jk}) \cdot I(|r_{jk}| > r), & j \neq k \\ 0, & j = k \end{cases}.$$

(2) 协方差法

利用协变量 \mathbf{X} 的协方差矩阵 $\text{Cov}(\mathbf{X}) = (\sigma_{jk})$ 反映图结构, 临界矩阵为:

$$w(j, k) = \begin{cases} \sigma_{jk}^3, & j \neq k \\ 0, & j = k \end{cases}.$$

(3) 相关系数法

与(1)的出发点类似, 计算协变量 \mathbf{X} 的相关系数矩阵 $\text{Cor}(\mathbf{X}) = (r_{jk})$ 。然后, 可以直接得到邻接矩阵:

$$w(j, k) = \begin{cases} \max\{0, r_{jk}\}, & j \neq k \\ 0, & j = k \end{cases}$$

对比三种方法, 由于“阈值法”得到的图结构舍弃了相关性过弱的边, 因此这种图结构最为简洁, 在可视化分析中也更为方便; 而“协方差法”和“相关系数法”得到的图结构, 包含除自环外几乎所有顶点(协变量)之间的边, 显得有些冗余。然而, 上述三种方法在性质上无本质差异, 尽管在实际应用中, 可能存在着预测性能上的细微区别。在第五章中, 将通过数据模拟, 对这三种方法进行性能的对比, 并挑选一个用于第六章的实证分析。

(三) 目标函数的构建

1. 惩罚函数

为了最大化对数偏似然函数(2), 在考虑协变量的图结构的情况下, 同时实现变量的筛选和估计, 使得估计量具有优良的性质, 现构造惩罚函数

$$\begin{aligned} p(\boldsymbol{\beta}; \lambda_1, \lambda_2) &= \lambda_1 \sum_{j=1}^p \tau_j |\beta_j| + \lambda_2 \boldsymbol{\beta}^T \mathbf{L} \boldsymbol{\beta} \\ &= \lambda_1 \sum_{j=1}^p \frac{|\beta_j|}{|\tilde{\beta}_j|} + \lambda_2 \sum_{j=1}^p \sum_{k=1}^p \frac{w_{jk}}{2} \left(\frac{\beta_j}{\sqrt{d_j}} - \frac{\beta_k}{\sqrt{d_k}} \right)^2. \end{aligned} \quad (3)$$

其中, $\lambda_1, \lambda_2 \geq 0$ 。 $\tau_j = 1/|\tilde{\beta}_j|$ 为 L_1 范数项的权重, 根据数据自适应地选取, 体现了自适应 Lasso 的特点。 $\tilde{\boldsymbol{\beta}} = (\tilde{\beta}_1, \dots, \tilde{\beta}_p)^T$ 是对数偏似然函数(2)的极大值点, 并且是一致估计量 (Andersen 和 Gill, 1982)。其中的每一个分量反映了对应协变量的相对重要性: 较大的 $|\tilde{\beta}_j|$ 赋予较小的惩罚, 最终选择较大的系数; 反之, 较小的 $|\tilde{\beta}_j|$ 赋予较大的惩罚, 最终选择较小的系数。当然, 并非一定要选择对数偏似然函数(2)的极大值点作为 $\tilde{\boldsymbol{\beta}}$, 任何一个 $\boldsymbol{\beta}$ 的一致估计量均可胜任。

选择自适应 Lasso 作为惩罚函数的一部分, 是由于其优良的性质。与 Lasso 相比, 自适应 Lasso 估计量具有 Oracle 性质, 即具有稀疏性和渐进正态性 (详见第四章); 而与 SCAD 相比, 自适应 Lasso 的惩罚函数是凸函数, 使得其优化方便高效, 且保证全局最优解的存在性。

注意到, L_2 范数项惩罚了图结构中相邻变量的“标准化”系数之差的平方和, 以提升局部的平滑性, 使得相关的变量更有可能被同时选择, 且估计得到的系数相近。所谓“标准化”, 指的是以度的平方根 $\sqrt{d_j}$ 作为对应系数 β_j 的分母, 从而使得在网络中拥有更多关

联的变量获得更大的系数。此外，当拉普拉斯矩阵 \mathbf{L} 为单位矩阵 \mathbf{I} 时（ \mathbf{L} 与 \mathbf{I} 的区别在于对角线外元素的不同），（3）就退化为自适应 Elastic Net 的惩罚函数，即自适应 Lasso 和岭回归的线性组合。

2. 目标函数

为了筛选和估计重要的变量，利用惩罚函数（3），可以得到最大化对数偏似然函数（2）的目标函数

$$R(\beta) = -\frac{1}{n} \ell(\beta) + \lambda_1 \sum_{j=1}^p \tau_j |\beta_j| + \lambda_2 \beta^T \mathbf{L} \beta. \quad (4)$$

从而 β 的估计量为

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} R(\beta).$$

在第三、四章中，为了计算的方便，有时会对参数 $(\lambda_1, \lambda_2)^T$ 进行变换。令 $\lambda = \lambda_1 + 2\lambda_2$ ， $\alpha = \lambda_1/(\lambda_1 + 2\lambda_2)$ ，则（4）可写为

$$R(\beta) = -\frac{1}{n} \ell(\beta) + \lambda \left(\alpha \sum_{j=1}^p \tau_j |\beta_j| + \frac{(1-\alpha)}{2} \beta^T \mathbf{L} \beta \right). \quad (5)$$

其中， $\lambda \geq 0, \alpha \in [0,1]$. 对于任意的 $\alpha \in [0,1]$ ，惩罚函数（5）都是严格凸函数，保留了稀疏性和平滑性的双重优良性质。在处理调节参数时，可以先固定 α 的取值，然后单独考虑 λ 的变化。

3. 惩罚函数的修正

尽管希望网络中相邻变量的系数相近，以实现平滑性，但这并不意味着相邻的系数符号相同。Li 和 Li（2007）指出，与 Lasso 或 Elastic Net 的惩罚函数不同，GLasso 的惩罚函数不是关于坐标轴对称的，因此对于符号不同的参数，惩罚函数也不同。Li 和 Li（2010）进一步指出，当网络结构中的相邻变量拥有符号相反的回归系数时，惩罚函数（3）表现不佳。所以，在这种情况下，探讨的系数相近应当是系数绝对值的相近。为此，可以对惩罚函数（3）进行修正：

$$\begin{aligned} p^*(\beta; \lambda_1, \lambda_2) &= \lambda_1 \sum_{j=1}^p \tau_j |\beta_j| + \lambda_2 \beta^T \tilde{\mathbf{L}} \beta \\ &= \lambda_1 \sum_{j=1}^p \frac{|\beta_j|}{|\tilde{\beta}_j|} + \lambda_2 \sum_{j=1}^p \sum_{k=1}^p \frac{w_{jk}}{2} \left(\frac{\operatorname{sgn}(\tilde{\beta}_j) \beta_j}{\sqrt{d_j}} - \frac{\operatorname{sgn}(\tilde{\beta}_k) \beta_k}{\sqrt{d_k}} \right)^2. \end{aligned} \quad (6)$$

其中 $\tilde{\mathbf{L}} = \mathbf{A}^T \mathbf{L} \mathbf{A}$ ， $\mathbf{A} = \operatorname{diag}(\operatorname{sgn}(\tilde{\beta}_1), \dots, \operatorname{sgn}(\tilde{\beta}_p))$ 。而 $\tilde{\beta} = (\tilde{\beta}_1, \dots, \tilde{\beta}_p)^T$ 可以使用与构造 τ_j 时相同的系数向量。

为了方便地区分（3）和（6），不妨将由惩罚函数（3）得到的方法称作 GLasso(Graphic

Lasso), 而将惩罚函数 (6) 得到的方法称作 aGLasso (Adjusted Graphic Lasso). 第三章和第四章的推导以 GLasso 为主, 而 aGLasso 的对应结论只需要在此基础上进行简单的修正. 第五章的模拟将比较 GLasso 和 aGLasso 两种方法的性能. 第六章的实证分析选择其中的一种方法进行.

三、坐标下降法

作为某种程度上的广义 Elastic Net 方法, GLasso 方法的目标函数是严格凸函数, 因此其全局最优解一定存在, 且它的极值点就是它的最值点. 坐标下降法是求解正则化家族模型的经典算法, 十分适合解决 GLasso 这类问题. 它的特点是每次迭代只更新一个维度的估计量, 而最终仍然能够收敛到真值. 坐标下降法以牛顿-拉弗森算法为框架, 但是更改了最小二乘估计这一步骤. 下面, 首先对式 (2) 进行泰勒展开, 接着推导每步迭代公式, 然后给出算法的完整表述. 最后, 对 GLasso 的路径解和交叉检验进行讨论.

(一) 对数偏似然函数的泰勒展开

将对数偏似然函数 (2) 进行二阶泰勒展开, 得

$$\begin{aligned}\ell(\boldsymbol{\beta}) &\approx \ell(\tilde{\boldsymbol{\beta}}) + (\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}})^T \nabla \ell(\tilde{\boldsymbol{\beta}}) + \frac{1}{2} (\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}})^T \nabla^2 \ell(\tilde{\boldsymbol{\beta}}) (\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}) \\ &= \ell(\tilde{\boldsymbol{\beta}}) + (\mathbf{X}\boldsymbol{\beta} - \tilde{\boldsymbol{\eta}})^T \mathbf{u}(\tilde{\boldsymbol{\eta}}) + \frac{1}{2} (\mathbf{X}\boldsymbol{\beta} - \tilde{\boldsymbol{\eta}})^T \mathbf{A}(\tilde{\boldsymbol{\eta}}) (\mathbf{X}\boldsymbol{\beta} - \tilde{\boldsymbol{\eta}}).\end{aligned}$$

其中, $\tilde{\boldsymbol{\eta}} = \mathbf{X}\tilde{\boldsymbol{\beta}}$, $\mathbf{u}(\tilde{\boldsymbol{\eta}}) = \partial \ell / \partial \boldsymbol{\eta} |_{\boldsymbol{\eta}=\tilde{\boldsymbol{\eta}}}$, $\mathbf{A}(\tilde{\boldsymbol{\eta}}) = \partial^2 \ell / \partial \boldsymbol{\eta} \boldsymbol{\eta}^T |_{\boldsymbol{\eta}=\tilde{\boldsymbol{\eta}}}$.

更具体地,

$$\begin{aligned}\mathbf{u}(\tilde{\boldsymbol{\eta}}) &= \begin{pmatrix} \left. \frac{\partial \ell}{\partial \eta_1} \right|_{\eta_1=\tilde{\eta}_1} \\ \left. \frac{\partial \ell}{\partial \eta_2} \right|_{\eta_2=\tilde{\eta}_2} \\ \vdots \\ \left. \frac{\partial \ell}{\partial \eta_n} \right|_{\eta_n=\tilde{\eta}_n} \end{pmatrix} = \begin{pmatrix} \Delta_1 - \sum_{i=1}^n \Delta_i \frac{I(\tilde{T}_1 \geq \tilde{T}_i) \exp(\tilde{\eta}_1)}{\sum_{j=1}^n I(\tilde{T}_j \geq \tilde{T}_i) \exp(\tilde{\eta}_j)} \\ \Delta_2 - \sum_{i=1}^n \Delta_i \frac{I(\tilde{T}_2 \geq \tilde{T}_i) \exp(\tilde{\eta}_2)}{\sum_{j=1}^n I(\tilde{T}_j \geq \tilde{T}_i) \exp(\tilde{\eta}_j)} \\ \vdots \\ \Delta_n - \sum_{i=1}^n \Delta_i \frac{I(\tilde{T}_n \geq \tilde{T}_i) \exp(\tilde{\eta}_n)}{\sum_{j=1}^n I(\tilde{T}_j \geq \tilde{T}_i) \exp(\tilde{\eta}_j)} \end{pmatrix}, \\ \mathbf{A}(\tilde{\boldsymbol{\eta}})_{kk} &= \left. \frac{\partial^2 \ell}{\partial \eta_k \partial \eta_k} \right|_{\eta_k=\tilde{\eta}_k} \\ &= - \sum_{i=1}^n \Delta_i \frac{I(\tilde{T}_k \geq \tilde{T}_i) \exp(\tilde{\eta}_k)}{\sum_{j=1}^n I(\tilde{T}_j \geq \tilde{T}_i) \exp(\tilde{\eta}_j)} + \sum_{i=1}^n \Delta_i \frac{I(\tilde{T}_k \geq \tilde{T}_i) \exp(2\tilde{\eta}_k)}{\left[\sum_{j=1}^n I(\tilde{T}_j \geq \tilde{T}_i) \exp(\tilde{\eta}_j) \right]^2},\end{aligned}$$

$$\begin{aligned} \mathbf{A}(\tilde{\boldsymbol{\eta}})_{kk'} &= \frac{\partial^2 \ell}{\partial \eta_k \eta_{k'}^T} \bigg|_{\eta_k = \tilde{\eta}_k, \eta_{k'} = \tilde{\eta}_{k'}} \\ &= \sum_{i=1}^n \Delta_i \frac{I(\tilde{T}_k \geq \tilde{T}_i) I(\tilde{T}_{k'} \geq \tilde{T}_i) \exp(\tilde{\eta}_k) \exp(\tilde{\eta}_{k'})}{\left[\sum_{j=1}^n I(\tilde{T}_j \geq \tilde{T}_i) \exp(\tilde{\eta}_j) \right]^2}. \end{aligned}$$

根据上式,可以看出 $\mathbf{A}(\tilde{\boldsymbol{\eta}})$ 是对称矩阵。令 $\mathbf{z}(\tilde{\boldsymbol{\eta}}) = \tilde{\boldsymbol{\eta}} - \mathbf{A}^{-1}(\tilde{\boldsymbol{\eta}})\mathbf{u}(\tilde{\boldsymbol{\eta}})$ 。利用 $\mathbf{A} = \mathbf{A}^T$ 的性质,经过代数运算,可以得到

$$\ell(\boldsymbol{\beta}) \approx \frac{1}{2}(\mathbf{z}(\tilde{\boldsymbol{\eta}}) - \mathbf{X}\boldsymbol{\beta})^T \mathbf{A}(\tilde{\boldsymbol{\eta}})(\mathbf{z}(\tilde{\boldsymbol{\eta}}) - \mathbf{X}\boldsymbol{\beta}) + \ell(\tilde{\boldsymbol{\beta}}) - \frac{1}{2}\mathbf{u}(\tilde{\boldsymbol{\eta}})\mathbf{A}^{-1}(\tilde{\boldsymbol{\eta}})\mathbf{u}(\tilde{\boldsymbol{\eta}}).$$

由于 $\ell(\tilde{\boldsymbol{\beta}}) - \frac{1}{2}\mathbf{u}(\tilde{\boldsymbol{\eta}})\mathbf{A}^{-1}(\tilde{\boldsymbol{\eta}})\mathbf{u}(\tilde{\boldsymbol{\eta}})$ 与变量 $\boldsymbol{\beta}$ 无关,因此将它们丢弃。进一步近似,得到

$$\ell(\boldsymbol{\beta}) \approx \frac{1}{2} \sum_{i=1}^n a(\tilde{\boldsymbol{\eta}})_i (z(\tilde{\boldsymbol{\eta}})_i - \boldsymbol{\beta}^T \mathbf{X}_{(i)})^2$$

其中, $z(\tilde{\boldsymbol{\eta}})_i$ 为 $\mathbf{z}(\tilde{\boldsymbol{\eta}})$ 的第 i 个元素, $a(\tilde{\boldsymbol{\eta}})_i$ 为 $\mathbf{A}(\tilde{\boldsymbol{\eta}})$ 第 i 行、第 i 列的元素。根据 Hastie 和 Tibshirani (1990), 由于 $\mathbf{A}(\tilde{\boldsymbol{\eta}})$ 的非对角线元素(成数量级地)远远小于对角线元素,因此可以用 $\mathbf{D}(\tilde{\boldsymbol{\eta}}) = \text{diag}\{a(\tilde{\boldsymbol{\eta}})_1, \dots, a(\tilde{\boldsymbol{\eta}})_n\}$ 来代替 $\mathbf{A}(\tilde{\boldsymbol{\eta}})$ 。最终,得到近似目标函数:

$$\begin{aligned} R(\boldsymbol{\beta}) \approx & -\frac{1}{2n} \sum_{i=1}^n a(\tilde{\boldsymbol{\eta}})_i (z(\tilde{\boldsymbol{\eta}})_i - \boldsymbol{\beta}^T \mathbf{X}_{(i)})^2 \\ & + \lambda \left(\alpha \sum_{j=1}^p \tau_j |\beta_j| + \frac{(1-\alpha)}{2} \boldsymbol{\beta}^T \mathbf{L} \boldsymbol{\beta} \right). \end{aligned} \quad (7)$$

(二) 迭代公式

如上所述,坐标下降法的特点是每次迭代只更新估计量的一个维度。这一更新,替代了传统牛顿-拉弗森算法中的最小二乘法参数估计步骤,是整个算法的关键。

为了得到上述更新的表达式,首先,需要将目标函数(7)进行改写,提取包含任一维度 β_u 的项:

$$\begin{aligned} R(\boldsymbol{\beta}) \approx & -\frac{1}{2n} \sum_{i=1}^n a(\tilde{\boldsymbol{\eta}})_i \left(z(\tilde{\boldsymbol{\eta}})_i - \sum_{j \neq u} \tilde{\beta}_j x_{ij} - \beta_u x_{iu} \right)^2 \\ & + \lambda \left\{ \alpha \sum_{j \neq u} \tau_j |\tilde{\beta}_j| + \alpha \tau_u |\beta_u| \right\} \\ & + \lambda \left\{ \frac{1-\alpha}{2} \sum_{\substack{j \neq u \\ k \neq u}} w_{jk} \left(\frac{\beta_j}{\sqrt{d_j}} - \frac{\beta_k}{\sqrt{d_k}} \right)^2 + \frac{1-\alpha}{2} \sum_{j=1}^p w_{uj} \left(\frac{\beta_u}{\sqrt{d_u}} - \frac{\beta_j}{\sqrt{d_j}} \right)^2 \right\} \end{aligned}$$

然后，目标函数对 β_u 求导，得到：

$$\begin{aligned} \frac{\partial R(\beta)}{\partial \beta_u} = & \frac{1}{n} \sum_{i=1}^n a(\tilde{\eta})_i x_{iu} \left(z(\tilde{\eta})_i - \sum_{j \neq u} \tilde{\beta}_j x_{ij} \right) - \lambda(1-\alpha) \sum_{j=1}^p w_{uj} \frac{\tilde{\beta}_j}{\sqrt{d_u d_j}} \\ & + \left[\lambda(1-\alpha) - \frac{1}{n} \sum_{i=1}^n a(\tilde{\eta})_i x_{iu}^2 \right] \beta_u + \lambda \alpha \tau_u \cdot \text{subdiff}(|\beta_u|) \end{aligned}$$

其中， $\tilde{\beta}_j$ 指本次迭代中不作为变量的参数，其取值来自上一次迭代。 $\text{subdiff}(\cdot)$ 指次微分，即以平均变化率的左极限和右极限为端点的闭区间。记

$$\begin{aligned} \rho_u = & -\frac{1}{n} \sum_{i=1}^n a(\tilde{\eta})_i x_{iu} \left(z(\tilde{\eta})_i - \sum_{j \neq u} \tilde{\beta}_j x_{ij} \right) + \lambda(1-\alpha) \sum_{j=1}^p w_{uj} \frac{\tilde{\beta}_j}{\sqrt{d_u d_j}}, \\ z_u = & \lambda(1-\alpha) - \frac{1}{n} \sum_{i=1}^n a(\tilde{\eta})_i x_{iu}^2, \end{aligned}$$

且令 $\partial R(\beta)/\partial \beta_u = 0$ ，得到

$$0 = -\rho_u + z_u \beta_u + \lambda \alpha \tau_u \cdot \text{subdiff}(|\beta_u|).$$

由于

$$\lambda \alpha \tau_u \cdot \text{subdiff}(|\beta_u|) = \begin{cases} \{-\lambda \alpha \tau_u\}, & \beta_u < 0 \\ [-\lambda \alpha \tau_u, \lambda \alpha \tau_u], & \beta_u = 0, \\ \{\lambda \alpha \tau_u\}, & \beta_u > 0 \end{cases}$$

解得

$$\beta_u = \begin{cases} \frac{\rho_u + \lambda \alpha \tau_u}{z_u}, & \rho_u < -\lambda \alpha \tau_u \\ 0, & -\lambda \alpha \tau_u \leq \rho_u \leq \lambda \alpha \tau_u. \\ \frac{\rho_u - \lambda \alpha \tau_u}{z_u}, & \rho_u > \lambda \alpha \tau_u \end{cases}$$

将 ρ_u 和 z_u 代表的原式代回，得到：

$$\hat{\beta}_u \leftarrow \frac{S \left(-\frac{1}{n} \sum_{i=1}^n a(\tilde{\eta})_i x_{iu} \left(z(\tilde{\eta})_i - \sum_{j \neq u} \tilde{\beta}_j x_{ij} \right) + \lambda(1-\alpha) \sum_{j=1}^p w_{uj} \frac{\tilde{\beta}_j}{\sqrt{d_u d_j}}, \lambda \alpha \tau_u \right)}{\lambda(1-\alpha) - \frac{1}{n} \sum_{i=1}^n a(\tilde{\eta})_i x_{iu}^2}. \quad (8)$$

其中， $S(z, \gamma)$ 为软阈值算子（soft-thresholding operator），即

$$\text{sign}(z)(|z| - \gamma)_+ = \begin{cases} z - \gamma, & z > 0 \text{ 且 } \gamma < |z| \\ z + \gamma, & z < 0 \text{ 且 } \gamma < |z|, \\ 0, & \text{其他} \end{cases} \quad \text{且 } \gamma \geq 0.$$

式（8）就是每次迭代的更新公式。

（三）算法表述

坐标下降法借用了牛顿-拉弗森算法的迭代框架，但是将每步更新估计量的方法、公式进行了更改。现将完整的坐标下降算法表述如下：

对于给定的调节参数组合 (λ, α) , 有

步骤 1. 将初始值设为 $\tilde{\beta} = \mathbf{0}$, $\tilde{\eta} = (\tilde{\beta}^T \mathbf{X}_{(1)}, \dots, \tilde{\beta}^T \mathbf{X}_{(n)})^T$ 。

步骤 2. 对于 $i = 1, \dots, n$, 计算 $a(\tilde{\eta})_i$ 和 $z(\tilde{\eta})_i$, 而其他的量在每一次迭代中不变。

步骤 3. 利用(8)式更新 $\hat{\beta}_u$ 。注意, 每次迭代只更新一个维度, 且在 $u = 1, 2, \dots, p, 1, \dots$ 内循环进行。

步骤 4. 更新 $\tilde{\eta} = (\tilde{\beta}^T \mathbf{X}_{(1)}, \dots, \tilde{\beta}^T \mathbf{X}_{(n)})^T$ 。

步骤 5. 重复步骤 2 至步骤 4 直至收敛, 得到 $\hat{\beta}$ 。

(四) 解的路径和交叉检验

注意到, 上述算法必须给定参数组合 (λ, α) 才能进行。为了挑选最佳参数组合, 需要事先给出 (λ, α) 的待选范围, 进而通过交叉检验选择性能最好的一组。

参考 Simon 等 (2011), 本文给出如下寻找参数路径的原则:

(1) 在 $[0, 1]$ 上进行细小分割, 从而为 α 设置参数值。将 α 进行固定, 下面讨论参数 λ 的取值路径。

(2) 将首个 λ 设置得足够大, 使得上一节步骤 1 中的初始值 $\tilde{\beta} = \mathbf{0}$ 。然后逐渐减小 λ 的取值, 直至接近无约束的解。最终可以得到一个递减的 λ 序列。为了实现初始的零估计量, 将其中的最大值取为

$$\lambda_{\max} = \max_j \frac{1}{n\alpha} \sum_{i=1}^n a(\mathbf{0})_i x_{ij} z(\mathbf{0})_i.$$

(3) 注意到, 当 $p > n$ 时, 无约束的解是无法定义的, 往往趋向于无穷大且极不稳定。因此, 当 λ 接近于 0 时, 解的表现很差。实证表明, 最后的几个 λ (即最小的几个 λ) 往往会耗费整个算法 99% 的运行时间, 且交叉检验也不会选择它们。因此, 放弃接近 0 的几个 λ 是完全可以接受的。基于这种考虑, 将 λ_{\min} 的设置分为两种情况。假设 $\lambda_{\min} = \epsilon \lambda_{\max}$, 则当 $n < p$ 时, $\epsilon = 0.05$; 当 $n \geq p$ 时, $\epsilon = 0.0001$ 。

(4) 最后, 在 $[\lambda_{\min}, \lambda_{\max}]$ 之间挑选 $(m+1)$ 个离散的 λ , 挑选方法为

$$\lambda_j = \lambda_{\max} \left(\frac{\lambda_{\min}}{\lambda_{\max}} \right)^{\frac{j}{m}}, j = 0, \dots, m$$

一般地 m 取 50 或者 100。

在选定解的路径后, 需要通过交叉检验得到最优的参数组合 (λ, α) 。本文拟通过 k -折交叉检验进行选择, 即将数据集分为 k 份, 用 $k-1$ 份建模, 用剩下的 1 份检验; 在 k 份数据集中循环上述过程, 直至每一份数据都被用做过检验。

交叉检验需要通过一定的指标来比较模型之间的优劣。由于对数偏似然函数(2)不像高斯(或者其他指数家族)的对数似然函数那样易于在样本之间分割, 因此传统的似然函数指标并不合适。采用 van Houwelingen 等 (2006) 提出的技术, 拟合优度指标为

$$\widehat{CV}(\lambda, \alpha) = -\frac{1}{n} \sum_{i=1}^k \left[\ell \left(\hat{\beta}^{(-i)}(\lambda, \alpha) \right) - \ell^{(-i)} \left(\hat{\beta}^{(-i)}(\lambda, \alpha) \right) \right]. \quad (9)$$

其中, $\hat{\beta}^{(-i)}(\lambda, \alpha)$ 是在给定 (λ, α) 时, 删除第 i 部分的数据后得到的估计量, $\ell^{(-i)}(\cdot)$ 是删除第 i 部分的数据后得到的对数偏似然函数。在实际应用中, 选取使得 $\widehat{CV}(\lambda, \alpha)$ 最小的 (λ, α) 组合。

四、理论性质

这一章主要讨论 GLasso 估计量 $\hat{\beta}$ 的一些理论性质, 以推导和证明为主要内容。GLasso 方法作为 Lasso 方法和 Elastic Net 方法的改进, 这些性质不仅能够说明它本身的特点, 而且能够突出它与其他方法相比的优良特性。本章从估计量 $\hat{\beta}$ 的方差估计、组效应和渐进性质这三个方面入手, 为 GLasso 估计量进行另一个维度的剖析。最后, 借助得到的渐进性质, 给出 GLasso 估计量显著性检验的方法。

(一) 估计量的方差

本节讨论 GLasso 估计量的方差。在不同的情境下, 前面的推导给出了多个不同形式的目标函数。在本节中, 为了讨论的方便, 采用目标函数 (4) 的形式, 并将对数似然函数 $\ell(\beta)$ 进行泰勒展开, 保留主要部分 (二阶部分) 并进行近似, 得到

$$R(\beta) = -(\mathbf{z} - \mathbf{X}\beta)^T \mathbf{A}(\mathbf{z} - \mathbf{X}\beta) + \lambda_1 \sum_{j=1}^p \tau_j |\beta_j| + \lambda_2 \beta^T \mathbf{L}\beta. \quad (10)$$

与前面的符号相对应, $\mathbf{z} = \mathbf{z}(\tilde{\eta})$, $\mathbf{A}(\tilde{\eta}) = 2n\mathbf{A}$ 。之所以选择目标函数 (4) 的惩罚函数形式, 是由于在估计方差时, 两个调节参数的取值范围对计算并无影响, 即 (5) 的形式并不会带来计算上的方便。

为了在形式上获得 $\hat{\beta}$, 需要对 L_1 范数项进行改写:

$$\sum_{j=1}^p \tau_j |\beta_j| \approx \sum_{j=1}^p \tau_j \frac{\beta_j^2}{|\beta_j|},$$

若记 $\mathbf{V} = \text{diag}\{I(\beta_1 \neq 0)\tau_1/|\beta_1|, \dots, I(\beta_p \neq 0)\tau_p/|\beta_p|\}$, 则式 (10) 的惩罚部分可以改写为 $\beta^T(\lambda_1 \mathbf{V} + \lambda_2 \mathbf{L})\beta$ 。由于

$$\frac{\partial R(\beta)}{\partial \beta} = 2\mathbf{X}^T \mathbf{A} \mathbf{X} \beta + 2\mathbf{X}^T \mathbf{A} \mathbf{z} + 2(\lambda_1 \mathbf{V} + \lambda_2 \mathbf{L})\beta,$$

令 $\partial R(\beta)/\partial \beta = 0$, 得到

$$\hat{\beta} = -(\mathbf{X}^T \mathbf{A} \mathbf{X} + \lambda_1 \widehat{\mathbf{V}} + \lambda_2 \mathbf{L})^{-1} \mathbf{X}^T \mathbf{A} \mathbf{z}, \quad (11)$$

其中 $\widehat{\mathbf{V}} = \text{diag}\{I(\hat{\beta}_1 \neq 0)\tau_1/|\hat{\beta}_1|, \dots, I(\hat{\beta}_p \neq 0)\tau_p/|\hat{\beta}_p|\}$ 。注意到, 式 (11) 只是估计量

形式上的表达式，并不能直接用作计算，因为等号的左右两边都包含 $\hat{\beta}_j$ 。但是，在进行 $\hat{\beta}$ 方差的估计时，往往已经通过坐标下降法等方法求得了估计量的具体数值，因此式(11)在此处出现，在估计的层面是可以接受的。

根据 Tibshirani (1997)， \mathbf{z} 的方差可以近似地视作 $\mathbf{A}^{-1}(\tilde{\eta}) = 1/2n \cdot \mathbf{A}^{-1}$ ，因此，由“三明治法则”， $\hat{\beta}$ 方差的估计可以近似为

$$\widehat{\text{cov}}(\hat{\beta}) = \frac{1}{2n} (\mathbf{X}^T \mathbf{A} \mathbf{X} + \lambda_1 \widehat{\mathbf{V}} + \lambda_2 \mathbf{L})^{-1} \mathbf{X}^T \mathbf{A} \mathbf{X} (\mathbf{X}^T \mathbf{A} \mathbf{X} + \lambda_1 \widehat{\mathbf{V}} + \lambda_2 \mathbf{L})^{-1}.$$

由上式可以看出，随着样本量的增大，估计量的方差会逐渐缩小。

(二) 组效应

本节讨论 GLasso 估计量的组效应。Zou 和 Hastie (2005) 提出了 Elastic Net，并对 Elastic Net 估计量的组效应问题进行了深入探讨；此后，Li 和 Li (2007)，Li 和 Li (2008) 以及 Li 等 (2015) 文献讨论了多个衍生于 Elastic Net 的正则化模型的组效应，为当下关于 GLasso 的讨论奠定了坚实且必不可少的基础。

根据 Zou 和 Hastie (2005)，“从定性的方面考虑，如果一组高度相关变量的回归系数趋于相等，则回归方法表现出分组效应。特别是在某些变量完全相同的极端情况下，回归方法应该为相同的变量分配相同的系数。”通俗地讲，所谓“组效应”，就是为高度相关的变量分配相近系数的效应。Lasso 回归的主要效应是稀疏性，在一组高度相关的协变量中只会挑选一个进入模型，因而不具有组效应。而岭回归则完全相反，如果一组 n 个变量完全相同，那么岭回归给它们分配的系数也完全相同，且为同条件下 Lasso 估计量的 $1/n$ 。作为 Lasso 回归和岭回归的组合，Elastic Net 已被证明具有组效应。既然 GLasso 是广义的自适应 Elastic Net 方法，那么十分自然地，可以猜测 GLasso 估计量也具有组效应。

为了简化讨论，不失一般性，仅考虑两个协变量之间的组效应问题，即这两个协变量在图结构中仅与彼此相连。此外，定理和证明中的“高度相关”，一般指的是正相关；当两个协变量为高度负相关时，只需要将对应的符号取负，即可得到完全一致的结论。

定理 1 (组效应) 假设协变量 \mathbf{X} 是标准化的，即

$$\sum_{i=1}^n x_{ij} = 0 \text{ 且 } \sum_{i=1}^n x_{ij}^2 = 1.$$

令 $\hat{\beta}(\lambda_1, \lambda_2)$ 为 GLasso 估计量。假设 $\hat{\beta}_u(\lambda_1, \lambda_2) \hat{\beta}_v(\lambda_1, \lambda_2) > 0$ ，且在图结构中，顶点 u 和 v 仅与彼此相连，即 $d_u = d_v = w(u, v)$ 。定义

$$D_{\lambda_1, \lambda_2}(u, v) = |\hat{\beta}_u(\lambda_1, \lambda_2) - \hat{\beta}_v(\lambda_1, \lambda_2)|,$$

则当相关系数 $\rho = \mathbf{X}_u^T \mathbf{X}_v \rightarrow 1$ 时，

$$D_{\lambda_1, \lambda_2}(u, v) \rightarrow 0.$$

定理 1 的实际意义在于，高度相关但十分重要的变量在 GLasso 过程的建模中不会被任意丢弃，使得模型的应用更具现实意义。在定理 1 中， $|\hat{\beta}_u(\lambda_1, \lambda_2) - \hat{\beta}_v(\lambda_1, \lambda_2)|$ 的上界，实际上依赖于自适 Lasso 部分 τ_u 和 τ_v 的选取。前文提到过， τ_j 中的 $\tilde{\beta}_j$ 可以是任意的一致估计量，那么岭回归估计量自然也是可行的。由于岭回归估计量具有组效应，在定理 1 的条件下，可以得到 $|\tau_u - \tau_v| \rightarrow 0$ 的结论，从而说明了自适应 Lasso 部分不对组效应产生干扰。定理的证明是在 Li 和 Li (2007) 的基础上进行修改得到的，有关细节详见附录 A.1。

(三) 渐进性质

本节讨论 GLasso 估计量的渐进性质。Fan 和 Li (2001) 对 Oracle 性质进行了详细的论述，指明“具有该性质的惩罚似然估计量，在选择正确的模型方面，其性能应当和 Oracle 过程表现得同样好”。具体而言，“当真实参数的一部分为零时，这部分参数以趋向于 1 的概率被估计为零，且非零部分被估计得好似这部分模型已经知道了”。Oracle 性质的重要性在于，“其估计量满足这一性质的过程，在性能上好过单纯的最大似然估计，且符合人们预期”。

Fan 和 Li (2001) 通过构造反例，证明了 Lasso 估计量不具有 Oracle 性质。Zou (2006) 对 Lasso 进行改进，在赋予 L_1 范数惩罚项不同的权重后，使自适应 Lasso 拥有了 Oracle 性质。此外，Zou 和 Hastie (2005) 提出了 Elastic Net 方法，其 L_2 范数项可被视作 GLasso 的特例，但是 Elastic Net 却被证明不具有 Oracle 性质；而 Zou 和 Zhang (2009) 对 Elastic Net 进行了同自适应 Lasso 的一样的改进（即对 L_1 范数惩罚项加权）后，使其拥有了 Oracle 性质。因此，十分自然地，由于 GLasso 是一种广义的自适应 Elastic Net 方法，所以猜想它也应当具有 Oracle 性质，并且拥有这一性质的关键就在于 L_1 范数惩罚项的自适应部分，而其他部分的扰动则是有限的。

为了更好地讨论 GLasso 方法的渐进性质，参考 Zhang 和 Lu (2007)，根据样本量的变化，重新定义一些符号的上下标（仅适用于本节及对应的附录）。记 β_0 为真实的参数向量，而 $\hat{\beta}_n = \arg\max Q_n(\beta)$ ，其中

$$Q_n(\beta) = \ell_n(\beta) - n\lambda_n \sum_{j=1}^p \frac{|\beta_j|}{|\tilde{\beta}_j|} - n\lambda_n^* \sum_{j=1}^p \sum_{k=1}^p \frac{w_{jk}}{2} \left(\frac{\beta_j}{\sqrt{d_j}} - \frac{\beta_k}{\sqrt{d_k}} \right)^2.$$

$\ell_n(\beta)$ 是样本量为 n 时的对数偏似然函数， λ_n 是样本量为 n 时的 L_1 范数项调节参数， λ_n^* 是样本量为 n 时的 L_2 范数项调节参数。此外，还可以将真实的参数向量写作 $\beta_0 = (\beta_{10}^T, \beta_{20}^T)^T$ ，其中 β_{10} 包括全部 q 个非零元素，而 β_{20} 包含剩下的非零元素。类似地，也可以将估计量写作 $\hat{\beta}_n = (\hat{\beta}_{10}^T, \hat{\beta}_{20}^T)^T$ 。

当 Andersen 和 Gill (1982) 中的正则性条件 (A) — (D) 成立时，可以得到如下两个定理。

定理 2 (一致性) 假设 $(\mathbf{X}_{(1)}, T_1, C_1), \dots, (\mathbf{X}_{(n)}, T_n, C_n)$ 独立同分布, 且 T_i 和 C_i 在给定的 $\mathbf{X}_{(i)}$ 时条件独立。若 $\sqrt{n}\lambda_n = O_p(1)$ 且 $\sqrt{n}\lambda_n^* = O_p(1)$, 则 GLasso 估计量满足 $\|\hat{\beta}_n - \beta_0\| = O_p(n^{-1/2})$ 。

定理 3 (Oracle 性质) 假设 $\sqrt{n}\lambda_n \rightarrow 0$, $n\lambda_n \rightarrow \infty$, 且 $n\lambda_n^* \rightarrow 0$, 那么, 在定理 2 的条件下, GLasso 估计量以趋向于 1 的概率:

(1) (稀疏性) $\hat{\beta}_{2n} = \mathbf{0}$;

(2) (渐进正态性) $\sqrt{n}(\hat{\beta}_{1n} - \beta_{10}) \xrightarrow{D} N\{\mathbf{0}, \mathbf{I}_1^{-1}(\beta_{10})\}$, 其中, $\mathbf{I}_1(\beta_{10})$ 是费雪信息矩阵中, 不为零的那一部分子矩阵; 而费雪信息矩阵

$$\mathbf{I}(\beta) = (I_{uv}(\beta))_{p \times p}$$

可以表示为

$$I_{uv}(\beta) = \sum_{i=1}^n \Delta_i \frac{\sum_{j=1}^n I(\tilde{T}_j \geq \tilde{T}_i) x_{ju} x_{jv} \exp(\beta^T \mathbf{X}_{(j)})}{\sum_{j=1}^n I(\tilde{T}_j \geq \tilde{T}_i) \exp(\beta^T \mathbf{X}_{(j)})}.$$

定理 2 对估计量的收敛率进行了阐释, 且说明如果适当地选取 λ_n 和 λ_n^* , 那么 $\hat{\beta}_n$ 是 \sqrt{n} 一致的。定理 3 的条件强于定理 2, 因此能够利用这一性质, 进而说明在 p 固定的情况下, 当 n 趋于无穷时, GLasso 可以表现得好像正确的子模型是已知的一样。

此外, 注意到, GLasso 的 L_1 范数惩罚项十分接近于 L_0 范数惩罚

$$\sum_{j=1}^p I(|\beta_j| \neq 0).$$

由于 $\tilde{\beta}_j$ 是一致估计量, 因此当 n 趋于无穷时, $|\beta_j|/|\tilde{\beta}_j|$ 以概率收敛到 $I(\beta_j \neq 0)$ 。从渐进的角度看, GLasso 的 L_1 范数惩罚项也可以被视作一个自动的最优子集选择过程。

上述两个定理的意义在于, 只要适当地选取调节参数, 那么图结构的加入就只会带来微小的扰动, 不占主导地位, 因而估计量仍然能够保持在自适应 Lasso 中的优良性质。两个定理的证明是在 Zhang 和 Lu (2007) 的基础上进行修改得到的。证明与有关 $\mathbf{I}_1(\beta_{10})$ 的细节一并详见附录 A.2 和附录 A.3。

(四) 显著性检验

本节讨论 GLasso 估计量的显著性检验方法。在传统的极大似然估计中, 估计得到的参数往往不为零, 因此需要通过检验判断它们是否显著。类比这一步骤, 也同样可以针对 GLasso 估计量, 构造相应的显著性检验。

结合定理 3, 在渐进的意义下, 估计量 $\hat{\beta}_{1n}$ 服从正态分布, 即

$$\sqrt{n}(\hat{\beta}_{1n} - \beta_{10}) \xrightarrow{D} N\{\mathbf{0}, \mathbf{I}_1^{-1}(\beta_{10})\}.$$

根据 Slutsky 定理, 可以得到

$$\sqrt{n}(\mathbf{I}_1(\beta_{10}))^{1/2} (\hat{\beta}_{1n} - \beta_{10}) \xrightarrow{D} N\{\mathbf{0}, \mathbf{I}\}.$$

对于服从正态分布的变量，其取平方后，服从卡方分布，因此

$$n(\hat{\beta}_{1n} - \beta_{10})^T \mathbf{I}_1(\beta_{10})(\hat{\beta}_{1n} - \beta_{10}) \xrightarrow{D} \chi^2(q).$$

然而，上式是未知参数 β_{10} 的函数，不能直接用于构造样本统计量。为此，需要对 $\mathbf{I}_1(\beta_{10})$ 进行估计。根据定理 3 的证明过程（详见附录 A.3.2），当样本容量 n 足够大时，有

$$\frac{\hat{\mathbf{I}}_{11}(\beta^*)}{n} \xrightarrow{P} \mathbf{I}_1(\beta_{10}),$$

其中， $\hat{\mathbf{I}}_{11}(\beta)$ 是 $-\ell_n(\beta)$ 对 β 二阶导数的前 $q \times q$ 子矩阵，且 β^* 在 β_0 和 $\hat{\beta}_n$ 之间。因此，可以用 $\hat{\mathbf{I}}_{11}(\hat{\beta}_n)/n$ 估计 $\mathbf{I}_1(\beta_{10})$ ，从而得到统计量

$$W = (\hat{\beta}_{1n} - \beta_{10})^T \hat{\mathbf{I}}_{11}(\hat{\beta}_n)(\hat{\beta}_{1n} - \beta_{10}) \xrightarrow{D} \chi^2(q).$$

1. 全局检验

类似于普通线性回归中的 F 检验，在大样本情形下，上述统计量 W 可以用来检验模型的整体显著性，步骤如下：

- (1) $H_0: \hat{\beta}_{1n} = \mathbf{0}. \quad \Longleftrightarrow \quad H_1: \hat{\beta}_{1n}$ 中的分量不全为 0.
- (2) 确定显著性水平 α .
- (3) 计算统计量

$$W = (\hat{\beta}_{1n} - \mathbf{0})^T \hat{\mathbf{I}}_{11}(\hat{\beta}_n)(\hat{\beta}_{1n} - \mathbf{0}).$$

(4) 考察这个统计量，可以看出， W 越小， $\hat{\beta}_{1n}$ 距离 $\mathbf{0}$ 越近，应当倾向于接受 H_0 ；而 W 越大， $\hat{\beta}_{1n}$ 距离 $\mathbf{0}$ 越远，应当倾向于拒绝 H_0 。因此，根据

$$pv = \Pr(X > W)$$

计算 p-值 pv ，其中 $X \sim \chi^2(q)$.

(5) 如果 $pv \geq \alpha$ ，那么接受原假设 H_0 ；如果 $pv < \alpha$ ，那么拒绝原假设 H_0 ，接受备择假设 H_1 .

事实上，上面的检验就是全局 Wald 检验；即，对于满足条件

$$\sqrt{n}(\hat{\theta}_n - \theta) \xrightarrow{D} N(0, V)$$

的 q 维估计量 $\hat{\theta}_n$ ，有

$$\sqrt{n}V^{-1/2}(\hat{\theta}_n - \theta) \xrightarrow{D} N(0, 1),$$

从而有

$$(\hat{\theta}_n - \theta)^T \left(\frac{V}{n}\right)^{-1} (\hat{\theta}_n - \theta) \xrightarrow{D} \chi^2(q).$$

将上式中的 $\hat{\theta}_n$ 替换为 $\hat{\beta}_{1n}$ ，将 θ 替换为 β_{10} （在假设中是零向量），将 V 替换为 $[\hat{\mathbf{I}}_{11}(\hat{\beta}_n)/n]^{-1}$ ，就能够得到与统计量 W 一致的表达式。

2. 局部检验

与普通线性回归中的 t 检验类似,我们也可以对 GLasso 估计量的每一个分量进行局部显著性检验。

记 $\hat{\beta}_{1n}$ 的第 i 个分量为 $\hat{\beta}_{1n}^{(i)}$. 与全局检验的原假设 $\hat{\beta}_{1n}^{(1)} = \dots = \hat{\beta}_{1n}^{(q)} = 0$ 不同,局部检验的原假设 $\hat{\beta}_{1n}^{(i)} = 0$ 只有一个限制条件,而其它参数不为零的协变量仍然保留在模型中。在这种情况下,为了构造 Wald 检验的统计量,应当将 $\hat{\theta}_n$ 替换为 $\hat{\beta}_{1n}^{(i)}$,将 θ 替换为 0,将 V 替换为利用 $\hat{\beta}_{1n}^{(i)} = 0$ 假设计算得到的 $[\hat{I}_{11}(\hat{\beta}_n)/n]^{-1}$ 的第 i 行、第 i 列的元素。

从而,在大样本情形下,可以得到局部 Wald 检验的步骤:

- (1) $H_0: \hat{\beta}_{1n}^{(i)} = 0. \quad \Leftrightarrow \quad H_1: \hat{\beta}_{1n}^{(i)} \neq 0.$
- (2) 确定显著性水平 α .
- (3) 计算统计量

$$W = (\hat{\beta}_{1n}^{(i)} - 0)^T (\sigma_i)^{-1} (\hat{\beta}_{1n}^{(i)} - 0),$$

其中, σ_i 是在 $\hat{\beta}_{1n}^{(i)} = 0$ 的假设下,计算得到的 $\hat{I}_{11}^{-1}(\hat{\beta}_n)$ 的第 i 行、第 i 列的元素。

(4) 考察这个统计量,可以看出, W 越小, $\hat{\beta}_{1n}^{(i)}$ 距离0越近,应当倾向于接受 H_0 ;而 W 越大, $\hat{\beta}_{1n}^{(i)}$ 距离0越远,应当倾向于拒绝 H_0 。因此,根据

$$pv = \Pr(X > W)$$

计算 p-值 pv ,其中 $X \sim \chi^2(1)$.

(5) 如果 $pv \geq \alpha$,那么接受原假设 H_0 ;如果 $pv < \alpha$,那么拒绝原假设 H_0 ,接受备择假设 H_1 .

最后需要说明的是,以 p-值为根基的显著性检验并不能作为筛选变量的决定性因素。特别地,当 p-值大于显著性水平时,不能将所谓的“不显著”变量轻易剔除。这是由于,检验统计量分布是建立在大样本基础上的渐进分布,当 $n \gg p$ 的条件不成立时,得到的结论就令人怀疑。另一方面,这些所谓的“不显著”变量,在交叉检验的意义下本应当保留在模型中。因此,这样的显著性检验,只能在解释变量时提供一种参考。当出现 $pv \geq \alpha$ 的情况时,正确的做法是结合其他分析方法,以及经济意义进行更加谨慎的判断。

五、模拟

在这一章,拟通过模拟数据,研究 GLasso 方法和 aGLasso 方法在实际中的性能;并将这两种方法与 Lasso 方法和 Elastic Net 方法(简称 Enet 方法)进行比较,从而说明 GLasso 方法和 aGLasso 方法的相对优势。数据模拟和程序执行利用 R 语言进行,主要使用的 R 包为 APML0 (Li 等, 2020)。下面,介绍模拟数据的设计框架和生成原则,给出用以比较各种方法的指标测度,并对最终的模拟结果进行分析。

(一) 生成数据

首先, 设计基本的模拟数据参数。为了测试 GLasso 方法在应对稀疏、分组协变量集合时的性能, 参考王小燕和袁欣 (2018)、Simon 等 (2011), 按照以下原则生成数据:

- (1) 样本量 $n = 100$, 协变量个数 $p = 27$, 将协变量集合分为 3 组。
- (2) 设计矩阵 $\mathbf{X} \sim N(0, \Sigma)$, $\Sigma \sim \mathbf{I}_3 \otimes [\rho^{I(i \neq j)}]$ 。
- (3) 令 $b_k = (-1)^k \exp[-(k-1)/4]$, $k = 1, 2, \dots, 9$, 则 3 组系数分别为

$$\begin{aligned}\beta^{(1)} &= (b_1, 0, 0, b_2, 0, 0, b_3, 0, 0), \\ \beta^{(2)} &= (b_4, 0, 0, b_5, 0, 0, b_6, 0, 0), \\ \beta^{(3)} &= (b_7, 0, 0, b_8, 0, 0, b_9, 0, 0).\end{aligned}$$
- (4) 生存时间 $T = \exp(\mathbf{X}\beta + \epsilon)$, $\epsilon \sim N(0, \mathbf{I})$ 。删失状况从 $B(n, 0.7)$ 中随机生成。

然后, 在基本数据的框架上, 考虑多个模型之间的对比。设置组内相关系数分别为 $\rho = 0.2, \rho = 0.5, \rho = 0.7, \rho = 0.9$, 探讨在不同强度的相关性下, 模型性能的变化。此外, 还需要对比第二章第二节中提出的三种邻接矩阵方法, 考虑不同的图结构是否会引引起模型性能的显著差异。综上, 本章将对比 12 种数据设计下, GLasso 过程、aGLasso 过程、Lasso 过程和 Enet 过程的模拟结果, 并给出对应的分析。

最后需要指出, 为了淡化随机因素的影响, 使得模型之间的比较更具有说服力, 所有的模拟结果都重复 $N = 50$ 次, 取平均值作为最终结果。

(二) 指标测度

模型的对比, 需要一定的测度指标作为模型优劣的衡量标准。本文选取五个指标, 旨在从多个维度综合地评价模型。指标选取如下所述。

1. 交叉检验似然值

交叉检验似然值 (Cross Validation Likelihood, CVM) 具体形式为式 (9), 是每一折交叉检验拟合优度取平均后, 再取负得到的。根据交叉检验选取调节参数的原则, 为了选择使得式 (9) 最小的 (λ, α) , 这一指标越小, 模型的性能就越好。

2. 真阳性变量个数

所谓“真阳性” (True Positive, TP) 的变量, 是真实参数为正, 且估计量也为正的变量。注意到, 在设计模拟数据时, 真实参数 β 被用作生成生存时间, 而估计得到的 $\hat{\beta}$ 被用作计算相对风险度; 这意味着, 在这一章的模拟中, β 与 $\hat{\beta}$ 的符号相反, 因此在计算 TP 时应先对估计得到的 $\hat{\beta}$ 取负。真阳性变量的个数能够反映四种方法在正效应中估计正确模型的能力, 因而这一指标越大 (越接近于真实值) 越好。

3. 真阴性变量个数

同“真阳性”的概念类似, “真阴性” (True Negative, TN) 的变量, 是真实参数为负, 且估计量也为负的变量。与之前的讨论类似, 在计算 TN 时应先对估计得到的 $\hat{\beta}$ 取

负。真阴性变量的个数能够反映四种方法在负效应中估计正确模型的能力，因而这一指标越大（越接近于真实值）越好。

4. C-指标

C-指标（Concordance Index, CI）这一统计量最先由 Harrel (1982) 提出，用以衡量 Cox 比例风险模型的拟合优度。其具体的思想是通过拟合值与失效时间的秩相关程度，说明模型预测的能力。定义个体 i 的预后评分为

$$PS_i = \hat{\beta}^T \mathbf{X}_{(i)}.$$

在所有个体中随机选择两个，比较他们生存时间的长短。如果两个个体都在观测期内失效，那么它们是可比的；如果一个个体在观测期内失效而另一个个体删失，而且失效时间短于删失时间，那么它们也是可比的；在其他情况下，即当删失时间短于失效时间，或者两个个体都删失时，这两个个体显然是不可比的。在计算 C-指标时，只考虑可比样本对，而忽略不可比样本对。在所有的可比样本对中，考虑预后评分相对大小与生存时间相对大小一致的比例，即

$$C = \frac{\sum_{i,j} \Delta_j I(\tilde{T}_i > \tilde{T}_j) I(PS_i > PS_j)}{\sum_{i,j} \Delta_j I(\tilde{T}_i > \tilde{T}_j)},$$

上式就是 C-指标的表达式。

C-指标估计了对于随机选择的两个个体，那个拥有更高预后评分的个体也会生存更久的概率，因此其取值范围是 $[0,1]$ 。在这一意义下，C-指标的不同范围可做如下解读：

- (1) 当 C 接近于 0.5 时，模型的预测能力近乎与随机，性能极差。
- (2) 当 C 大于 0.5 时，说明预后评分与生存时间正相关； C 大于 0.7 时，拟合优度尚可， C 大于 0.8 时拟合优度极佳。
- (3) 当 C 小于 0.5 时，说明预后评分与生存时间负相关； C 小于 0.3 时，拟合优度尚可， C 小于 0.2 时拟合优度极佳。

事实上，只要做变换 $\gamma = 2(c - 0.5)$ ，就可以将 C-指标映射至 $[-1, 1]$ ，得到与传统相关系数意义一致的非参数相关系数。以方便起见，本文仅使用原始的 C-指标。最后需要说明，当 C-指标大于 0.5 时，其取值等于 ROC 曲线下的面积 AUC；当 C-指标小于 0.5 时，AUC 为 $1 - C$ 。

5. 非零变量个数

非零变量个数（Number of Non-zero Variables, NN）计算了估计量中不为 0 的分量个数。由第四章 Oracle 性质中的“稀疏性”，就不难发现，这一指标能够在一定程度上反映模型的变量选择能力。因此，这一指标越接近真实的非零变量个数 q 就越好。

(三) 模拟结果

表 1、表 2、表 3 分别展示了“阈值法”、“协方差法”和“相关系数法”图结构的模拟结果。

首先，不难看出，无论是哪种图结构，随着相关系数从低到高，四种方法（GLasso、aGLasso、Lasso、Enet）的整体性能都在下降，这是广义线性模型在面对高度相关的协变量组时的必然结果。类似于“多重共线性”效应，尽管正则化方法能够解决无法估计参数，或者估计量极不稳定的情况，但仍然会受到高度相关性的影响。此时，式（11）中的 $(\mathbf{X}^T \mathbf{A} \mathbf{X} + \lambda_1 \widehat{\mathbf{V}} + \lambda_2 \mathbf{L})$ 尽管可逆，却仍然比相关性较低时的性能差一些。

其次，在同一个样本数据设计下，Lasso 过程的表现一般而言是最差的，这一点在相关性系数 ρ 较大时比较明显。当 ρ 较小时，Lasso 过程的某几项指标有时会好过 GLasso 和 Enet 过程；但是当 ρ 逐渐变大时，Lasso 过程的指标就随之变差。相关系数 ρ 的大小，能够体现图结构的复杂程度和惩罚力度；因此弱相关的情况就难以体现出 GLasso 过程或者 Enet 过程的优势。

表 1 “阈值法”图结构的模拟结果

$\rho=0.2$						$\rho=0.5$					
	CVM	TP	TN	CI	NN		CVM	TP	TN	CI	NN
GLasso	4.112978	2.56	2.98	0.81719	4.32	GLasso	4.179794	2.28	2.68	0.7917427	6.94
aGLasso	4.066847	2.62	3.32	0.8169169	6.72	aGLasso	4.138735	2.46	3.06	0.7923784	6.6
Lasso	4.104801	2.58	3.02	0.8157945	7.12	Lasso	4.161345	2.08	2.66	0.7890151	6.02
Enet	4.111153	2.58	2.84	0.8074196	7.02	Enet	4.161831	2.48	2.74	0.7993144	8
$\rho=0.7$						$\rho=0.9$					
	CVM	TP	TN	CI	NN		CVM	TP	TN	CI	NN
GLasso	4.243348	1.62	2.34	0.7653542	6.88	GLasso	4.255208	0.96	2.14	0.752411	6.36
aGLasso	4.189707	2.26	3.04	0.7743902	6.16	aGLasso	4.223517	1.6	2.36	0.755088	5.12
Lasso	4.241613	1.76	2.26	0.7672983	6.54	Lasso	4.274585	1.02	1.7	0.7475264	4.6
Enet	4.235473	2	2.32	0.7710681	7.42	Enet	4.263258	1.14	2.16	0.7503381	6.96

表 2 “协方差法”图结构的模拟结果

$\rho=0.2$						$\rho=0.5$					
	CVM	TP	TN	CI	NN		CVM	TP	TN	CI	NN
GLasso	4.167013	2.36	2.8	0.8023857	7.52	GLasso	4.199783	2.34	2.9	0.7961778	7.58
aGLasso	4.066699	2.6	3.4	0.8174979	6.84	aGLasso	4.137477	2.5	3.08	0.7937771	6.8
Lasso	4.101866	2.2	2.8	0.8121094	6.34	Lasso	4.208985	1.94	2.72	0.7831363	6.48
Enet	4.104945	2.5	2.9	0.8163631	7.68	Enet	4.169228	2.44	2.98	0.8032662	8.58
$\rho=0.7$						$\rho=0.9$					
	CVM	TP	TN	CI	NN		CVM	TP	TN	CI	NN
GLasso	4.227207	1.64	2.6	0.7718323	7.66	GLasso	4.23458	1.22	2.18	0.7609502	7.38
aGLasso	4.1892260	2.22	3.02	0.7741648	6.22	aGLasso	4.223183	1.62	2.36	0.7555958	5.14
Lasso	4.239761	1.7	2.34	0.7745107	6.26	Lasso	4.286362	0.9	1.88	0.7552767	5.26
Enet	4.209871	2.18	2.54	0.7761011	7.94	Enet	4.26115	1.14	2.08	0.7510774	6.72

表 3 “相关系数法”图结构的模拟结果

$\rho=0.2$						$\rho=0.5$					
	CVM	TP	TN	CI	NN		CVM	TP	TN	CI	NN
GLasso	4.138109	2.4	3.02	0.811275	7.8	GLasso	4.1649	2.36	2.7	0.7966229	7.66
aGLasso	4.06662	2.62	3.42	0.8177064	6.88	aGLasso	4.138704	2.44	3.06	0.7926071	6.74
Lasso	4.126515	2.2	2.68	0.8008782	6.16	Lasso	4.196573	2.18	2.62	0.7853329	6.84
Enet	4.111153	2.58	2.84	0.8074196	7.02	Enet	4.146348	2.16	2.82	0.7937971	7.26
$\rho=0.7$						$\rho=0.9$					
	CVM	TP	TN	CI	NN		CVM	TP	TN	CI	NN
GLasso	4.241093	1.8	2.42	0.7679569	7.12	GLasso	4.269805	0.98	1.9	0.7518752	6.58
aGLasso	4.1894540	2.24	3.04	0.7745399	6.26	aGLasso	4.223469	1.62	2.36	0.7555496	5.1
Lasso	4.242436	1.72	2.26	0.7651813	6.4	Lasso	4.248502	0.92	1.62	0.7509707	4.72
Enet	4.21687	1.72	2.3	0.771113	6.9	Enet	4.235578	1.32	2.34	0.7636063	8.32

当相关性较大时, GLasso 过程和 Enet 过程往往不相上下, 在前四项指标的对比中难分伯仲。Enet 过程在指标 NN 上要优于 GLasso, 大部分情况下更加接近真实值 9; 而 aGLasso 过程在指标 NN 上, 似乎面临更加严重的问题。这一较为反常的结论, 反映出实践中模型对于稀疏性还是有所偏好。

尽管如此, aGLasso 在剩下四项指标中的表现, 在绝大多数情况下都是最佳的。由于在数据设计中, 正负系数比例大致相等, 因此 L_2 范数惩罚项中针对系数符号的修正, 使得图结构是局部平滑而非间断的; 模拟的结果也进一步说明, 这样的修正使得模型性能显著提升。

与 GLasso 过程和 Enet 过程比较, aGLasso 过程的优良特点, 淋漓尽致地体现在其显著高的 TP、NP、CI 和显著低的 CVM 中。尽管 aGLasso 过程选择的非零系数较少, 但是却能够更加准确地区分这些非零系数的正负效应, 从而估计相对正确的模型。

最后, 将三种图结构进行对比, 发现它们不存在显著的差异。当相关系数较高时, “协方差法”得到的图结构略有优势, 但不是十分明显。这一观察表明, 只要是基于相关性的、能够反映协变量集合内在联系的图结构, 最终都能较好地完成模型选择和模型估

计的任务。鉴于 aGLasso 过程在估计上的优良性能,再加上“阈值法”图结构的相对简洁性(见第二章第二节),因此,第六章采用“阈值法”构建图结构,并执行 aGLasso 过程进行实证分析。

六、实证分析

前面的章节对 aGLasso 过程的理论进行了分析,并说明了 aGLasso 过程的优良性质。第六章将这一过程应用于实证分析,建立上市公司的财务预警体系。通过 aGLasso 过程, Cox 比例风险模型能够将财务指标的分组性和组内相关性纳入考虑,合理高效地筛选财务指标,并据此估计各家上市公司的生存函数。本章首先介绍实证数据的设计,然后执行 aGLasso 过程拟合 Cox 模型并分析估计结果,接着对各项财务指标的图结构进行分析,随后利用生存函数的估计进行动态预测,最终进行总结和分析。

(一) 数据设计与财务指标

根据规定,证券交易所(上交所、深交所)应当对“财务状况异常”的上市公司实行股票的特别处理(Special Treatment,简称 ST)。“财务状况异常”包括最近两个会计年度的审计结果显示的净利润为负值、最近一个会计年度的审计结果显示其股东权益低于注册资本等六种情况,是上市公司出现财务危机的有效预警信号。因此,将上市公司被 ST 作为失效事件,相对应的,未被 ST 的上市公司数据就是删失数据。

在选取样本时,需要考虑生存分析数据左截断(left-truncated)的问题。根据 Kleinbaum 和 Klein (2012),第一类左截断问题指个体在观测开始之前就已经失效,因此这样的个体并不会被包含在研究中,从而给研究带来生存偏差;然而,由于数据获取困难等阻碍因素,第一类左截断问题几乎无法避免。第二类左截断问题指个体的失效时刻在观测开始之后,但是其在观测开始之前的生存时间也应被算入总的生存时间中;为了避免第二类左截断问题,可以将上市公司的寿命,即从上市到被 ST 的时间,或从上市到观测结束的时间,视作生存时间 \tilde{T} 。此外,对 ST 和 *ST 不进行特别区分,且数据中不包含 S*ST、SST 和 NST。对于部分 ST 摘帽后又重新戴帽的企业,只考虑其从上市到第一次被 ST 这段时期。

本文的观测窗口为 2003 年 4 月 9 日至 2020 年 1 月 13 日,在窗口内随机筛选了 243 家企业。其中,117 家企业被 ST,删失率为 51.85%。生存时间 \tilde{T} 的单位为“天”,平均生存期为 2993(天),中值生存期为 2925(天);因此,对于上市企业而言,经营至 8-9 年时应当格外关注企业的财务状况。

参考顾云燕(2016),本文从盈利能力、营运能力、偿债能力、成长能力、现金流量和市值价值这六个方面出发,共选取 30 个指标作为模型的协变量集合,如表 4 所示。

数据来源是万德数据库。注意到，由于所讨论的 Cox 模型（1）不是时变的，因此需要确定数据收集的时间。为了较好地反映企业被 ST 前期的财务状况，并为被预警的企业留出足够的应对时间，因此选择被 ST 前两年（或者观测结束前两年）的财务报表数据作为指标值。

表 4 财务指标

指标含义	指标符号	指标名称	指标含义	指标符号	指标名称
盈利能力	X_1	总资产净利率	偿债能力	X_{16}	资产负债比率
	X_2	营业利润率		X_{17}	清算价值比率
	X_3	销售净利率		X_{18}	现金比率
	X_4	成本费用利润率		X_{19}	产权比率
	X_5	资产报酬率	成长能力	X_{20}	营业收入增长率
	X_6	主营业务利润率		X_{21}	净利润增长率
	X_7	净资产收益率		X_{22}	净资产增长率
营运能力	X_8	应收账款周转率		X_{23}	总资产增长率
	X_9	固定资产周转率	现金流量	X_{24}	现金净流量
	X_{10}	流动资产周转率		X_{25}	经营现金流动负债比
	X_{11}	每股营业收入		X_{26}	资产现金回收率
	X_{12}	存货周转率	市场价值	X_{27}	每股收益
	X_{13}	总资产周转率		X_{28}	每股净资产
偿债能力	X_{14}	流动比率		X_{29}	每股现金流量净额
	X_{15}	速动比率		X_{30}	每股资本公积

这 30 个指标按照经济含义被分为 6 组，且在组内高度相关。将指标之间的相关系数作为图结构 G 的权重 W ，可以得到初始的网络关系，如图 1 所示。

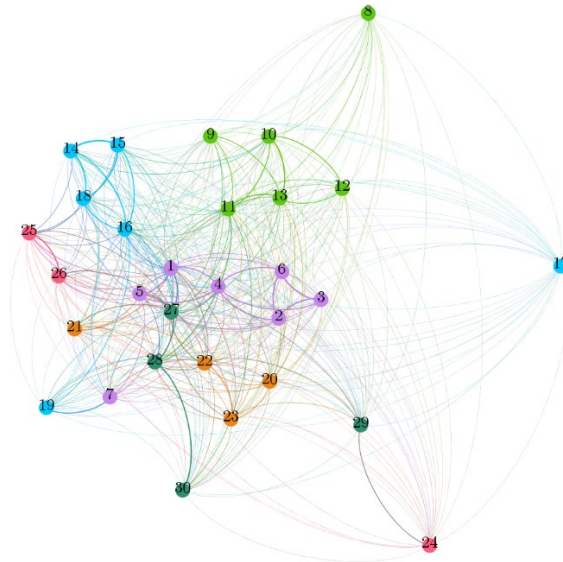


图 1 30 个财务指标的初始图结构

在图 1 中，属于同一组的顶点（协变量）用同一种颜色表示，边的粗细代表了相关系数绝对值的大小。可以观察到，属于同一组顶点的分布呈现聚集状，且它们之间基本

都存在较粗的边连接。在整个网络结构中，一共有 $C(30,2) - 30 = 405$ 条边，这是非常冗杂的。因此，采用“阈值法”对图 1 进行简化。

取 p -值为 $pv = 0.001$ ，则标准正态分布的阈值取 $c = \Phi^{-1}(1 - pv) = 3.09$ ，从而相关系数绝对值的阈值取 $r = [\exp(2c/\sqrt{n-3}) - 1] / [\exp(2c/\sqrt{n-3}) + 1] = 0.1968694$ 。按照“阈值法”图结构的构造方法，可以得到更新后的网络关系，如图 2 所示。

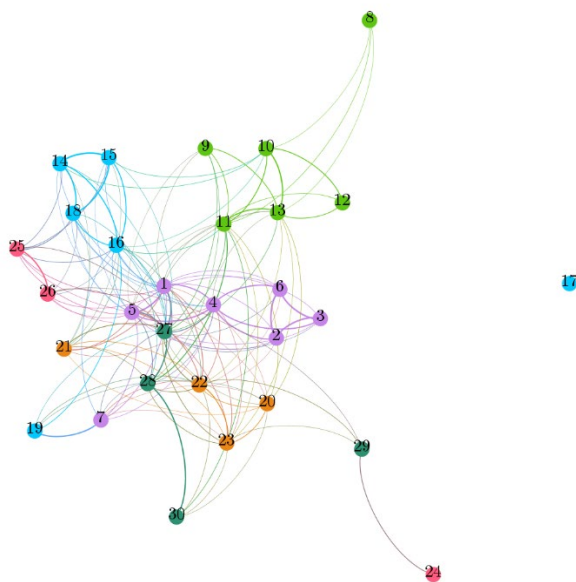


图 2 30 个财务指标的“阈值法”图结构

在丢弃了不显著相关系数代表的边之后，更新的图结构显得简洁了许多。其中， \mathbf{X}_{17} 成为了孤立点，因此在适当的调节参数下，其系数在 aGLasso 过程中必定会被压缩至零。事实上，aGLasso 过程会把大部分的变量系数压缩至零，而只选择图 2 中一个较小的子图结构。

(二) 估计结果

计算图 2 的拉普拉斯矩阵 \mathbf{L} ，建立 Cox 比例风险模型 (1)，并执行 aGLasso 过程。注意，在这之前需要执行一个“初步回归”，得到一致估计量 $\tilde{\beta}$ ，从而得以确定 $\tau_j = 1/|\tilde{\beta}_j|$ 和 $\text{sgn}(\tilde{\beta}_j)$ 的值。aGLasso 过程交叉检验的各项结果如下所示。

表 5 交叉检验结果	
调节参数 λ	0.003536778
调节参数 α	0.16
交叉检验似然值	5.187291
C-指标	0.701375
非零参数个数	10
全局 Wald 检验 p-value	1.737638e-18

交叉检验选择的模型，其 C-指标为 0.701375，拟合优度尚佳。模型从原先 30 个协变量中，仅仅选择了 10 个，构成了如图 3 所示的子图结构。在这一结构中，属于同一组的高度相关协变量能够被同时选择，是因为 L_2 范数惩罚项激励平滑的网络结构，从而使得具有代表性的指标组和其组内指标被同时选出。此外，对该模型进行全局 Wald 检验，得到的 p-值几乎为 0，说明模型整体是显著的。

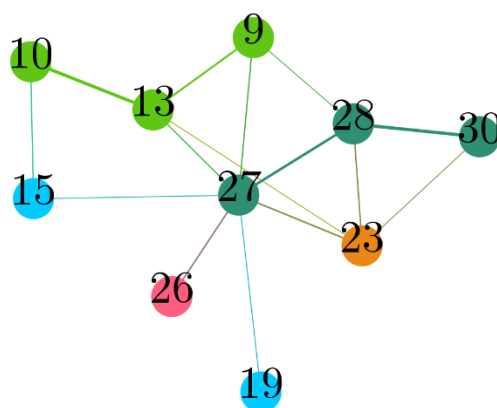


图 3 aGLasso 选择的子图结构

结合表 6 进行分析，可以发现，aGLasso 选择的模型十分偏重上市公司的营运能力、偿债能力和市场价值。在上市公司可能被 ST 的前两年这个节点，这些能力以及他们的指标具有一定的前瞻作用，从而为潜在的财务危机发出预警信号。具体而言，固定资产周转率、流动资产周转率、速动比率、产权比率、资产现金回收率、每股收益和每股净资产是保护性因素，降低了发生财务危机的相对风险。例如，在其他指标不变的情况下，流动资产周转率每增加一个单位，上市公司发生财务危机的相对风险度就变为先前的 54.9%。

在上述几个保护性因素中，资产现金回收率和每股收益的 p-值比较大，暗示着它们相对地“不显著”。然而，如前所述，p-值并不是变量筛选的决定性因素，既然交叉检验选择它们进入模型，这些变量就不可轻易丢弃。从经济意义上看，资产现金回收率指的是经营现金净流量与全部资产的比率，反映了收回的资金占付出资金的比例，能够考察企业全部资产产生现金的能力。因此，资产现金回收率对于企业应对财务危机是很有帮助的，应当保留在模型中。此外，每股收益指税后利润与股本总数之比，是衡量企业经营能力优良指标，自然也能够反映企业的财务状况。

表 6 aGLasso 过程的估计量

分组	符号	名称	系数 $\hat{\beta}_i$	相对风险度 $\exp(\hat{\beta}_i)$	p-value
营运能力	X_9	固定资产周转率	-0.124483045	0.8829532	0.000986***
	X_{10}	流动资产周转率	-0.598824516	0.5494571	0.018955*
	X_{13}	总资产周转率	1.786282945	5.9672307	0.044926*
偿债能力	X_{15}	速动比率	-0.297380847	0.7427611	0.000199***
	X_{19}	产权比率	-0.031651957	0.9688437	0.000017***
成长能力	X_{23}	总资产增长率	0.004866881	1.0048787	0.369582
现金流量	X_{26}	资产现金回收率	-0.026771392	0.9735838	0.095590
市场价值	X_{27}	每股收益	-0.322309567	0.7244739	0.150882
	X_{28}	每股净资产	-0.220922417	0.8017789	0.022635*
	X_{30}	每股资本公积	0.140258796	1.1505715	0.401465

注：***在 0.001 水平上显著；**在 0.01 水平上显著；*在 0.05 水平上显著。

另一方面，总资产周转率、总资产增长率和每股资本公积为危险性因素，他们每增加一个单位，就分别使得上市公司发生财务危机的相对风险度平均增加 4.97 倍、0.005 倍和 0.15 倍。由于总资产周转率与固定资产周转率、流动资产周转率的效应相反，因此应当考虑无形资产和递延资产对上市公司财务状况产生的负面影响。例如，无形资产中的商誉，是一种顺周期的指标，容易受到突发事件影响，一旦经营出现问题，将快速缩水；递延资产实际上是一种长期待摊费用，过高的摊销成本会导致未来利润的下降。事实上，虚高的无形资产和递延资产是常见的财务造假手段，需要引起额外的注意。

由于产权比率、总资产增长率和资产现金回收率的影响很小，它们的相对风险度大致为 1，所以不再对其进行赘述。

最后需要说明的是，与前面几个 p-值较大的保护性因素——资产现金回收率（0.095590）、每股收益（0.150882）相比，两个危险性因素——总资产增长率（0.369582）和每股资本公积（0.401465）的 p-值显然过大。根据显著性检验所提供的参考意义，在进行财务危机分析时可以适当地降低对这两个危险性指标的关注度。

（三）图结构分析

这一节从调节参数的角度出发，考察 λ 和 α 的变化对子图结构选择的影响。回顾 aGLasso 的目标函数：

$$R(\beta) = -\frac{1}{n} \ell(\beta) + \lambda \left[\alpha \sum_{j=1}^p \tau_j |\beta_j| + \frac{(1-\alpha)}{2} \sum_{j=1}^p \sum_{k=1}^p \frac{w_{jk}}{2} \left(\frac{\text{sgn}(\tilde{\beta}_j) \beta_j}{\sqrt{d_j}} - \frac{\text{sgn}(\tilde{\beta}_k) \beta_k}{\sqrt{d_k}} \right)^2 \right].$$

不难猜测，当 α 固定、 λ 增大时，渐强的惩罚函数使得模型逐渐稀疏；而当 λ 固定、 α 减小时，惩罚函数内部更多地偏向于 L_2 范数惩罚部分，使得最终的子图结构牺牲部分稀疏性

以换取平滑性。

表 7 展示了在 $\lambda = \{0.0001, 0.001, 0.01, 0.1\} \otimes \alpha = \{0.2, 0.5, 0.7\}$ 这 12 组调节参数下, 模型所选择的子图结构; 选择结果与上述的分析和猜想一致。由于 λ 的取值范围在数量级之间变动, 而 α 的取值范围仅在 $[0, 1]$ 之间; 因此 λ 从 0.0001 向 0.1 变动时, 网络的稀疏化非常明显; 而 α 在从 0.7 向 0.2 变动的过程中, 网络的平滑化则更加微观。

特别地, 当 $\lambda = 0.1, \alpha = 0.9$ 时, 图结构中只剩下唯一的一个变量, 即每股收益 (\mathbf{X}_{27})。随着调节参数的变动, 在网络逐渐复杂化的过程中, 图结构始终以 \mathbf{X}_{27} 为中心点向四周扩散, 将与它较为接近的指标连入网络。因此, 判断每股收益 (\mathbf{X}_{27}) 是整个网络的中心点, 而每股收益、经营现金流动负债比、速动比率是网络的核心。事实上, 在最终选择的结构图 3 中, 也正以每股收益为中心点。

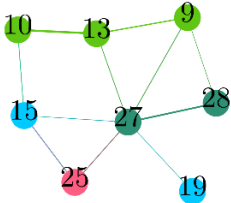
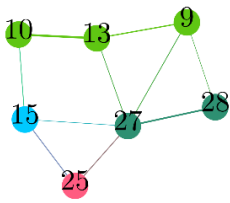
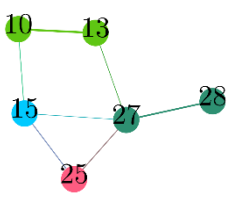
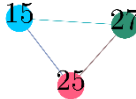


结合上一节的分析, 可知每股收益是一项保护性因素, 越大越好。它不仅能够衡量市场价值, 还能够衡量企业的盈利能力。根据估计的结果, 在其他指标保持不变的情况下, 每股收益每增加一个单位, 上市公司发生财务危机的相对风险度就平均降低 27.6%, 是贡献性很强的预警指标。事实上, 除了反映企业的经营成果, 每股收益对企业历史走势分析、企业行业地位分析、跨行业分析等都具有重大的意义。如果说建立财务危机预警模型有一个指标必不可少, 那么它一定是每股收益。

值得注意的是, 尽管每股收益这一指标如此重要, 在显著性检验中, 它的参数却并不显著。这也从另一个方面说明了显著性检验的非决定性, 因此完全依据 p-值剔除变量是十分危险的。

GLasso 过程或 aGLasso 过程作为正则化家族中的一员, 其独特的优势在于对内嵌图结构的深度利用。aGLasso 过程能够找到普通分析方法所难以找到的中心点和核心指标群, 在解释变量的层面上对显著性检验进行了有益的补充, 为财务预警等应用场景提供了更多的信息。

表 7 不同组合的调节参数下图结构的变化

	$\alpha=0.2$	$\alpha=0.5$	$\alpha=0.7$
$\lambda=0.0001$			
$\lambda=0.001$			

$\lambda=0.01$	 <p>A network graph with 8 nodes: 10 (green), 13 (green), 9 (green), 15 (blue), 27 (teal), 28 (teal), 25 (pink), and 19 (blue). Edges connect (10,13), (13,9), (9,28), (28,27), (27,15), (15,25), (25,19), and (19,27). Edges (10,13), (13,9), and (9,28) are green; (15,27), (15,25), and (27,28) are teal; (25,19) and (19,27) are blue.</p>	 <p>A network graph with 7 nodes: 10 (green), 13 (green), 9 (green), 15 (blue), 27 (teal), 28 (teal), and 25 (pink). Edges connect (10,13), (13,9), (9,28), (28,27), (27,15), (15,25), and (25,27). Edges (10,13), (13,9), and (9,28) are green; (15,27), (15,25), and (27,28) are teal; (25,27) is pink.</p>	 <p>A network graph with 6 nodes: 10 (green), 13 (green), 27 (teal), 28 (teal), 15 (blue), and 25 (pink). Edges connect (10,13), (13,27), (27,28), (28,27), (27,15), (15,25), and (25,27). Edges (10,13) and (13,27) are green; (15,27), (15,25), and (27,28) are teal; (25,27) is pink.</p>
$\lambda=0.1$	 <p>A network graph with 3 nodes: 15 (blue), 27 (teal), and 25 (pink). Edges connect (15,27), (15,25), and (25,27). Edges (15,27) and (15,25) are blue; (25,27) is pink.</p>	 <p>A network graph with 2 nodes: 27 (teal) and 25 (pink). Edges connect (27,25). The edge is pink.</p>	 <p>A network graph with 1 node: 27 (teal).</p>

（四）动态预测

根据 Klein 和 Moeschberger 等 (2005), 完成对 Cox 比例风险模型的估计后, 可以利用 Breslow 估计量, 基于基准风险函数 $\lambda_0(t)$, 考察每一个个体生存超过给定时刻的概率, 即估计每一个个体的生存函数。首先考虑累积风险函数

$$\Lambda_0(t) = \int_0^t \lambda_0(u) du,$$

它的估计量为:

$$\hat{\Lambda}_0(t) = \sum_{i=1}^n \frac{I(\tilde{T}_i < t)}{\sum_{j=1}^n I(\tilde{T}_j \geq \tilde{T}_i) \exp(\hat{\beta}^T \mathbf{X}_{(j)})}.$$

$\hat{\Lambda}_0(t)$ 是一个在观测失效时间处跳跃的阶梯函数; 随着时间的推移, 求和项数越来越多, 因此它也是一个非递减的函数, 每次间断都在向上跳跃。然后, 基准生存函数 $S_0(t) = \exp[-\Lambda_0(t)]$ 可以被估计为:

$$\hat{S}_0(t) = \exp[-\hat{\Lambda}_0(t)].$$

这是一个非递增的阶梯函数, 含义为某个协变量为 $\mathbf{X}_{(i)} = \mathbf{0}$ 的个体的生存函数。对于一般的个体而言, 给定协变量 $\mathbf{X}_{(i)} = \mathbf{X}_0$, 则其生存函数的估计量是:

$$\hat{S}(t|\mathbf{X}_0) = \hat{S}_0(t)^{\exp(\hat{\beta}^T \mathbf{X}_0)}. \quad (12)$$

依据上述理论, 表 8 给出了利用样本中 243 家企业拟合得到的基准生存函数 $\hat{S}_0(t)$, 其中的时刻以“天”为单位, 并且时刻间距大致为 365 天。上市公司寿命大于 3 年的概率超过 90%; 但是当寿命大于 4 年时, 这一概率就忽然降至不到 80%。生存概率为 50% 的临界点发生在第 6 至 7 年之间, 而公司寿命大于 9 年的概率就已经下降到低于 30%。尽管表 8 讨论了 18 年的生存期, 但是后 9 年的生存概率已经非常低; 这说明, 平均而言, 上市公司的生存期为 8-9 年, 而这也与第六章第一节中的叙述相一致。

表 8 基准生存函数估计量

t (天)	302	696	1083	1456	1827	2191	2572	2925	3288
$\hat{S}_0(t)$	0.9898	0.9596	0.9090	0.7966	0.7469	0.6661	0.4729	0.3876	0.2911
t (天)	3639	4044	4389	4758	5039	5422	5832	6187	6389
$\hat{S}_0(t)$	0.2050	0.1634	0.1331	0.1101	0.0650	0.0377	0.0261	0.0090	0.0042

由式 (12) 可知, 对于某一个特定的企业, 影响其生存函数的因素, 除了基准生存函数, 还包括其各项财务指标情况。为了进一步考察不同企业的生存函数, 本文选取了五个具有代表性的企业进行分析。他们的生存函数估计量如图 4 所示。

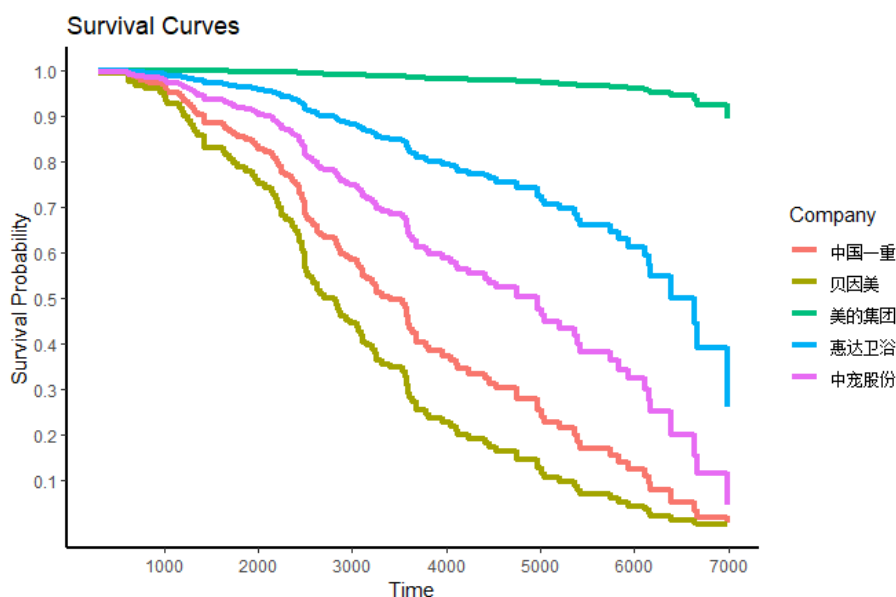


图 4 5 家企业的生存函数估计

美的集团（000333.SZ）的财务状况是五个企业中最好的，其生存函数只有小幅度下降，生存超过 7000 天的概率将近 90%。惠达卫浴（603385.SH）的生存函数下降也较为缓慢，考察生存概率为 50%这一点，其预期寿命高达 17 年；惠达卫浴上市时间不长，目前正处于企业生存的早期，财务状况较好，应当将这一趋势保持下去。中宠股份（002891.SZ）也是生存函数下降相对缓慢的企业，考察生存概率为 50%这一点，对应的寿命为 13 年；与惠达卫浴类似，中宠股份上市时间较短，目前仍然处于公司快速成长时期，因此应当顺应宠物食品市场蓬勃发展的大背景，延续良好的财务状况。

与前面的三家企业相比，中国一重（601106.SH）和贝因美（002570.SZ）的生存函数显然下降较快。这两家企业分别上市于 2010 和 2011 年，目前已经达到了样本集中的平均失效时间。考察他们生存超过 10 年和 9 年的概率，大致为 40%，因此，这两家企业的财务状况必须引起经营者、债权人以及投资者足够的重视。随着时间的推移，如果不加以改善，它们很有可能陷入财务危机的困境。

（五）结论与建议

在上述分析中，这一章建立了 Cox 比例风险模型，并利用 aGLasso 过程，同时实现了财务预警模型的指标选择和系数估计，并且在原始的图结构中筛选得到了更为简洁的子图结构。模型的选择和估计表明，固定资产周转率、流动资产周转率、速动比率、每股收益和每股净资产是保护性因素，这些指标的提高有利于降低财务危机发生的风险。特别地，利用不同的调节参数组合，得到的多个子图结构指明了以每股收益为中心的网络关系。因此，上市公司应当特别重视每股收益这一指标和以它为中心的指标群，在陷入财务危机前就采取措施进行补救。

此外，估计得到的生存函数，以可视化的方法，进一步帮助企业动态检测自己的财

务状况。处于生存早期的企业往往状况良好，而生存至 8-9 年的企业则需要密切关注自己的生存曲线；当生存概率低于 50% 时，就需要从上述各项保护性因素入手，及时地预防、化解潜在的财务危机。

最后需要说明的是，在实际应用中，经营者、债权人、投资者、监管者等应当注意数据更新。随着时间的发展，各家上市公司采取行动调整经营，财务状况会发生较大的变化，而筛选出的财务指标、子图结构，以及估计得到的生存曲线也应当与发展相适应，从而为模型的使用者提供准确、合理的参考。

七、讨论与展望

（一）研究结论

本文将 Lasso 方法和图结构相结合，构造广义的 Elastic Net 惩罚函数，并将其应用于 Cox 风险比例模型，用以估计模型的参数。这一方法能够将协变量集合“分组相关”的结构纳入考虑，筛选出平滑、性能优良的网络结构，对模型的解释提供新的见解。文章给出了模型估计的算法，并在此基础上进行推导、模拟、实证，得到了如下结论：

（1）GLasso 估计量的理论性质优良。首先，它的方差可以利用“三明治”公式得到，且随着样本容量的增加递减。其次，GLasso 估计量具有组效应，倾向于给相关性强的若干个协变量分配相近的回归系数；这实现了网络结构的平滑性，以及其与稀疏性的平衡。此外，GLasso 估计量具有 Oracle 性质，表现得如同真值一样好，是极为优良的估计量。最后，根据估计量非零部分的渐近正态性，本文构造得到了显著性检验的统计量，使得对 GLasso 方法的分析更加丰富、系统、全面。

（2）模拟研究将 Lasso 方法、Enet 方法、GLasso 方法和 aGLasso 方法进行了性能上的对比。根据模拟结果，在变量间存在较强相关性时，Enet 方法、GLasso 方法和 aGLasso 方法的优势能够得到充分体现。此外，aGLasso 过程区分非零系数正负效应的能力最强，在其他的多个指标上也表现最佳。除了理论上的证明，实践也能够验证它的优良性质。

（3）实证研究将 GLasso 方法应用于上市公司财务预警模型的建立，通过交叉检验筛选得到了 10 个财务指标，并结合经济意义进行了分析。此外，通过调节参数的放缩，模型内嵌的图结构精准、形象地识别了 30 个指标之间中心点和核心部分，为上市公司的财务预警和危机应对提供了有效着力点。最后，通过对多家企业生存函数的估计，可以更加直观地向管理者、投资人等展示企业未来的财务能力和发展潜力。

（二）不足和展望

（1）Cox 风险比例模型能够在不同的时点对企业的生存概率进行预测，是一种跟进时刻的动态模型；然而，这也意味着，在更加广义的情况下，协变量也应当是时变的而非固定的。本文选取的数据为上市公司被 ST（或观测结束）的前 2 年数据，在一定程度上实现了协变量和时间的相关。尽管如此，时变的 Cox 风险比例模型，显然是提高预测性能更好的解决方案，对理论分析、数据质量和实证检验也提出了更高的要求。将 GLasso 方法应用于时变 Cox 风险比例模型的求解，是下一步的研究方向。

（2）作为高维统计家族的一员，GLasso 过程能够处理 $n < p$ 的情况，而不会受到多重共线性导致的无法进行参数估计的困扰。然而本文讨论的大多是当 n 趋于无穷大时的情形，即大样本情形。过去的文献已证明，对于正则化家族的其他成员，相比于大样本情形， $n < p$ 的情况会在一定程度上降低模型性能。这一结论是否适用于 GLasso 方法，以及它在 $n < p$ 的情况具体表现如何，是下一步的研究方向。

参考文献

- [1] D. R. Cox. Regression Models and Life-Tables [J]. *Journal of the Royal Statistical Society: Series B*, 1972: V34(2) 187-220.
- [2] R. Tibshirani. Regression Shrinkage and Selection via the Lasso [J]. *Journal of the Royal Statistical Society: Series B*, 1996: V58(1) 267-288.
- [3] R. Tibshirani. The Lasso Method for Variable Selection in the Cox Model [J]. *Statistics in Medicine*, 1997: V16 385-395.
- [4] J. Li and R. Li. Variable Selection via Nonconcave Penalized Likelihood and its Oracle Properties [J]. *Journal of the American Statistical Association*, 2001: V96(456) 1348-1360.
- [5] J. Li and R. Li. Variable Selection for Cox's Proportional Hazards Model and Frailty Model [J]. *Annals of Statistics*, 2002: V30(1) 74-99.
- [6] H. Zou. The Adaptive Lasso and Its Oracle Properties [J]. *Journal of the American Statistical Association*, 2006: V101(476) 1418-1429.
- [7] H. Zhang and W. Lu. Adaptive Lasso for Cox's proportional hazards model [J]. *Biometrika*, 2007: V94(3) 691-703.
- [8] H. Zou and T. Hastie. Regularization and Variable Selection via the Elastic Net [J]. *Journal of the Royal Statistical Society: Series B*, 2005: V67(2) 301-320.
- [9] T. Wu and K. Lange. Coordinate Descent Algorithms for Lasso Penalized Regression [J]. *Annals of Applied Statistics*, 2008: V2(1) 224-244.
- [10] C. Li and H. Li. Network-constrained Regularization and Variable Selection for Analysis of Genomic Data [J]. *Bioinformatics*, 2009: V24(9) 1175-1182.
- [11] C. Li and H. Li. Variable Selection and Regression Analysis for Graph-Structured Covariates with an Application to Genomics [J]. *Annals of Applied Statistics*, 2010: V4(3) 1498-1516.
- [12] H. Sun, W. Lin, R. Feng, and H. Li. Network-regularized High-dimensional Cox Regression for Analysis of Genomic Data [J]. *Statistica Sinica*, 2014: V24(3) 1433-1459.
- [13] J. Huang, S. Ma, H. Li, and C. Zhang. The Sparse Laplacian Shrinkage Estimator for High-dimensional Regression [J]. *Annals of Statistics*, 2011: V39(4) 2021-2046.
- [14] W. Fu. Penalized Regressions: The Bridge Versus the Lasso [J]. *Journal of Computational and Graphical Statistics*, 1998: V7(3) 397-416.
- [15] J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani. Pathwise Coordinate Optimization [J]. *Annals of Applied Statistics*, 2007: V1(2) 302-332.
- [16] J. Friedman, T. Hastie, and R. Tibshirani. Regularization Paths for Generalized Linear

- Models via Coordinate Descent [J]. *Journal of Statistical Software*, 2010: V33(1) 1-22.
- [17] N. Simon, J. Friedman, T. Hastie, and R. Tibshirani. Regularization Paths for Cox's Proportional Hazards Model via Coordinate Descent [J]. *Journal of Statistical Software*, 2011: V39(5) 1-13.
- [18] F. Chung. *Spectral Graph Theory* [M]. Providence: Amer. Math. Soc., 2007: V92.
- [19] P. Andersen and R. Gill. Cox's Regression Model for Counting Processes: A Large Sample study [J]. *Annals of Statistics*, 1982: V10(4) 1100-1120.
- [20] C. Li and H. Li. Network-constrained Regularization and Variable Selection for Analysis of Genomic Data. Working Paper, University of Pennsylvania, 2007.
- [21] T. Hastie and R. Tibshirani. *Generalized Additive Models* [M]. London: Chapman and Hall, 1990: 213-214.
- [22] van Houwelingen, Bruinsma, Hart Augustinus, Van't Veer, and Wessels Lodewyk. Cross-validated Cox Regression on Microarray Gene Expression Data [J]. *Statistics in Medicine*, 2006: V25(18) 3201-3216.
- [23] C. Li, X. Wei, and H. Dai. Adaptive Elastic Net Method for Cox Model [J]. math.ST/1507.06371, 2015.
- [24] H. Zou and H. Zhang. On the Adaptive Elastic-net with a Diverging Number of Parameters [J]. *Annals of Statistics*, 2009: V37(4) 1733-1751.
- [25] X. Li, S. Xie, D. Zeng, and Y. Wang. APML0: Augmented and Penalized Minimization Method L0 [EB/OL]. R package version 0.10,
URL <https://cran.r-project.org/web/packages/APML0/index.html>
- [26] 王小燕, 袁欣. 基于惩罚组变量选择的 COX 财务危机预警模型[J]. 系统工程, 2018: V36(3) 113-121.
- [27] F. Harrell, R. Califf, D. Pryor, K. Lee, and R. Rosati. Evaluating the Yield of Medical Tests [J]. *Journal of the American Medical Association*, 1982: V247(18) 2543-2546.
- [28] D. Kleinbaum and M. Klein. *Survival Analysis: A Self-Learning Text*, 3rd edition [M]. New York: Springer, 2012: 131-142.
- [29] 顾云燕. 基于 Lasso 和 Cox 模型的上市中小企业财务预警分析[D]. 兰州大学, 2016.
- [30] J. Klein and M. Moeschberger. *Survival Analysis: Techniques for Censored and Truncated Data*, 2nd edition [M]. New York: Springer, 2005: 283-285.

致谢

时光荏苒，岁月如梭，四年的本科生涯即将画上句号，回顾往事，心中泛起的层层涟漪不知如何诉说。从懵懂无知的青涩少年，到寻得自己的志趣所在：一路走来，我要感谢的人有许多。首先，我要感谢数学院的黄超群、晏华辉老师，正是他们给我打开了一扇窗，通过严谨的分析思维、丰富的模型实践，使我看到了数学的深刻、厚重与广阔，为我学习统计学打下了扎实的基础。第二，我要感谢金统院的苑迪、谭发龙老师，他们分别是我从入门到进阶的引路人；他们的专业课深入浅出，观点新颖，帮助我开拓了专业的国际视野，进而找寻到了自己的兴趣点。第三，我要特别感谢加州大学洛杉矶分校和斯坦福大学的 Ciprian Manolescu 教授，他在我赴美交流期间给予了极大的帮助与肯定；尽管相处时间不长，我却被这位谦逊的大师深深打动，感受到了学术的纯净和魅力。这几位老师在我学习和升学的各个阶段，都给予了弥足珍贵的帮助，我对此的感激溢于言表。第四，我要感谢统计 1601 班的各位同学，尤其是三位舍友，一起相处的四年，我因为你们而收获关怀和快乐；即将别离，情谊永在！

然而，在充满焦虑的毕业季，对我帮助最大的莫过于我的指导老师王小燕老师。感谢她向我介绍了这样一个有趣的话题，使得我能够走出“模型套娃”的困境，像一个真正的研究者那样，站在学术发展的角度、以成体系的大局观思考问题。王小燕老师推荐的参考文献、给出的研究思路清晰而明确，帮助我在研究初期的混乱中迅速找到切入点，准确把握前进的方向。随着研究的深入，各类文献和模型的特点都迅速指明了题目设置本身的巧妙之处——作为多个部分的结合，所研究的模型兼具了独特的性质，参考文献又不至于缺乏。如果没有王小燕老师的指点与鼓励，仅凭自己粗浅的了解，我无论如何也不可能识别到这样一个方向，更遑论得到一些自己的心得。这一段研究经历，将使我更加从容、自信地面对未来的研究生生涯，并坚定了继续探索的决心。

最后，我要感谢我亲爱的父母。正是他们在背后一直默默地关心我、支持我、爱护我，一边耳提面命，另一边却又帮我解决各种棘手的问题，是我最强大的后盾。

本科四年，永生难忘。其中的苦涩、泪水与快乐，将在未来的道路中始终鞭策着我，告诉我初心勿忘，要守住自己所信的道。

附录 A

(一) 定理 1 的证明

下面，参考 Li 和 Li (2007)，对定理 1 进行证明。根据 $\hat{\beta}_u(\lambda_1, \lambda_2)\hat{\beta}_v(\lambda_1, \lambda_2) > 0$ 的假设，易得

$$\text{sgn}\{\hat{\beta}_u(\lambda_1, \lambda_2)\} = \text{sgn}\{\hat{\beta}_v(\lambda_1, \lambda_2)\}, \quad \hat{\beta}_u(\lambda_1, \lambda_2) \neq 0, \quad \hat{\beta}_v(\lambda_1, \lambda_2) \neq 0.$$

考虑利用二阶泰勒展开近似的目标函数

$$R(\beta) \approx -\frac{1}{2n} \sum_{i=1}^n a(\tilde{\eta})_i (z(\tilde{\eta})_i - \beta^T \mathbf{X}_{(i)})^2 + \lambda_1 \sum_{j=1}^p \tau_j |\beta_j| + \lambda_2 \sum_{j=1}^p \sum_{k=1}^p \frac{w_{jk}}{2} \left(\frac{\beta_j}{\sqrt{d_j}} - \frac{\beta_k}{\sqrt{d_k}} \right)^2.$$

$R(\beta)$ 对 β_u 和 β_v 分别求导，且令

$$\left. \frac{\partial R(\beta)}{\partial \beta_u} \right|_{\beta=\hat{\beta}} = 0, \quad \left. \frac{\partial R(\beta)}{\partial \beta_v} \right|_{\beta=\hat{\beta}} = 0,$$

从而有

$$\begin{aligned} \frac{1}{n} \sum_{i=1}^n x_{iu} a(\tilde{\eta})_i (z(\tilde{\eta})_i - \hat{\beta}^T \mathbf{X}_{(i)})^2 + \lambda_1 \tau_u \text{sgn}(\hat{\beta}_u) + \lambda_2 \hat{\beta}_u - \lambda_2 \sum_{j=1}^p w_{uj} \frac{\hat{\beta}_j}{\sqrt{d_u d_j}} &= 0 \\ \frac{1}{n} \sum_{i=1}^n x_{iv} a(\tilde{\eta})_i (z(\tilde{\eta})_i - \hat{\beta}^T \mathbf{X}_{(i)})^2 + \lambda_1 \tau_v \text{sgn}(\hat{\beta}_v) + \lambda_2 \hat{\beta}_v - \lambda_2 \sum_{j=1}^p w_{vj} \frac{\hat{\beta}_j}{\sqrt{d_v d_j}} &= 0. \end{aligned}$$

上面两式相减，得到

$$\begin{aligned} \frac{1}{n} \sum_{i=1}^n (x_{iu} - x_{iv}) a(\tilde{\eta})_i (z(\tilde{\eta})_i - \beta^T \mathbf{X}_{(i)})^2 + \lambda_1 \tau_v (\tau_u - \tau_v) \text{sgn}(\hat{\beta}_v) + 2\lambda_2 (\hat{\beta}_u - \hat{\beta}_v) \\ = 0. \end{aligned}$$

上面等式的推导用到了 $\text{sgn}\{\hat{\beta}_u(\lambda_1, \lambda_2)\} = \text{sgn}\{\hat{\beta}_v(\lambda_1, \lambda_2)\}$ 和 $d_u = d_v = w(u, v)$ 的条件。然后，将上式进行变换，可以得到不等式

$$|\hat{\beta}_u - \hat{\beta}_v| \leq \frac{1}{2n\lambda_2} \sum_{i=1}^n |x_{iu} - x_{iv}| |a(\tilde{\eta})_i| (z(\tilde{\eta})_i - \beta^T \mathbf{X}_{(i)})^2 + \frac{\lambda_1}{2\lambda_2} |\tau_u - \tau_v|. \quad (\text{A1})$$

(1) 由于 \mathbf{X} 是标准化的，所以

$$2(1 - \rho) = \|\mathbf{X}_u - \mathbf{X}_v\|^2 = \sum_{i=1}^n (x_{iu} - x_{iv})^2,$$

因此当 $\rho \rightarrow 1$ 时， $|x_{iu} - x_{iv}| \rightarrow 0$ ，从而 (A1) 的第一项趋向于 0。

(2) 考虑 $\tau_u = 1/|\tilde{\beta}_u|$ ， $\tau_v = 1/|\tilde{\beta}_v|$ 。由于 $\tilde{\beta}_u$ 和 $\tilde{\beta}_v$ 可以是任意的一致估计量，所以不妨选择岭回归估计 $\tilde{\beta}_u$ 和 $\tilde{\beta}_v$ 。岭回归过程与式 (A1) 对应的不等式为

$$|\tilde{\beta}_u - \tilde{\beta}_v| \leq \frac{1}{2n\lambda_2} \sum_{i=1}^n |x_{iu} - x_{iv}| |a(\tilde{\eta})_i| (z(\tilde{\eta})_i - \tilde{\beta}^T \mathbf{X}_{(i)})^2.$$

基于与 (1) 同样的原因, 当 $\rho \rightarrow 1$ 时, $|x_{iu} - x_{iv}| \rightarrow 0$, 从而 $|\tilde{\beta}_u - \tilde{\beta}_v| \rightarrow 0$.

由于岭回归估计量不为零, 记 $m = \min\{|\tilde{\beta}_1|, \dots, |\tilde{\beta}_p|\} > 0$, 由此得到

$$0 \leq |\tau_u - \tau_v| = \left| \frac{1}{|\tilde{\beta}_u|} - \frac{1}{|\tilde{\beta}_v|} \right| \leq \frac{|\tilde{\beta}_u - \tilde{\beta}_v|}{|\tilde{\beta}_u||\tilde{\beta}_v|} \leq \frac{|\tilde{\beta}_u - \tilde{\beta}_v|}{m^2} \rightarrow 0,$$

由夹逼定理, 得到 $|\tau_u - \tau_v| \rightarrow 0$, 因此 (A1) 的第二项趋向于 0.

将 (1) 与 (2) 结合, 考虑式 (A1), 再次使用夹逼定理, 得到

$$|\hat{\beta}_u - \hat{\beta}_v| \rightarrow 0.$$

从而定理 1 得证。

(二) 定理 2 的证明

定义计数过程 $N_i(t) = \Delta_i I(\tilde{T}_i \leq t)$, 在险过程 $Y_i(t) = I(\tilde{T}_i \geq t)$. 在定理 2 的证明中, 协变量 \mathbf{X} 可以是时变的, 记作 $\mathbf{X}(t)$. 不失一般性, 假定 $t \in [0, 1]$, 则对数偏似然函数的“费雪信息矩阵”为

$$\mathbf{I}(\beta_0) = \int_0^1 \mathbf{v}(\beta_0, t) s^{(0)}(\beta_0, t) \lambda_0(t) dt.$$

其中,

$$\mathbf{v}(\beta, t) = \frac{\mathbf{s}^{(2)}(\beta, t)}{s^{(0)}(\beta, t)} - \left(\frac{\mathbf{s}^{(1)}(\beta, t)}{s^{(0)}(\beta, t)} \right) \left(\frac{\mathbf{s}^{(1)}(\beta, t)}{s^{(0)}(\beta, t)} \right)^T,$$

且 $\mathbf{s}^{(k)}(\beta, t) = E[\mathbf{X}(t)^{\otimes k} Y(t) \exp\{\beta^T \mathbf{X}(t)\}]$, $k = 0, 1, 2$. (对向量 \mathbf{v} , $\mathbf{v}^{\otimes 0} = 1$, $\mathbf{v}^{\otimes 1} = \mathbf{v}$, $\mathbf{v}^{\otimes 2} = \mathbf{v}\mathbf{v}^T$.) 需要说明 2 点:

(1) 由于 $\mathbf{I}(\beta_0)$ 描述的是总体, 因此上述 $\mathbf{X} = (X_1, \dots, X_p)^T$ 是一个随机向量, 其中 X_j 为第 j 个协变量.

(2) 记 $\mathbf{I}_1(\beta_{10}) = \mathbf{I}_{11}(\beta_{10}, \mathbf{0})$, 其中 $\mathbf{I}_{11}(\beta_{10}, \mathbf{0})$ 是 $\mathbf{I}(\beta_0)$ 中的不为零的 $s \times s$ 子矩阵, 而 $\beta_{20} = \mathbf{0}$.

下面, 对 Zhang 和 Lu (2007) 的相关部分进行修改, 对定理 2 进行证明. 定义 $\mathbf{s}_n(\beta) = \partial \ell_n(\beta) / \partial \beta$, $\nabla \mathbf{s}_n(\beta) = \partial \mathbf{s}_n(\beta) / \partial \beta^T$. 则对数偏似然函数可以写作

$$\ell_n(\beta) = \sum_{i=1}^n \int_0^1 \beta^T \mathbf{X}_{(i)}(t) dN_i(t) - \int_0^1 \log \left[\sum_{i=1}^n Y_i(t) \exp\{\beta^T \mathbf{X}_{(i)}(t)\} \right] d\bar{N}(t).$$

其中,

$$\bar{N}(t) = \sum_{i=1}^n N_i(t).$$

由 Andersen 和 Gill (1982) 中的定理 4.1 和引理 3.1, 有, 对于 β_0 的某个邻域中的任意 β ,

$$\begin{aligned} \frac{1}{n} \{ \ell_n(\beta) - \ell_n(\beta_0) \} &= \int_0^1 \left[(\beta - \beta_0)^T s^{(1)}(\beta_0, t) - \log \left\{ \frac{s^{(0)}(\beta, t)}{s^{(0)}(\beta_0, t)} \right\} s^{(0)}(\beta_0, t) \right] \lambda_0(t) dt \\ &\quad + O_p \left(\frac{\|\beta - \beta_0\|}{\sqrt{n}} \right). \end{aligned}$$

考虑 C -球 $B_n(C) = \{\beta: \beta = \beta_0 + n^{-1/2}\mu, \|\mu\| \leq C\}$, $C > 0$, 并将它的边界记作 $\partial B_n(C)$. 注意到, 当 n 很大时, $Q_n(\beta)$ 是严格凸的; 此时, $Q_n(\beta)$ 存在唯一的极大值点。因此, 要证明定理 2, 只需证明 $\forall \epsilon > 0, \exists C$ (足够大), 使得

$$\Pr \left\{ \sup_{\beta \in \partial B_n(C)} Q_n(\beta) < Q_n(\beta_0) \right\} \geq 1 - \epsilon \quad (\text{A2})$$

即可。这意味着, 以至少 $1 - \epsilon$ 的概率, 在 $B_n(C)$ 中存在着 $Q_n(\beta)$ 的一个局部极大值点; 由 $Q_n(\beta)$ 的严格凸性, 这个局部极大值点就是最大值点。因此, 最大值点 $\hat{\beta}_n$ 一定满足 $\|\hat{\beta}_n - \beta_0\| = O_p(n^{-1/2})$.

下面, 就来证明式 (A2) 成立。由于 $s_n(\beta_0)/\sqrt{n} = O_p(1)$ 且 $\nabla s_n(\beta_0)/n = I(\beta_0) + o_p(1)$. $\forall \beta \in \partial B_n(C)$, 利用对数偏似然函数的二阶泰勒展开, 有

$$\begin{aligned} &\frac{1}{n} \{ \ell_n(\beta_0 + n^{-1/2}\mu) - \ell_n(\beta_0) \} \\ &= \frac{1}{n} s_n^T(\beta_0) n^{-1/2}\mu - \frac{1}{2n} \mu^T \left\{ \frac{\nabla s_n(\beta_0)}{n} \right\} \mu + \frac{1}{n} \mu^T o_p(1) \mu \\ &= -\frac{1}{2n} \mu^T \{ I(\beta_0) + o_p(1) \} \mu + \frac{1}{n} O_p(1) \sum_{j=1}^p |\mu_j|, \end{aligned} \quad (\text{A3})$$

其中, $\mu = (\mu_1, \dots, \mu_p)^T$. 从而, 可以得到

$$\begin{aligned} D_n(\mu) &= \frac{1}{n} \{ Q_n(\beta_0 + n^{-1/2}\mu) - Q_n(\beta_0) \} \\ &= \frac{1}{n} \{ \ell_n(\beta_0 + n^{-1/2}\mu) - \ell_n(\beta_0) \} - \lambda_n \sum_{j=1}^p \left(\frac{|\beta_{j0} + n^{-1/2}\mu_j|}{|\tilde{\beta}_j|} - \frac{|\beta_{j0}|}{|\tilde{\beta}_j|} \right) \\ &\quad - \lambda_n^* \sum_j^p \sum_k^p \frac{w_{jk}}{2} \left[\left(\frac{\beta_{j0} + n^{-1/2}\mu_j}{\sqrt{d_j}} - \frac{\beta_{k0} + n^{-1/2}\mu_k}{\sqrt{d_k}} \right)^2 - \left(\frac{\beta_{j0}}{\sqrt{d_j}} - \frac{\beta_{k0}}{\sqrt{d_k}} \right)^2 \right] \\ &\leq \frac{1}{n} \{ \ell_n(\beta_0 + n^{-1/2}\mu) - \ell_n(\beta_0) \} - \lambda_n \sum_{j=1}^s \left(\frac{|\beta_{j0} + n^{-1/2}\mu_j|}{|\tilde{\beta}_j|} - \frac{|\beta_{j0}|}{|\tilde{\beta}_j|} \right) \end{aligned}$$

$$\begin{aligned}
 & -\lambda_n^* \sum_j^p \sum_k^p \frac{w_{jk}}{2} \left[\left(\frac{2\beta_{j0} + n^{-1/2}\mu_j}{\sqrt{d_j}} - \frac{2\beta_{k0} + n^{-1/2}\mu_k}{\sqrt{d_k}} \right) \left(\frac{\mu_j}{\sqrt{d_j}} \right. \right. \\
 & \quad \left. \left. - \frac{\mu_k}{\sqrt{d_k}} \right) n^{-1/2} \right] \\
 & \leq \frac{1}{n} \{ \ell_n(\beta_0 + n^{-1/2}\boldsymbol{\mu}) - \ell_n(\beta_0) \} + n^{-1/2} \lambda_n \sum_{j=1}^s |\mu_j| / |\tilde{\beta}_j| \\
 & \quad - n^{-1/2} \lambda_n^* \sum_j^p \sum_k^p w_{jk} \left(\frac{\beta_{j0}}{\sqrt{d_j}} - \frac{\beta_{k0}}{\sqrt{d_k}} \right) \left(\frac{\mu_j}{\sqrt{d_j}} - \frac{\mu_k}{\sqrt{d_k}} \right) \\
 & \quad - n^{-1} \lambda_n^* \sum_j^p \sum_k^p \frac{w_{jk}}{2} \left(\frac{\mu_j}{\sqrt{d_j}} - \frac{\mu_k}{\sqrt{d_k}} \right)^2. \tag{A4}
 \end{aligned}$$

下面，对 (A4) 的各项进行讨论。

1. 由式 (A3) 可以得到，(A4) 式的第一项

$$\frac{1}{n} \{ \ell_n(\beta_0 + n^{-1/2}\boldsymbol{\mu}) - \ell_n(\beta_0) \} = C^2 n^{-1} O_p(1) + C n^{-1} O_p(1),$$

2. 下面处理式 (A4) 第二项中的 $1/|\tilde{\beta}_j|$ 。注意到 $\tilde{\beta}$ 是一致估计量，因此它满足 $\|\tilde{\beta} - \beta_0\| = O_p(n^{-1/2})$ 。因此，对于 $1 \leq j \leq s$ ，由泰勒展开，

$$\frac{1}{|\tilde{\beta}_j|} = \frac{1}{|\beta_{j0}|} - \frac{\text{sgn}(\beta_{j0})}{\beta_{j0}^2} (\tilde{\beta}_j - \beta_{j0}) + o_p(|\tilde{\beta}_j - \beta_{j0}|) = \frac{1}{|\beta_{j0}|} + \frac{O_p(1)}{\sqrt{n}}.$$

此外，由于 $\sqrt{n}\lambda_n = O_p(1)$ ，因此 (A4) 式的第二项

$$\begin{aligned}
 n^{-1/2} \lambda_n \sum_{j=1}^s |\mu_j| / |\tilde{\beta}_j| &= n^{-1/2} \lambda_n \sum_{j=1}^s \left(\frac{|\mu_j|}{|\beta_{j0}|} + \frac{|\mu_j| O_p(1)}{\sqrt{n}} \right) \\
 &\leq C n^{-1/2} \lambda_n O_p(1) \\
 &= C n^{-1} (\sqrt{n} \lambda_n) O_p(1) = C n^{-1} O_p(1).
 \end{aligned}$$

3. 由于 $\sqrt{n}\lambda_n^* = O_p(1)$ ，因此 (A4) 式的第三项

$$\begin{aligned}
 & -n^{-1/2} \lambda_n^* \sum_j^p \sum_k^p w_{jk} \left(\frac{\beta_{j0}}{\sqrt{d_j}} - \frac{\beta_{k0}}{\sqrt{d_k}} \right) \left(\frac{\mu_j}{\sqrt{d_j}} - \frac{\mu_k}{\sqrt{d_k}} \right) \\
 & \leq C n^{-1/2} \lambda_n^* O_p(1) = C n^{-1} (\sqrt{n} \lambda_n^*) O_p(1) = C n^{-1} O_p(1).
 \end{aligned}$$

4. 由于 $\sqrt{n}\lambda_n^* = O_p(1)$ ，因此 (A4) 式的第四项

$$-n^{-1} \lambda_n^* \sum_j^p \sum_k^p \frac{w_{jk}}{2} \left(\frac{\mu_j}{\sqrt{d_j}} - \frac{\mu_k}{\sqrt{d_k}} \right)^2$$

$$\leq C^2 n^{-1} \lambda_n^* O_p(1) = C^2 n^{-3/2} (\sqrt{n} \lambda_n^*) O_p(1) = C^2 n^{-3/2} O_p(1).$$

综上，将 (A4) 各项的阶数进行比较。当 C 和 n 都很大时，有

$$C n^{-1} O_p(1) < C^2 n^{-1} O_p(1) \text{ 且 } C^2 n^{-3/2} O_p(1) < C^2 n^{-1} O_p(1)$$

因此，(A4) 中含有 $C^2 n^{-1} O_p(1)$ 的部分 $n^{-1} \{\ell_n(\beta_0 + n^{-1/2} \mu) - \ell_n(\beta_0)\}$ 占主导地位。从而 (A2) 成立，定理 2 得证。

(三) 定理 3 的证明

下面，对 Zhang 和 Lu (2007) 的相关部分进行修改，对定理 3 进行证明。

1. 稀疏性

下面证明 $\hat{\beta}_{2n} = 0$ 。要证明定理 3 的第 1 部分“稀疏性”，只需证明 $\forall \beta_1$ 满足 $\|\beta_1 - \beta_{10}\| = O_p(n^{-1/2})$ ，且 $\forall C$ ，

$$Q_n(\beta_1, \mathbf{0}) = \max_{\|\beta_2\| \leq C n^{-1/2}} Q_n(\beta_1, \beta_2)$$

即可。下面将会证明，以趋向于 1 的概率，在 $\beta_j \in (-C n^{-1/2}, C n^{-1/2})$ ， $j = s+1, \dots, p$ 中， $\partial Q(\beta)/\partial \beta_j$ 和 β_j 的符号相反。对于 β_0 某个邻域中的任意 β ，由泰勒展开，

$$\ell_n(\beta) = \ell_n(\beta_0) + n f(\beta) + O_p(\sqrt{n} \|\beta - \beta_0\|),$$

其中 $f(\beta) = -1/2(\beta - \beta_0)^T \{I(\beta_0) + o(1)\}(\beta - \beta_0)$ 。对于 $j = s+1, \dots, p$ ，有

$$\begin{aligned} \frac{\partial Q_n(\beta)}{\partial \beta_j} &= \frac{\partial \ell_n(\beta_0)}{\partial \beta_j} - n \lambda_n \frac{\text{sgn}(\beta_j)}{|\tilde{\beta}_j|} - n \lambda_n^* \beta_j - n \lambda_n^* \sum_{k=1}^p w_{jk} \frac{\beta_k}{\sqrt{d_j d_k}} \\ &= O(n^{1/2}) - n^{3/2} \lambda_n \frac{\text{sgn}(\beta_j)}{|n^{1/2} \tilde{\beta}_j|} - n \lambda_n^* \beta_j - n \lambda_n^* O_p(1) \\ &= O(n^{1/2}) - n^{3/2} \lambda_n \frac{\text{sgn}(\beta_j)}{O_p(1)} - n \lambda_n^* \beta_j - n \lambda_n^* O_p(1) \end{aligned}$$

最后一个等号成立，是由于 $\tilde{\beta}_j$ 为一致估计量，且 $|\tilde{\beta}_j - 0| = O_p(n^{-1/2})$ 。考虑定理 3 所给的条件 $n \lambda_n \rightarrow \infty$ ，且 $n \lambda_n^* \rightarrow 0$ 。因此，当 n 足够大时， $\partial Q_n(\beta)/\partial \beta_j$ 的符号完全由 $\text{sgn}(\beta_j)$ 决定，且他们的符号相反。从而定理 3 的第 1 部分得证。

2. 渐进正态性

下面证明 $\hat{\beta}_{1n}$ 的渐进正态性。根据定理 1，易得 $Q_n(\beta_1, \mathbf{0})$ 存在 \sqrt{n} 一致的极大值点 $\hat{\beta}_{1n}$ ，即

$$\left. \frac{\partial Q_n(\beta)}{\partial \beta_1} \right|_{\beta=\{\hat{\beta}_{1n}^T, \mathbf{0}^T\}^T} = \mathbf{0}.$$

令 $\mathbf{s}_{1n}(\beta)$ 为 $\mathbf{s}_n(\beta)$ 的前 q 个元素, 令 $\hat{\mathbf{I}}_{11}(\beta)$ 为 $\nabla \mathbf{s}_n(\beta)$ 的前 $q \times q$ 个子矩阵。则

$$\begin{aligned} \mathbf{0} &= \left. \frac{\partial Q_n(\beta)}{\partial \beta_1} \right|_{\beta=\{\hat{\beta}_{1n}^T, \mathbf{0}^T\}^T} \\ &= \left. \frac{\partial \ell_n(\beta)}{\partial \beta_1} \right|_{\beta=\{\hat{\beta}_{1n}^T, \mathbf{0}^T\}^T} - n\lambda_n \left(\frac{\text{sgn}(\hat{\beta}_{1n})}{|\tilde{\beta}_1|}, \dots, \frac{\text{sgn}(\hat{\beta}_{qn})}{|\tilde{\beta}_q|} \right)^T \\ &\quad - n\lambda_n^* \left(\hat{\beta}_1 + \sum_{j=1}^p w_{1j} \frac{\hat{\beta}_j}{\sqrt{d_1 d_j}}, \dots, \hat{\beta}_q + \sum_{j=1}^p w_{qj} \frac{\hat{\beta}_j}{\sqrt{d_q d_j}} \right)^T \\ &= \mathbf{s}_{1n}(\beta_0) - \hat{\mathbf{I}}_{11}(\beta^*) (\hat{\beta}_{1n} - \beta_{10}) - n\lambda_n \left(\frac{\text{sgn}(\beta_{10})}{|\tilde{\beta}_1|}, \dots, \frac{\text{sgn}(\beta_{q0})}{|\tilde{\beta}_q|} \right)^T \\ &\quad - n\lambda_n^* \left(\hat{\beta}_1 + \sum_{j=1}^p w_{1j} \frac{\hat{\beta}_j}{\sqrt{d_1 d_j}}, \dots, \hat{\beta}_q + \sum_{j=1}^p w_{qj} \frac{\hat{\beta}_j}{\sqrt{d_q d_j}} \right)^T, \end{aligned}$$

其中 β^* 在 β_0 和 $\hat{\beta}_n$ 之间。在最后一个等号中由于 $\hat{\beta}_n$ 是 \sqrt{n} 一致的估计量, 所以当 n 足够大时, $\text{sgn}(\hat{\beta}_{jn}) = \text{sgn}(\beta_{j0})$ 。根据 Andersen 和 Gill (1982) 的定理 3.2, 可以得到

$$\frac{\mathbf{s}_{1n}(\beta_0)}{\sqrt{n}} \xrightarrow{D} N(\mathbf{0}, \mathbf{I}_1(\beta_{10})) \text{ 且 } \frac{\hat{\mathbf{I}}_{11}(\beta^*)}{n} \xrightarrow{P} \mathbf{I}_1(\beta_{10}).$$

如果 $\sqrt{n}\lambda_n \rightarrow \lambda_0 \geq 0$, $\sqrt{n}\lambda_n^* \rightarrow 0$, 那么有

$$\sqrt{n}(\hat{\beta}_{1n} - \beta_{10}) = \mathbf{I}_1^{-1}(\beta_{10}) \left\{ \frac{1}{\sqrt{n}} \mathbf{s}_{1n}(\beta_0) - \lambda_0 \mathbf{b}_1 \right\} + o_p(1),$$

其中,

$$\mathbf{b}_1 = \left(\frac{\text{sgn}(\beta_{10})}{|\beta_{10}|}, \dots, \frac{\text{sgn}(\beta_{q0})}{|\beta_{q0}|} \right)^T,$$

此时对于 $1 \leq j \leq q$, $\tilde{\beta}_j \rightarrow \beta_{j0} \neq 0$ 。然后, 由 Slutsky 定理,

$$\sqrt{n}(\hat{\beta}_{1n} - \beta_{10}) \xrightarrow{D} N\{-\lambda_0 \mathbf{I}_1^{-1}(\beta_{10}) \mathbf{b}_1, \mathbf{I}_1^{-1}(\beta_{10})\}$$

特别地, 当 $\lambda_0 = 0$ 时,

$$\sqrt{n}(\hat{\beta}_{1n} - \beta_{10}) \xrightarrow{D} N\{\mathbf{0}, \mathbf{I}_1^{-1}(\beta_{10})\}.$$

从而定理 3 的第 2 部分得证。

附录 B

（一）模拟代码

囿于篇幅限制下面仅展示“阈值法”的模拟代码。“协方差法”和“相关系数法”类似。

```

1. ## Load the Packages
2. rm(list = ls())
3. library(MASS)
4. library(Matrix)
5. library(APML0)
6. library(Hmisc)
7.
8.
9. ## =====
10. ## PART I: Foundation
11. ## =====
12. q <- 27
13. n <- 100
14. mu <- rep(0,q)
15. Sigma <- matrix(0, nrow = q, ncol = q)
16.
17. ## ## Specify rho ##
18. rho <- 0.2
19. ## ## Specify epsilon ##
20. l_sigma <- 1
21.
22. ## The Covariance Matrix
23. for(k in 1:3)
24. {
25.   for(i in (9*k-8):(9*k))
26.   {
27.     for(j in (9*k-8):(9*k))
28.     {
29.       if(i==j) Sigma[i,j]=1
30.       else Sigma[i,j]=rho
31.     }
32.   }
33. }
34.
35. ## The Coefficients beta
36. beta.c <- sapply(c(1:9),function(x)(-1)^x*exp(-(x-1)/4))
37. beta <- c()
38. for(i in 1:9)
39. {
40.   beta <- c(beta, beta.c[i])
41.   beta <- c(beta, 0, 0)
42. }
43.
44.
45.
46. ## =====
47. ## PART II: GLASSO Encapsulation

```

```

48. ## =====
49.
50. ## GLASSO: GLASSO(0,0,0,0,0)
51. GLASSO <- function(CVM, TP, TN, CI, NN)
52. {
53.   ## -----
54.   ## (1.1) Generate Data
55.   ## -----
56.   ## The Design Matrix X
57.   X <- mvrnorm(n = n, mu = mu, Sigma = Sigma)
58.   ## epsilon
59.   e <- mvrnorm(n = n, mu = 0, Sigma = l_sigma)
60.   ## Survival Time
61.   Time <- exp(X %*% beta + e)
62.   ## Status
63.   status <- rbinom(n, 1, 0.7)
64.
65.   ## -----
66.   ## (1.2) The Graph Laplacian 'Omega'
67.   ## -----
68.   Omega <- cor(X)
69.   Omega[Omega < 0] <- 0
70.
71.   ## -----
72.   ## (1.3) Conduct the GLasso
73.   ## -----
74.   y <- cbind(Time, status)
75.   colnames(y) <- c("time", "status")
76.   alpha <- seq(0.1, 0.9, 0.1)
77.   cv <- c()
78.   for(i in 1:9)
79.   {
80.     fit.gLasso <- APML0(X, y, family = "cox",
81.                         penalty = "Net", Omega = Omega, alpha = alpha[i],
82.                         lambda = NULL, nlambdas = 50, rlambdas = 0.0001,
83.                         nfolds = 5, keep.betas = FALSE)
84.     cv <- as.numeric(c(cv, fit.gLasso$fit0[2]))
85.   }
86.   cv.min <- min(cv)
87.   i <- c(1:9)[cv==cv.min]
88.   fit.gLasso <- APML0(X, y, family = "cox",
89.                       penalty = "Net", Omega = Omega, alpha = alpha[i],
90.                       lambda = NULL, nlambdas = 50, rlambdas = 0.0001,
91.                       nfolds = 5, keep.betas = FALSE)
92.
93.   ## -----
94.   ## (1.4) Store the Performance
95.   ## -----
96.   ## index 1: cvm
97.   CVM <- c(CVM, as.numeric(fit.gLasso$fit0[2]))
98.   ## index 2: TP - Number of true positive covariates
99.   tp <- sum(beta > 0 & -fit.gLasso$Beta0 > 0)
100.  TP <- c(TP, tp)
101.  ## index 4: TN - Number of true negative covariates
102.  tn <- sum(beta < 0 & -fit.gLasso$Beta0 < 0)
103.  TN <- c(TN, tn)
104.  ## index 4: CI - Correspondence Index
105.  CI <- c(CI, as.numeric(1-
    rcorr.cens(X %*% fit.gLasso$Beta0, Surv(Time, status))[1]))

```

```

106. ## index 5: Number of non-zero coefficients
107. NN <- c(NN, sum(fit.gLasso$Beta0!=0))
108. return(list(CVM = CVM, TP = TP, TN = TN, CI = CI, NN = NN))
109.}
110.
111.
112.
113.## =====
114.## PART III: Repication of GLASSO
115.## =====
116.N <- 50
117.CVM <- c()
118.TP <- c()
119.TN <- c()
120.CI <- c()
121.NN <- c()
122.for(K in 1:N)
123.{
124.  fun.glasso <- GLASSO(CVM, TP, TN, CI, NN)
125.  CVM <- fun.glasso$CVM
126.  TP <- fun.glasso$TP
127.  TN <- fun.glasso$TN
128.  CI <- fun.glasso$CI
129.  NN <- fun.glasso$NN
130.}
131.## Record the CVM, TP, TN, CI, NN NOW!
132.mean(CVM)
133.mean(TP)
134.mean(TN)
135.mean(CI)
136.mean(NN)
137.
138.
139.
140.## =====
141.## PART IV: aGLASSO Encapsulation
142.## =====
143.
144.## aGLASSO: aGLASSO(wbeta, sgn, 0, 0, 0, 0, 0)
145.aGLASSO <- function(wbeta, sgn, CVM, TP, TN, CI, NN)
146.{
147.  ## -----
148.  ## (1.1) Generate Data
149.  ## -----
150.  ## The Design Matrix X
151.  X <- mvrnorm(n = n, mu = mu, Sigma = Sigma)
152.  ## epsilon
153.  e <- mvrnorm(n = n, mu = 0, Sigma = l_sigma)
154.  ## Survival Time
155.  Time <- exp(X %*% beta + e)
156.  ## Status
157.  status <- rbinom(n, 1, 0.7)
158.
159.  ## -----
160.  ## (1.2) The Graph Laplacian 'Omega'
161.  ## -----
162.  Omega <- cor(X)
163.  Omega[Omega < 0] <- 0
164.

```

```

165. ## -----
166. ## (1.3) Conduct the aGLasso
167. ## -----
168. y <- cbind(Time, status)
169. colnames(y) <- c("time", "status")
170. alpha <- seq(0.1, 0.9, 0.1)
171. cv <- c()
172. for(i in 1:9)
173. {
174.   fit.aGLasso <- APML0(X, y, family = "cox",
175.                         penalty = "Net", Omega = Omega, alpha = alpha[i],
176.                         lambda = NULL, nlambdas = 50, rlambdas = 0.0001,
177.                         wbeta = wbeta, sign = sign,
178.                         nfolds = 5, keep.beta = FALSE)
179.   cv <- as.numeric(c(cv, fit.aGLasso$fit0[2]))
180. }
181. cv.min <- min(cv)
182. i <- c(1:9)[cv==cv.min]
183. fit.aGLasso <- APML0(X, y, family = "cox",
184.                      penalty = "Net", Omega = Omega, alpha = alpha[i],
185.                      lambda = NULL, nlambdas = 50, rlambdas = 0.0001,
186.                      wbeta = wbeta, sign = sign,
187.                      nfolds = 5, keep.beta = FALSE)
188.
189. ## -----
190. ## (1.4) Store the Performance
191. ## -----
192. ## index 1: cvm
193. CVM <- c(CVM, as.numeric(fit.aGLasso$fit0[2]))
194. ## index 2: TP - Number of true positive covariates
195. tp <- sum(beta > 0 & -fit.aGLasso$Beta0 > 0)
196. TP <- c(TP, tp)
197. ## index 4: TN - Number of true negative covariates
198. tn <- sum(beta < 0 & -fit.aGLasso$Beta0 < 0)
199. TN <- c(TN, tn)
200. ## index 4: CI - Correspondence Index
201. CI <- c(CI, as.numeric(1-
  rcorr.cens(X %>% fit.aGLasso$Beta0, Surv(Time, status))[1]))
202. ## index 5: Number of non-zero coefficients
203. NN <- c(NN, sum(fit.aGLasso$Beta0!=0))
204. return(list(CVM = CVM, TP = TP, TN = TN, CI = CI, NN = NN))
205.}
206.
207.
208.
209.## =====
210.## PART V: Replication of aGLASSO
211.## =====
212.## Conduct the Preliminary Ridge Regression for aGLasso
213.## The Design Matrix X
214.set.seed(1213)
215.X <- mvrnorm(n = n, mu = mu, Sigma = Sigma)
216.## epsilon
217.set.seed(1213)
218.e <- mvrnorm(n = n, mu = 0, Sigma = l_sigma)
219.## Survival Time
220.Time <- exp(X %>% beta + e)
221.## Status
222.set.seed(1213)

```

```

223.status <- rbinom(n, 1, 0.7)
224.## Ridge Regression
225.y <- cbind(Time, status)
226.colnames(y) <- c("time", "status")
227.fit.pre <- APML0(X, y, family = "cox",
228.                penalty = "Enet", alpha = 0)
229.## Generate wbeta
230.beta.pre <- fit.pre$Beta[,50]
231.wbeta <- abs(1/beta.pre)
232.## Generate sgn
233.sgn <- rep(-1, 27)
234.sgn[beta.pre < 0] <- 1
235.
236.## -----
237.## START!
238.N <- 50
239.CVM <- c()
240.TP <- c()
241.TN <- c()
242.CI <- c()
243.NN <- c()
244.for(K in 1:N)
245.{
246.  fun.aglasso <- aGLASSO(wbeta, sgn, CVM, TP, TN, CI, NN)
247.  CVM <- fun.aglasso$CVM
248.  TP <- fun.aglasso$TP
249.  TN <- fun.aglasso$TN
250.  CI <- fun.aglasso$CI
251.  NN <- fun.aglasso$NN
252.}
253.## Record the CVM, TP, TN, CI, NN NOW!
254.mean(CVM)
255.mean(TP)
256.mean(TN)
257.mean(CI)
258.mean(NN)
259.
260.
261.
262.## =====
263.## PART VI: LASSO Encapsulation
264.## =====
265.
266.## LASSO: LASSO(0,0,0,0,0)
267.LASSO <- function(CVM, TP, TN, CI, NN)
268.{
269.  ## -----
270.  ## (1.1) Generate Data
271.  ## -----
272.  ## The Design Matrix X
273.  X <- mvrnorm(n = n, mu = mu, Sigma = Sigma)
274.  ## epsilon
275.  e <- mvrnorm(n = n, mu = 0, Sigma = l_sigma)
276.  ## Survival Time
277.  Time <- exp(X %*% beta + e)
278.  ## Status
279.  status <- rbinom(n, 1, 0.7)
280.
281.  ## -----

```



```

282. ## (1.2) Conduct the GLasso
283. ## -----
284. y <- cbind(Time, status)
285. colnames(y) <- c("time", "status")
286. fit.Lasso <- APLM0(X, y, family = "cox",
287.                   penalty = "Lasso",
288.                   lambda = NULL, nfolds = 5, keep.beta = FALSE)
289.
290. ## -----
291. ## (1.3) Store the Performance
292. ## -----
293. ## index 1: cvm
294. CVM <- c(CVM, as.numeric(fit.Lasso$fit0[2]))
295. ## index 2: TP - Number of true positive covariates
296. tp <- sum(beta > 0 & -fit.Lasso$Beta0 > 0)
297. TP <- c(TP, tp)
298. ## index 4: TN - Number of true negative covariates
299. tn <- sum(beta < 0 & -fit.Lasso$Beta0 < 0)
300. TN <- c(TN, tn)
301. ## index 4: CI - Correspondence Index
302. CI <- c(CI, as.numeric(1-
  rcorr.cens(X %%% fit.Lasso$Beta0, Surv(Time, status))[1]))
303. ## index 5: Number of non-zero coefficients
304. NN <- c(NN, sum(fit.Lasso$Beta0!=0))
305. return(list(CVM = CVM, TP = TP, TN = TN, CI = CI, NN = NN))
306.}
307.
308.
309.
310.## =====
311.## PART VII: Replication of LASSO
312.## =====
313.N <- 50
314.CVM <- c()
315.TP <- c()
316.TN <- c()
317.CI <- c()
318.NN <- c()
319.for(K in 1:N)
320.{
321.  fun.lasso <- LASSO(CVM, TP, TN, CI, NN)
322.  CVM <- fun.lasso$CVM
323.  TP <- fun.lasso$TP
324.  TN <- fun.lasso$TN
325.  CI <- fun.lasso$CI
326.  NN <- fun.lasso$NN
327.}
328.## Record the CVM, TP, TN, CI, NN NOW!
329.mean(CVM)
330.mean(TP)
331.mean(TN)
332.mean(CI)
333.mean(NN)
334.
335.
336.
337.## =====
338.## PART VIII: Enet Encapsulation
339.## =====

```

```

340.
341.## Enet: Enet(0,0,0,0,0)
342.Enet <- function(CVM, TP, TN, CI, NN)
343.{
344.  ## -----
345.  ## (1.1) Generate Data
346.  ## -----
347.  ## The Design Matrix X
348.  X <- mvrnorm(n = n, mu = mu, Sigma = Sigma)
349.  ## epsilon
350.  e <- mvrnorm(n = n, mu = 0, Sigma = l_sigma)
351.  ## Survival Time
352.  Time <- exp(X %*% beta + e)
353.  ## Status
354.  status <- rbinom(n, 1, 0.7)
355.
356.  ## -----
357.  ## (1.2) Conduct the Enet
358.  ## -----
359.  y <- cbind(Time, status)
360.  colnames(y) <- c("time", "status")
361.  alpha <- seq(0.1, 0.9, 0.1)
362.  cv <- c()
363.  for(i in 1:9)
364.  {
365.    fit.Enet <- APMLO(X, y, family = "cox",
366.                      penalty = "Enet", alpha = alpha[i],
367.                      lambda = NULL, nfolds = 5, keep.beta = FALSE)
368.    cv <- as.numeric(c(cv, fit.Enet$fit0[2]))
369.  }
370.  cv.min <- min(cv)
371.  i <- c(1:9)[cv==cv.min]
372.  fit.Enet <- APMLO(X, y, family = "cox",
373.                    penalty = "Enet", alpha = alpha[i],
374.                    lambda = NULL, nfolds = 5, keep.beta = FALSE)
375.
376.  ## index 1: cvm
377.  CVM <- c(CVM, as.numeric(fit.Enet$fit0[2]))
378.  ## index 2: TP - Number of true positive covariates
379.  tp <- sum(beta > 0 & -fit.Enet$Beta0 > 0)
380.  TP <- c(TP, tp)
381.  ## index 4: TN - Number of true negative covariates
382.  tn <- sum(beta < 0 & -fit.Enet$Beta0 < 0)
383.  TN <- c(TN, tn)
384.  ## index 4: CI - Correspondence Index
385.  CI <- c(CI, as.numeric(1-
    rcorr.cens(X %*% fit.Enet$Beta0, Surv(Time, status))[1]))
386.  ## index 5: Number of non-zero coefficients
387.  NN <- c(NN, sum(fit.Enet$Beta0!=0))
388.  return(list(CVM = CVM, TP = TP, TN = TN, CI = CI, NN = NN))
389.}
390.
391.
392.
393.## =====
394.## PART IX: Repication of Enet
395.## =====
396.N <- 50
397.CVM <- c()

```

```

398.TP <- c()
399.TN <- c()
400.CI <- c()
401.NN <- c()
402.for(K in 1:N)
403.{
404.  fun.Enet <- Enet(CVM, TP, TN, CI, NN)
405.  CVM <- fun.Enet$CVM
406.  TP <- fun.Enet$TP
407.  TN <- fun.Enet$TN
408.  CI <- fun.Enet$CI
409.  NN <- fun.Enet$NN
410.}
411.## Record the CVM, TP, TN, CI, NN NOW!
412.mean(CVM)
413.mean(TP)
414.mean(TN)
415.mean(CI)
416.mean(NN)

```

(二) 实证代码

```

1.  ## Load the Packages
2.  rm(list = ls())
3.  library(MASS)
4.  library(Matrix)
5.  library(APML0)
6.  library(Hmisc)
7.
8.  ## =====
9.  ## PART I: Load Data
10. ## =====
11. ## -----
12. ## (1.1) The Foundation
13. ## -----
14. raw.dat <- read.csv("Company.csv")
15. X <- raw.dat[,-c(1,2,3,4,5,6)]
16. q <- ncol(X)
17. n <- nrow(X)
18. y <- cbind(raw.dat$time, raw.dat$status)
19. colnames(y) <- c("time", "status")
20.
21.
22.
23. ## -----
24. ## (1.2) The Graph Laplacian 'Omega'
25. ## -----
26. Corr <- cor(X)
27. ## n=243, sqrt(n-3)*z ~ N(0,1)
28. ## set threshold c for sqrt(n-3)*z
29. c <- qnorm(1-0.001, 0, 1, lower.tail = T, log.p = F)
30. r <- (exp(2*c/sqrt(n-3))-1)/(exp(2*c/sqrt(n-3))+1)
31. Omega <- matrix(0,q,q)
32. Omega[abs(Corr) > r] <- 1
33. diag(Omega) <- rep(0, q)
34.
35.
36.

```

```

37. ## =====
38. ## PART II: Regression
39. ## =====
40. ## -----
41. ## (2.1) Conduct the Preliminary Ridge Regression
42. ## -----
43. set.seed(1213)
44. fit.pre <- APML0(as.matrix(X), y, family = "cox",
45.                 penalty = "Enet", alpha = 0)
46. beta.pre <- fit.pre$Beta[,50]
47. wbeta <- abs(1/beta.pre)
48. sgn <- rep(-1, q)
49. sgn[beta.pre < 0] <- 1
50.
51.
52.
53. ## -----
54. ## (2.2) Conduct the aGLasso
55. ## -----
56. alpha <- seq(0.1, 0.9, 0.01)
57. cv <- c()
58. for(i in 1:81)
59. {
60.   set.seed(1213)
61.   fit.apLasso <- APML0(as.matrix(X), y, family = "cox",
62.                       penalty = "Net", Omega = Omega, alpha = alpha[i],
63.                       lambda = NULL, nlambda = 50, rlambd = 0.0001,
64.                       wbeta = wbeta, sgn = sgn,
65.                       nfolds = 5, keep.beta = FALSE)
66.   cv <- as.numeric(c(cv, fit.apLasso$fit0[2]))
67. }
68. cv.min <- min(cv)
69. i <- c(1:81)[cv==cv.min]
70. # [1] 7
71. set.seed(1213)
72. fit.apLasso <- APML0(as.matrix(X), y, family = "cox",
73.                     penalty = "Net", Omega = Omega, alpha = alpha[i],
74.                     lambda = NULL, nlambda = 50, rlambd = 0.0001,
75.                     wbeta = wbeta, sgn = sgn,
76.                     nfolds = 5, keep.beta = FALSE)
77.
78.
79.
80. ## =====
81. ## PART III: Results Analysis
82. ## =====
83. ## -----
84. ## (3.1) The Coefficients
85. ## -----
86. fit.apLasso$Beta0
87.
88. ## Hazard Ratio
89. exp(fit.apLasso$Beta0)
90.
91.
92.
93. ## -----
94. ## (3.2) Other indexes
95. ## -----

```

```

96. ## lambda
97. fit.apLasso$fit0[1]
98.
99. ## alpha
100.alpha[i]
101.
102.## CVM
103.fit.apLasso$fit0[2]
104.
105.## C-index
106.1-
      rcorr.cens(as.matrix(X) %%% fit.apLasso$Beta0, Surv(raw.dat$time, raw.dat$status))[1
    ]
107.
108.## #(Non-Zero Variables)
109.sum(fit.apLasso$Beta0!=0)
110.
111.
112.
113.## -----
114.## (3.3) Significance Test
115.## -----
116.## Preparation
117.index <- (1:q)[fit.apLasso$Beta0!=0]
118.len <- length(index)
119.beta.n <- fit.apLasso$Beta0[fit.apLasso$Beta0!=0]
120.Xn <- as.matrix(X[, index])
121.
122.
123.## Global Wald Test
124.I <- matrix(rep(0, len * len), len)
125.for(u in 1:len)
126.{
127.  for(v in 1:len)
128.  {
129.    ## working on I[u,v]
130.    for(i in 1:n)
131.    {
132.      if(status[i]==1)
133.      {
134.        nu <- sum((Time >= Time[i]) * Xn[,u] * Xn[,v] * exp(Xn %%% beta.n))
135.        de <- sum((Time >= Time[i]) * exp(Xn %%% beta.n))
136.        I[u,v] <- I[u,v] + nu/de
137.      }
138.    }
139.  }
140.}
141.## Evaluate
142.qChi <- as.numeric(t(beta.n) %%% I %%% beta.n)
143.qChi
144.pchisq(qChi, len, ncp=0, lower.tail = F, log.p = F)
145.
146.
147.## Local Wald Tests
148.## Write a function
149.Wald.test <- function(k)
150.{
151.  ## Prepare
152.  beta.k <- beta.n

```

```

153. beta.k[k] <- 0
154. ## Start!
155. I <- matrix(rep(0, len * len), len)
156. for(u in 1:len)
157. {
158.   for(v in 1:len)
159.   {
160.     ## working on I[u,v]
161.     for(i in 1:n)
162.     {
163.       if(status[i]==1)
164.       {
165.         nu <- sum((Time >= Time[i]) * Xn[,u] * Xn[,v] * exp(Xn %%% beta.k))
166.         de <- sum((Time >= Time[i]) * exp(Xn %%% beta.k))
167.         I[u,v] <- I[u,v] + nu/de
168.       }
169.     }
170.   }
171. }
172. ## Evaluate
173. qChi <- t(beta.n[k]-0) %%% solve(solve(I)[k,k]) %%% (beta.n[k]-0)
174. p.value <- pchisq(qChi, 1, ncp=0, lower.tail = F, log.p = F)
175. ## return
176. return(list(Chisq = qChi, p.value = p.value))
177.}
178.
179.for(k in 1:10)
180.{
181.  Tst <- Wald.test(k)
182.  print(Tst$p.value)
183.}
184.
185.
186.
187.## =====
188.## PART IV: Network Analysis
189.## =====
190.## (4.1) Correlation Network
191.Corr.gephi <- abs(Corr)
192.Corr.gephi[Corr.gephi==1] <- 0
193.write.table(Corr.gephi,"Corr.csv", row.names=FALSE, col.names=FALSE,sep=",")
194.
195.## (4.2) Adfacency Matrix
196.write.table(Omega,"Omega.csv", row.names=FALSE, col.names=FALSE,sep=",")
197.
198.## (4.3) Boxes
199.## Now start loop!
200.alpha <- c(0.2, 0.5, 0.7, 0.9)
201.lambda <- c(0.1, 0.01, 0.001, 0.0001)
202.i=4
203.j=1
204.fit.gephi <- APMLO(as.matrix(X), y, family = "cox",
205.                    penalty = "Net", Omega = Omega, wbeta = wbeta, sgn = sgn,
206.                    alpha = alpha[i], lambda = lambda[j])
207.(1:q)[as.numeric(fit.gephi$Beta)!=0]
208.
209.
210.
211.## =====

```

```

212.## PART V: Dynamic Projection
213.## =====
214.## Calculate H0(t)
215.Time <- raw.dat$time
216.status <- raw.dat$status
217.t <- sort(Time)
218.t <- unique(t)
219.H0 <- c()
220.for(i in 1:length(t))
221.{
222.  term <- 0
223.  for(j in 1:n)
224.  {
225.    if(Time[j] <= t[i] & status[j]==1) # Ti[j]=='ti', t[i]=='t'
226.    {
227.      term <- term + 1/sum(exp(as.matrix(X) %%% fit.apLasso$Beta0) * as.numeric(Time
        >=Time[j]))
228.    }
229.  }
230.  H0 <- c(H0, term)
231.}
232.## Calculate S0(t)=exp(-H0(t))
233.S0 <- exp(-H0)
234.## Save S0
235.write.table(data.frame(t,S0),"S0.csv", row.names=FALSE, col.names=FALSE,sep=",")
236.
237.## Calcualte S(t|X)
238.S.t.X <- c()
239.for(i in 1:length(S0))
240.{
241.  S.t.X <- cbind(S.t.X, S0[i]^as.numeric(exp(as.matrix(X) %%% fit.apLasso$Beta0)))
242.}
243.
244.S.t.X.df <- data.frame(rbind(t, S.t.X))
245.row.names(S.t.X.df) <- c("time", as.character(raw.dat$证券代码))
246.write.table(S.t.X.df, "S_t_X.csv", row.names=TRUE, col.names=FALSE,sep=",")
247.
248.
249.
250.## Reconstruct the dataset
251.S1 <- t(S.t.X)
252.S2 <- c(as.numeric(S1[,74]), as.numeric(S1[,84]), as.numeric(S1[,191]),
253.  as.numeric(S1[,233]), as.numeric(S1[,241]))
254.S2 <- cbind(rep(t, 5), S2)
255.S2 <- cbind(S2, c(rep(1, length(t)), rep(2, length(t)), rep(3, length(t)),
256.  rep(4, length(t)), rep(5, length(t)) ))
257.S2 <- data.frame(S2)
258.colnames(S2) <- c("Time", "SP", "Company")
259.S2$Company <- as.factor(S2$Company)
260.levels(S2$Company) <- c('中国一重','贝因美','美的集团','惠达卫浴','中宠股份')
261.
262.## Plot
263.ggplot(S2, aes(Time, SP, color = Company)) +
264.  geom_step(size = 1.5, linetype = 1) +
265.  labs(title="Survival Curves", y = "Survival Probability") +
266.  scale_x_continuous(breaks = c(1000,2000,3000,4000,5000,6000,7000)) +
267.  scale_y_continuous(breaks = c(0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0)) +
268.  theme_bw() +
269.  theme(panel.grid = element_blank(), panel.border = element_blank(),

```

```
270.         axis.line = element_line(size = 1, colour="black"))
271.
272.
273.## Calculate  $S(t|\bar{X})$ 
274.a=t(apply(X,2,mean)) %*% fit.apLasso$Beta0
```