

南京审计大学硕士研究生毕业论文  
—— $L_1$  范数主成分分析及其在宏观经济因子模型中的应用

蒯强

2021 年 3 月 3 日

## 目录

第一章 绪论	1
§1.1 研究背景及意义	1
§1.2 基本概念和研究现状综述	2
1.2.1 因子模型及其在宏观经济中的应用	2
1.2.2 $L_1$ 范数主成分分析	3
§1.3 研究创新	4
§1.4 本文结构	5
第二章 基于 $L_1$ 范数主成分分析的高维因子模型估计	6
§2.1 因子模型及其估计	6
2.1.1 正交因子模型	6
2.1.2 动态因子模型	7
2.1.3 近似因子模型	7
§2.2 $L_2$ 范数主成分分析	7
§2.3 $L_1$ 范数主成分分析	9
2.3.1 求解 $L_1$ 范数主成分分析的交替凸优化算法	9
2.3.2 算法步骤	10
§2.4 稳健性实验	10
2.4.1 数据准备	11
2.4.2 实验结果	11
§2.5 基于国内主要月度宏观经济数据的实证研究	12
2.5.1 扩散指数模型	13
2.5.2 因子个数判定	13
2.5.3 数据说明	13
2.5.4 实证结果	14
§2.6 本章小结	16
第三章 最小绝对值回归的性能研究	17
§3.1 简介	17
3.1.1 最小绝对值回归的稳健性	17
3.1.2 最小绝对值回归的估计方法	18
§3.2 聚类——迭代拆解算法	19
3.2.1 聚类——迭代拆解算法说明	19
3.2.2 优化最小绝对值回归	20
3.2.3 优化带有惩罚项的最小绝对值回归	21
§3.3 一种基于替代变量的估计方法	21
3.3.1 基于替代变量的迭代算法	22
3.3.2 优化带有惩罚项的最小绝对值回归	24
§3.4 模拟实验	24

3.4.1	数据准备	25
3.4.2	实验结果	25
3.4.3	结论	28
§3.5	SVN-AID 算法	28
3.5.1	算法步骤	28
3.5.2	数值模拟实验	29
§3.6	本章小结	29
第四章	近似因子模型的 $L_1$ 范数主成分估计算法研究	30
§4.1	改进的 $L_1$ 主成分分析交替凸优化算法	30
4.1.1	基于替代变量算法的交替凸优化算法改进	30
4.1.2	基于聚类——迭代拆解算法的交替凸优化算法改进	30
4.1.3	数值模拟	30
4.1.4	高频金融数据的主成分分析	30
§4.2	求解特征空间 $L_1$ 范数最大化的贪婪算法	30
4.2.1	求解子问题	30
4.2.2	贪婪策略求解主成分	31
4.2.3	模拟与实证	31
§4.3	本章小结	31
参考文献		32

## 第一章 绪论

本章阐述了本篇论文的研究背景和研究意义，综述了国内外学者在相关领域的研究现状、主要的成果、结论和观点。由此展开，叙述了论文的主要研究内容、研究所使用的技术路线、本文的主要创新点和行文的结构。

### §1.1 研究背景及意义

在宏观经济学理论中，政府是指导和调控经济运行的主体。国民经济的发展和政府的指导和调控紧密相关，政府部门需要时刻把握国民经济的方方面面，不仅仅要关心主要产业 GDP 的增长，还要关心经济中的通货膨胀、利率和汇率等等许多复杂的经济变量，并且需要研究这些经济变量发生变化的原因以及各种经济变量之间的作用关系。只有这样，政府才能发现经济中存在的问题，并且给出针对性的指导和调控手段。

在古典经济学理论中，对于国民经济作为一个整体的活动情况一般不予考察，古典经济学往往从围观主体出发来解释观察到的宏观经济问题。然而，在市场经济高度发展的当代，市场规模空前扩大，对人类社会已经产生了深远影响。宏观经济的发展已经密切关系到民生发展，宏观经济的动荡也通过金融危机和失业直接关系到社会稳定。特别是一战后的大萧条使得古典经济学对宏观经济的分析进入了困境，在凯恩斯理论的影响下，现代宏观经济学进入主流。在现代宏观经济学的主张下，政府应该采取随机应变的宏观经济政策，通过财政和货币手段对国民经济发展进行调控和指导。

政府管理国民经济的关键是制定正确的经济政策，而做决策的前提就是预测。因此制定经济政策不仅需要各级政府对经济发展的当前状态有充分把握，并且能够对经济的未来变化有一定的预测能力。经济预测是根据经济发展的时间和空间纬度的综合信息来试图研究经济变化趋势的技术。政府为了提高经济政策的有效性，就需要通过更加准确的预测来分析和判断信息，因此政府部门需要不断追求更加准确的经济预测技术，才能在复杂的经济形式变化中掌握主动权。

除了经济发展的需要，为了规避国民经济的风险，把控我国宏观经济安全大局，也对政府更加准确地把握和预测国民经济提出了要求。当前我国正处于经济和社会向前发展的关键时期，经济虽然多年持续高速发展，但是，经济结构发展不均衡的现象逐渐显现，社会经济系统的复杂性也日渐增强。这就更需要对经济系统进行实时准确的检测和对未来经济形势准确预测，提前做好准备工作，防范经济增长的重大风险。经济风险不仅仅和宏观经济运行状况紧密关联，同样和国计民生密不可分，不管是通过对经济系统的宏观把控还是对经济运行中某些指标的局部观察，我们都可以了解经济运行的一些趋势，并且通过对经济中的某些重要运行趋势进行预测，从而规避经济风险，稳定宏观经济安全大局。把控宏观经济安全大局对于国民经济平稳快速发展、改善国计民生、促进经济又好又快发展、促进国家安全稳定有重要的战略和实际意义。习近平总书记于 2014 年首次提出总体国家安全观，并构建了集政治安全、国土安全、军事安全、经济安全和文化安全等领域为一体的国家安全体系。其中，经济安全是国家安全体系的重要组成部分，是国家安全的基础。宏观经济安全问题需研究 1 个或多个指标（如经济增长量）与其他指标（如资本、劳动力、人口和技术等）相互间的关系，从而揭示经济现象背后的经济规律，发现经济运行风险点，制定经济政策。为了避免宏观经济中的冲击带来的风险，需要对宏观经济总量进行实时预测和短期预测来减少不确定性，为宏观经济调控政策提供决策支持。通过对宏观经济指标进行有效监测和预测，并提前识别和防范相关风险，对保证国民经济稳步增长具有重要意义。

宏观经济预测方法繁多，有的经济学家使用经济学方程组来给宏观经济系统建立模型，从而通过模型来对某些宏观经济指标进行预测并且给出经济学分析。然而国民经济是一个非常复杂的整体，它是一

个内部极其复杂的系统，并且经常伴随难以预测的外部冲击，对于复杂的经济变量，想到做到准确预测往往非常艰难。虽然现在传统的经济学理论和现代计算机技术的结合下，已经发展出了许多预测方法，并且学者们仍在不断利用新的理论和技术来完善这些已有的方法。然而，在我国国情下，进行准确的经济预测仍然非常困难。现代经济系统极为复杂，意味着使用这种方法我们就可能需要建立非常复杂的经济分析模型，有可能导致分析困难，并且往往预测结果和真实的经济运行结果有较大出入。因为系统过于复杂，学者们就开始使用基于历史数据进行向前预测的经验方法，这种方法大量应用数据和数据分析、大数据技术和统计学知识，并且融合了成熟的经济学理论和模型，往往能够起到不错的效果。在现代经济系统中，经济社会中每时每刻都在产生大量的数据，这些海量的经济活动数据给经济预测提供了丰富的数据支撑，如何有效地利用大数据时代丰富的数据资源进行宏观经济现时预测已成为一个重要的研究课题。当今的宏观经济数据具有高维度、数量多、处理困难等特征。结合计算机科学和统计学等方面的技术和理论知识，我们能够采取一些方法来操纵数据，从中过挖掘提炼出有效的信息。关于这个方面的研究正在火热进行，人们不断尝试使用更新的技术来处理数据，提取信息，希望利用高维度下丰富多面的经济信息来对未来的经济发展趋势作较为准确的分析。

由于宏观经济变量往往具有维数很高、分布重尾、数据集不完整等特征，导致在宏观经济的数量研究中遇到稳健性问题。例如线性回归模型的估计、主成分分析、因子模型的估计等场景下，基于最小二乘法的经典方法往往难以得到满意的结果。因此，在宏观经济数据的分析和预测中往往还需要引入更加稳健的方法。使用更加稳健的基于  $L_1$  范数的方法来代替基于  $L_2$  范数的方法例如使用最小绝对值回归代替最小二乘法，是一个很常见的做法。

基于  $L_1$  范数的稳健方法在宏观经济预测中已经得到了一定的应用，例如最小绝对值回归等。高维宏观经济数据可以通过因子模型来分析和处理，得到一些对于经济预测很有价值的信息，然而由于经济收到外部冲击等原因，许多宏观经济指标不能够服从正态分布，往往在分布上呈现重尾特征。这就影响了采用普通主成分分析方法估计因子模型，然而然而在该领域相关问题还有待进一步研究。

## §1.2 基本概念和研究现状综述

本文的研究内容是  $L_1$  范数主成分分析及其在宏观经济因子模型中的应用。因此主要围绕近似因子模型和  $L_1$  范数主成分分析两大主题进行了文献的搜集和整理，下面简要介绍它们的基本概念，并梳理了关于这两个主题国内外学者在相关领域的研究现状和存在问题。

### 1.2.1 因子模型及其在宏观经济中的应用

因子模型最早由英国心理学家 Charles Spearman 提出，最早用于智力测验得分数据的分析中。提出因子模型是为了使用少量潜在的、不可观测的因子来描述多个相依变量间的关系。假设一组变量能够用因子模型来描述，那么我们就把这组变量称为观测变量，因为它们的值可以实际测度，而观测变量的值是由一组不可观测的因子的线性组合决定的。

经典因子模型提出后，迅速在心理学、社会学等多个学科内得到了广泛的使用。然而因子模型在宏观经济研究领域的应用出现较晚，这可能是因为经典因子模型主要用于处理截面数据，而宏观经济研究中主要研究的数据为时间序列数据。然而近年来因子模型的理论和应用已经得到了很大的完善和发展，作为一种重要分析工具因子模型已经成为计量经济学的必修内容，并且动态因子模型常被应用于宏观经济政策的评估、经济指数的构建和经济指标的预测中。

因子模型在宏观经济研究中的广泛使用和当今宏观经济数据集的特征密切相关。从政府部门和一些

经济研究机构公布的数据和报告可以发现，在制定宏观经济政策时，政策制定者在决策过程中考虑到了非常多的经济变量。一个简单的经济政策背后可能需要分析成百上千不同的经济指标，并且这些经济指标的可靠数据记录往往较少，许多宏观经济指标往往只有 30 年以内的可靠记录。因此宏观经济研究中面对的数据较为特殊，常用的计量经济模型例如向量自回归模型等在这种数据集下很难可靠地估计模型系数。在建立模型时将所有的经济指标都进行研究，大大提升了问题的计算复杂度，难以快速给出分析结果。而因子模型可从高维数据集中提炼出少量公共因子，这些因子中包含了经济运行的许多有效信息，因而可以通过分析这些公共因子准确了解宏观经济的运行状况。

经典因子模型适用于截面数据，对于分析经济时间序列而言不十分适用，因此国内外学者们在经典因子模型上发展出了动态因子模型。Geweke[15] 与 Sargent[26] 等人于 1977 年分别提出了动态因子模型，动态因子模型是经典因子模型在时间序列数据上的延伸，它提供了从维数众多的经济时间序列数据中提取公共因子来研究和解释经济波动的手段。一般认为动态因子模型提出基于以下观点：宏观经济具有周期性的波动是通过一系列经济变量的活动来传递和扩散的，任何一个经济变量本身的波动都不足以代表宏观经济的整体波动。对于经济波动的研究需要从多个具有相依性的经济变量同时着手。因此，需要从一国许多经济时间序列数据中估计出驱动各变量波动的共同动态因子，并且给出解释。动态因子模型已经成为判别和分析经济周期波动的有效工具。

动态因子模型主要应用于仅包含几个总体宏观经济变量的小型数据集，这主要是由于经济学家们假定影响所有变量的普遍性结构冲击是经济变量协同变动的原因。但是在包含了成百上千个变量的高维宏观经济数据集中，常常存在某个仅影响某一组变量的部门冲击或是局部冲击，若根据经济理论将这些非普遍性冲击归于异质性部分，那么就可能造成异质性部分的截面相关，这又将违背经典因子模型中对于异质性部分相互正交的假定。Chamberlain 和 Rothschild 于 1983 年提出应该放松关于异质性部分协方差矩阵为对角阵的假定，即允许其为非对角阵，从而发展出了近似动态因子模型 [5]。在众多因子模型及其拓展模型中，近似动态因子模型得到了最为广泛的使用。

Forni 等人于 2000 年提出应该在近似动态因子模型的基础上，进一步放松假设，即允许观测变量收到因子的无限之后项的影响，从而发展出了广义动态因子模型 [12]。但是在模型的参数估计方面，广义动态因子模型只能使用频域方法估计因此其使用范围比较有限。

因子模型在宏观经济分析中有两个最主要的应用：1) 构建指数。最早的尝试是 Stock 和 Watson 等于 1989 年提出的一致和先行景气指数 [27]。另一个例子是基于 Stock 和 Wasto 在 1999 年提出的方法 [6]，构建了著名的 CFNAI 指数，它就基于美国的 85 个月度宏观经济数据。2) 预测。Stock 和 Watson 于 2002 年提出了扩散指数模型 [31]，使用因子的主成分估计量进行预测。通过少量的因子，不仅利用了大量的信息而且避免了变量过多带来的模型自由度问题。Eickmeier 等于 2008 年进行的实证研究 [10]，使用美国宏观经济数据证明了使用扩散指数模型的预测效果要好于大部分其他方法。

目前在国外已经存在大量关于使用因子模型的进行宏观经济预测的实证研究 (Luciani, 2014[19])，而目前在国内使用因子模型进行宏观预测的实证研究较少 (高华川和张晓峒, 2015[33])。

### 1.2.2 $L_1$ 范数主成分分析

主成分分析是一种应用广泛的数据降维技术。通过选择高维数据在特征空间中的数个基向量构成投影，这些基向量代表了高维数据在特征空间的投影方差最大的几个方向。对应到优化理论，传统主成分分析算法解决  $L_2$  范数优化问题，因而也称为  $L_2$  范数主成分分析。

而面对噪声干扰高的数据， $L_2$  范数主成分分析不具有稳健性。而在优化理论中， $L_1$  范数的优化问题对噪声不敏感，具有很好的稳健性。若将  $L_1$  范数应用于主成分分析中，这种方法就称为  $L_1$  主成分分析，

在理论上它具有很好的稳健性。因此，国内外学者从很早就开始了对  $L_1$  范数主成分分析的研究。众多应用性的研究（Galpin and Hawkins, 1987[13]；Baccini et al., 1996[1]；Ke and Kanade, 2003[16]；Ding et al., 2006[9]；Choulakian, 2006[8]；Gao, 2008[14]；Kwak, 2008[18]）已经表明， $L_1$  范数主成分分析具有很强的抗噪声能力，应用于机器视觉 [17]、人脸识别 [18] 等领域有非常好的效果。

然而  $L_1$  范数主成分分析在求解上比  $L_2$  范数主成分分析困难许多，具体体现在难以获得全局最优解、计算复杂度高等方面。 $L_1$  主成分分析有两种问题形式，一种是最小化重构误差的  $L_1$  范数；另一种是最大化特征空间投影的  $L_1$  的范数。

对于前者，一个具有代表性的早期研究 [1] 提出了一种基于典型相关分析的算法，该算法具有一定的启发意义；Ke 和 Kanade 等人于 2005 年提出了一种方法 [17]，该算法将全局的非凸优化问题使用一种交替的凸优化算法解决，每次交替求解的问题可以获得全局最优，每一次交替中需要求解若干个线性规划问题；Gao 于 2008 年提出一种在 Laplace 噪声下估计最优  $L_1$  特征空间的 Bayesian 方法 [14]；而 Brooks 等人于 2013 年提出一种算法 [4]，该算法逐次降低优化问题的维数，递进地寻找从  $m$  特征维空间向  $m-1$  特征维空间的最佳投射，而获得这样的最佳投射需要求解一系列最小绝对值回归问题；Park 等人于 2016 年提出一种算法 [25]，在迭代中使用一种再加权最小二乘法求问题的近似解。而对于该问题，至今没有一种算法能够给出全局最优解。

对于后者，Choulakian 于 2006 年对该问题进行了完整描述 [8]，基于协方差矩阵的  $L_1$  估计 [13] 来求解；比较具有代表性的是 Kwak 于 2008 年提出的一种贪婪算法 [18]，该算法在给出前  $k$  个主成分的局部最优估计下，求解第  $k+1$  个主成分的最优解；Nie 等人于 2011 年扩展了该算法，并给出一种采用非贪婪策略的实现方式 [23]；Markopoulos 等人于 2014 年给出了一种在特征维数已知情况下可获得全局最优解的算法 [20]，并且证明了在原始数据维数和样本数一起趋于无穷时，本问题是一个 NP 难题，另外该结论同样被 McCoy[22] 等人证明；Park 等人于 2020 年提出了一种基于聚类——迭代拆解的解法 [24]，在没有进行理论论证的情况下进行了数值模拟，实验结果表明了该算法在性能上有所改善。

### §1.3 研究创新

1) 近似因子模型在宏观经济研究中被广泛使用，其因子的主成分估计量在宏观经济预测中能够起到很好的效果。然而，传统主成分分析本质上求解是  $L_2$  范数优化问题，面对高维重尾宏观经济变量在数据处理上不具有稳健性。 $L_1$  范数主成分分析作为  $L_2$  主成分分析的一种稳健替代手段，在信号处理、机器视觉和人脸识别中已经有了大量应用。本文创新性地采用  $L_1$  范数主成分分析作为近似因子模型的估计，来代替  $L_2$  主成分分析。采用  $L_1$  范数主成分分析估计得到的因子和  $L_2$  是不同的，本文通过基于国内高维月度宏观经济数据集的实证研究，比较了两种不同因子在宏观经济预测中的表现，论证了将  $L_1$  主成分分析作为近似因子模型估计来进行宏观经济预测的可行性。

2) 并且  $L_1$  范数在宏观经济预测中的一大应用是采用最小绝对值回归建立线性模型，而该模型求解采用的线性规划方法在当今数据维数越来越高、数据量越来越大的场合下显得性能不足，因而限制了最小绝对值回归在这一领域的应用。本文阐述了两种适用于最小绝对值回归性能优化的最新算法，它们在应对大规模和高维数据集上较传统线性规划方法有着更好的性能表现，本文通过数值模拟实验测试和分析了两种算法的优缺点，并且给出其在实际应用中的一些建议。并在两种算法的基础上，融合彼此的优点提出了一种新的算法，通过模拟实验说明，该算法在一定程度上同样可以提升最小绝对值回归的计算性能。

3) 本文对  $L_1$  范数主成分分析的算法根据目标问题不同分，可以有两种不同的解法。本文分别介绍了一种求解重构误差最小化的交替凸优化算法和一种求解特征空间  $L_1$  范数最大化的算法，在国内月度

宏观经济数据集上使用两种算法分别进行了近似因子估计，结果表明两者同样适用，并给出一些应用建议。最后，本文结合对最小绝对值回归性能的研究，改进了交替凸优化算法，使其在估计高维近似因子模型时拥有更好的性能。

### §1.4 本文结构

本文主要内容分为 5 个章节：

第一章为绪论。该章节介绍了本文的写作背景和课题的研究意义，并且分析了课题相关的理论和应用研究的进展和不足，指出了本文写作的创新点。

第二章为基于  $L_1$  范数主成分分析的高维因子模型估计。该章节介绍了因子模型的基本概念，提出了将  $L_1$  范数主成分分析引入近似因子模型的估计，介绍了一种求解  $L_1$  范数主成分分析的交替凸优化算法并给出了具体步骤。接下来，通过模拟实验探究  $L_1$  范数主成分分析相比于传统基于  $L_2$  范数的主成分分析的稳健性。最后，通过一个基于国内月度高维宏观经济数据的实证研究，来验证  $L_1$  主成分分析能否应用在宏观经济预测中。

第三章为高维最小绝对值回归的性能研究。该章节介绍了最小绝对值回归，它是稳健  $L_1$  范数在宏观经济建模中的另一个主要应用。我们说明了最小绝对值回归的稳健性，介绍了其基本求解方法，并针对其求解性能问题进行了研究。这里介绍了两种适用于高维最小绝对值回归的最新算法：聚类——迭代拆解算法和基于替代变量的估计方法，详细阐述了两种算法的基本思想和具体步骤。通过模拟高维宏观经济数据的数值实验，测试了算法的优缺点，并给出建议。本章的研究有助于我们改进  $L_1$  主成分分析的交替凸优化算法的求解性能。

第四章为  $L_1$  范数主成分分析求解算法。第二章已经将  $L_1$  范数主成分分析引入了近似因子模型的估计，并探究其应用于经济预测中的可行性。这一章将进一步讨论  $L_1$  范数主成分分析，较深入地研究其求解问题。这里首先介绍了一种解决低维投影  $L_1$  范数最大化问题的求解算法，它和前面的交替凸优化算法求解  $L_1$  主成分分析的角度有所不同。接下来，利用第三章的研究成果对交替凸优化算法的求解性能做出了优化。我们进行了数值模拟实验，和人脸图像去噪实验，来测试算法的表现。最后进行了基于国内月度宏观经济数据的实证研究，并对实际应用中  $L_1$  算法的选择提出了建议。

第五章为结论与展望。该章节总结了本文的主要研究结论和存在的问题，并为进一步的研究提出了方向。



## 第二章 基于 $L_1$ 范数主成分分析的高维因子模型估计

### §2.1 因子模型及其估计

因子模型是主成分分析的推广和发展，它也是多元统计中广泛使用的一种降维方法。因子模型研究相关矩阵或者协方差矩阵的内部依赖关系，它将多个变量综合为少数几个因子，以再现原始变量和因子之间的关系。

#### 2.1.1 正交因子模型

设  $\mathbf{X} = (X_1, X_2, \dots, X_p)^T$  是可观测的随机向量， $\mathbb{E}\mathbf{X} = \boldsymbol{\mu}$ ， $\mathbb{D}(\mathbf{X}) = \boldsymbol{\Sigma}$ ，设  $\mathbf{f} = (f_1, f_2, \dots, f_m)^T$  ( $m < p$ ) 为不可观测的随机向量，且  $\mathbb{E}\mathbf{f} = \mathbf{0}$ ， $\mathbb{D}(\mathbf{f}) = \mathbf{I}_m$ （即  $\mathbf{f}$  的各分量方差为 1，且互不相关）。最后，设  $\mathbf{e} = (e_1, e_2, \dots, e_p)^T$  和  $\mathbf{f}$  互不相关，并且

$$\mathbb{E}(\mathbf{e}) = \mathbf{0}, \mathbb{D}(\mathbf{e}) = \text{diag}(\sigma_1^2, \dots, \sigma_p^2) \text{ 为对角矩阵。}$$

假设随机向量  $\mathbf{X}$  满足以下模型：

$$\begin{cases} X_1 - \mu_1 = a_{11}f_1 + a_{12}f_2 + \dots + a_{1m}f_m + e_1, \\ X_2 - \mu_2 = a_{21}f_1 + a_{22}f_2 + \dots + a_{2m}f_m + e_2, \\ \dots \\ X_p - \mu_p = a_{p1}f_1 + a_{p2}f_2 + \dots + a_{pm}f_m + e_p, \end{cases}$$

我们称该模型为正交因子模型，矩阵表示为：

$$\mathbf{X} = \boldsymbol{\mu} + \mathbf{A}\mathbf{f} + \mathbf{e} \quad (2.1)$$

其中  $f_1, f_2, \dots, f_m$  称为  $\mathbf{X}$  的公共因子； $e_1, e_2, \dots, e_p$  称为  $\mathbf{X}$  的特殊因子。公共因子一般对  $\mathbf{X}$  的每一个分量都起作用，而  $e_i$  一般仅仅对  $X_i$  起作用。并且各个特殊因子之间以及特殊因子和所有公共因子之间都是不相关的。其中  $\mathbf{A} = (a_{ij})_{p \times m}$  是待估计的系数矩阵，称为因子载荷矩阵。 $a_{ij}$  称为第  $i$  个变量在第  $j$  个因子上的载荷，它反映了第  $i$  个变量在第  $j$  个因子上的相对重要性。

对于正交因子模型，在因子载荷矩阵  $\mathbf{A}$  中，我们计算各列的平方和，记为  $q_j^2$ ，即

$$q_j^2 = \sum_{i=1}^p a_{ij}^2 (j = 1, 2, \dots, m) \quad (2.2)$$

$q_j^2$  表示第  $j$  个公共因子  $f_j$  对  $\mathbf{X}$  的所有分量的总影响，称为公共因子  $f_j$  对  $\mathbf{X}$  的方差贡献。因此如果我们列出  $\mathbf{A}$  的所有列平方和，按照方差贡献大小就可以选出最有影响的公共因子。因此因子分析的关键步骤就是估计出因子载荷矩阵。

对于正交因子模型可以通过主成分法、主因子法和极大似然法来估计因子载荷矩阵。不可观测的公共因子有时候也需要进行估计，例如用来诊断因子模型或者作为进一步分析的原始数据，这就需要我们给出因子得分，一般在得出了因子载荷矩阵之后，对于因子得分，可以通过加权最小二乘法（巴特莱因子得分）或者通过回归法（汤普森因子得分）进行估计。

但是对于动态因子模型和近似动态因子模型，因为不满足正交因子模型的一些关键模型假设，因此需要不同的估计手段。

### 2.1.2 动态因子模型

令  $\mathbf{X}_t = (\mathbf{X}_{1t}, \mathbf{X}_{2t}, \dots, \mathbf{X}_{pt})^T$  为一组宏观经济变量在  $t$  时刻的水平，并且  $\mathbf{X}_t$  可以表达为如下形式：

$$\mathbf{X}_t = \lambda(L)\mathbf{f}_t + \mathbf{e}_t \quad (2.3)$$

其中， $\mathbf{f}_t$  为  $q$  维的动态因子向量， $\lambda(L)$  为由  $s$  阶滞后多项式算子组成的  $p \times q$  矩阵。动态因子模型通常假设动态因子向量服从某一个向量随机过程。即动态因子模型不仅允许观测变量受因子滞后项的影响，而且也允许因子本身具有独立的动态演化过程。

动态因子模型的提出者 Geweke 和 Sargent 等人都是使用频域方法估计模型，这种方法不能够直接估计出动态因子  $\mathbf{f}_t$ ，因此也就不能将因子作为预测或者扩展模型等用途。使得动态因子模型的应用收到限制。

### 2.1.3 近似因子模型

Chamberlain 和 Rothschild 放弃异质性部分的协方差矩阵为对角阵的假定，允许异质性部分存在一定程度的截面相关，模型假设允许  $\mathbf{f}_t$  的各个分量  $f_1, f_2, \dots, f_m$  具有相依性， $e_1, e_2, \dots, e_p$  可以不独立，并且允许  $\mathbf{f}_t, \mathbf{e}_t$  具有时间序列相依性。从而将模型扩展为了为动态近似因子模型，也简称为近似因子模型。

由于动态近似因子模型估计困难，因此在预测中，往往使用模型的静态形式：

$$\mathbf{X}_t = \mathbf{A}\mathbf{F}_t + \mathbf{e}_t \quad (2.4)$$

式(2.4)中， $\mathbf{F}_t$  是  $m$  维向量，称为静态因子，即  $\mathbf{F}_t$  仅在当期影响  $\mathbf{X}_t$ （因为  $\mathbf{F}_t$  包括了动态因子  $\mathbf{f}_t$  的当期项和滞后项），它本身可以不具有经济学含义。 $\mathbf{A}$  是因子载荷矩阵。本文采用(2.4)中的模型进行宏观经济指标的预测。

对于静态形式的估计，Stock 和 Watson 于 1989 年最早提出了一种方法 [27]：这种方法采用 Kalman 滤波构造似然函数，并采用极大似然方法来估计参数。这种方法的优点是：在误差项服从正态分布的假定下，能够得到因子的有效估计量。然而，由于在估计参数过程中会应用到非线性数值优化算法，而待估参数的个数与变量维数  $p$  成比例，限于当时的计算能力，这种方法只能用于处理低维的精确动态因子模型。

针对维数很高的宏观经济变量，Stock 和 Watson 于 2002 年给出了一种非参数的方法 [30]，并且得到广泛使用。这种方法使用主成分分析法进行估计，对(2.4)，因子载荷矩阵  $\mathbf{A}$  的估计量  $\hat{\mathbf{A}}$  即为  $\mathbf{X}_t$  的样本协方差矩阵

$$\hat{\Sigma}_{\mathbf{X}} = \frac{1}{n} \sum_{t=1}^n \mathbf{X}_t \mathbf{X}_t^T \quad (2.5)$$

的前  $r$  个最大特征值所对应的特征向量组成的  $p \times r$  维矩阵，静态因子的主成分估计量为：

$$\hat{\mathbf{F}} = \hat{\mathbf{A}}\mathbf{X}_t \quad (2.6)$$

主成分估计量具有一系列良好的性质。例如，在  $p \rightarrow \infty, n \rightarrow \infty$  且  $p^2/n \rightarrow \infty$  的条件下，主成分估计量  $\hat{\mathbf{F}}_t$  是因子空间的一致估计量，且在随后的建模过程中可以当作观测到的数据。

## §2.2 $L_2$ 范数主成分分析

在许多领域的数据分析中，常会遇到输入变量维数很高的情况，为了方便对建立模型或处理数据，常常需要通过某些方法将输入变量个数减少。人们希望同时做到减少输入变量的个数、简化问题复杂度的同时尽量不损失原始数据的信息，因此直接舍弃变量的做法一般是不可取的。

比较合适的解决办法是通过某种手段，将高维变量组成的数据集映射到低维，并且尽量保持数据集的信息量。这种方法称为数据降维。降维后的数据既包含原数据绝大部分的信息，同时又具备容易分析和处理的特点。

主成分分析 (principal component analysis, PCA) 就是一种十分流行的数据降维方法，它已经在信号处理、机器学习等众多领域得到了广泛应用。在经济学、心理学等社会科学领域的一个重要应用就是对因子模型进行估计。

通常，一个数据集总是由若干随机变量的若干观测组成。主成分分析的目标就是将原始数据集进行降维，将这些观测投射到一个低维空间中。这样的投射有无数种，主成分分析希望找到这样一种投射，可以使得数据在低维空间的投影拥有最大的方差，因为在统计学上，方差反映了样本数据中包含的信息量。

如果使用规范的数学表述， $L_2$  主成分估计分析问题可以表述为

$$P_1 : \hat{\mathbf{A}}_{p \times m}, \hat{\mathbf{F}}_{m \times n} = \arg \min_{\mathbf{A}, \mathbf{F}} \|\mathbf{X} - \mathbf{A}\mathbf{F}\|_{L_2} = \arg \min_{\mathbf{A}, \mathbf{F}} \sum_{i=1}^p \sum_{j=1}^m (x_{ij} - a_i^T f_j)^2. \quad (2.7)$$

其中  $\mathbf{X}$  为  $p \times n$  的高维数据矩阵， $\mathbf{A}$  的列构成了  $\mathbf{X}$  的  $m$  维线性子空间的基，这个子空间也称为特征空间。 $\mathbf{F}$  为一系数矩阵，给出了  $\mathbf{X}$  各列元素在特征空间中的坐标，根据矩阵投影理论，在给定  $\mathbf{A}$  的条件下， $\mathbf{F} = \mathbf{A}^T \mathbf{X}$ 。问题  $P_1$  可以解释为，需要找到一个合适的投射矩阵，使得数据在低维的投影上升回高维空间后和原矩阵各元素的误差平方和最小。

对于问题  $P_1$  常用奇异值分解法求解。同样地我们也可考虑其对偶问题  $P_2$ ，

$$P_2 : \hat{\mathbf{A}} = \arg \max_{\mathbf{A}} \|\mathbf{A}^T \mathbf{X}\|_{L_2}, \text{ 其中 } \mathbf{A}^T \mathbf{A} = \mathbf{I}_m \quad (2.8)$$

问题  $P_2$  可以理解为，需要找到一个合适的投射矩阵，使得数据在低维空间的投影有最大的方差。在统计学上，数据的方差反映了数据中信息的多少，因此在特征空间中选取方差最大的方向作为主成分是很恰当的。因此常用特征值分解的方法求解  $P_2$ ，原因是原始数据协方差矩阵的某一特征值就是该特征值对应特征向量的方差，因此我们只要对原始数据的协方差矩阵进行特征值分解，找出最大的一些特征值，它们对应的特征向量就可以作为主成分了。通常用主成分方向上的方差的和占据总方差的比例来衡量主成分包含原始数据信息量的多少。

对于(2.5)，下面给出使用特征值分解进行主成分分析的步骤：记矩阵  $\mathbf{X}_{p \times n}$  为  $p$  维宏观经济变量  $\mathbf{X}_t$  在  $t_1, \dots, t_n$  的  $n$  次观测，

1. 将  $\mathbf{X}$  进行中心化；
2. 计算样本协方差矩阵  $\frac{1}{n} \mathbf{X} \mathbf{X}^T_{p \times p}$ ；
3. 对样本协方差矩阵特征值分解，并从小到大排列这些特征值；
4. 因子载荷矩阵估计量  $\hat{\mathbf{A}}_{p \times m} (m < p)$  为前  $m$  个特征值对应的特征向量组成的矩阵。

不难看出，主成分分析法进行估计的关键计算步骤是对样本协方差矩阵的特征值分解。而奇异值分解解决将矩阵  $\mathbf{A}$  分解成正交矩阵  $\mathbf{U}$  和对角矩阵  $\mathbf{\Sigma}$  和另一正交矩阵  $\mathbf{V}^T$  的问题，即

$$\mathbf{A}_{m \times n} = \mathbf{U}_{m \times m} \mathbf{\Sigma}_{m \times n} \mathbf{V}_{n \times n}^T$$

奇异值分解求解的关键就是得到奇异值，而后者是  $\mathbf{A}^T \mathbf{A}$  的特征值的平方根，即奇异值分解的关键计算步骤是对  $\mathbf{A}^T \mathbf{A}$  进行特征值分解。这里假设我们取  $\mathbf{A} = \mathbf{X}^T / \sqrt{m}$ ，不难看出和特征值分解法等价。

为了方便起见，下面我们把求解  $P_1$  和  $P_2$  统称为  $L_2$  范数主成分分析。

§2.3  $L_1$  范数主成分分析

$L_2$  范数主成分分析具有所有的  $L_2$  范数优化方法普遍的缺陷，那就是对离群值十分敏感。并且特征分解和奇异值分解都不能直接处理具有缺失值的数据，因此对于缺失数据必须进行插补。

然而由于各种原因，高维宏观经济数据中往往具有大量的缺失值和离群值，这就给因子模型的估计带来很多的麻烦，从而进一步使用因子的估计量进行预测可能会变得不够准确。

将  $P_1$  中的目标函数更换为使用  $L_1$  范数，可得

$$P_3 : \hat{\mathbf{A}}_{p \times m}, \hat{\mathbf{F}}_{m \times n} = \arg \min_{\mathbf{A}, \mathbf{F}} \|\mathbf{X} - \mathbf{A}\mathbf{F}\|_{L_1} = \arg \min_{\mathbf{A}, \mathbf{F}} \sum_{i=1}^p \sum_{j=1}^m |x_{ij} - \mathbf{a}_i^T \mathbf{f}_j| \quad (2.9)$$

若同样假设  $\mathbf{A}_{p \times m}$  为  $\mathbf{X}$  子空间的基组成的矩阵，即问题  $P_1$  就转化为问题  $P_3$ 。同样地，我们改变问题  $P_2$ ，若我们想要寻找一个投射使得数据在低维的投影有最大的  $L_1$  范数，这就得到问题  $P_4$ 。我们把  $P_3$  和  $P_4$  称为  $L_1$  范数主成分分析，注意到这里  $P_3$  和  $P_4$  不再是对偶问题。所以求解  $L_1$  范数主成分分析，常常是解决这两个问题其中之一。

2.3.1 求解  $L_1$  范数主成分分析的交替凸优化算法

一种经典的方法是使用交替凸优化算法 [17] 求解  $P_3$ ，即每一步固定  $\mathbf{A}$  或者  $\mathbf{F}$  的值，来求解另一个参数。(2.9)中定义的问题不是一个凸优化问题。但是一旦矩阵  $\mathbf{A}$  或者  $\mathbf{F}$  固定为常数，那么该问题就成为了一个凸优化问题，可以找到全局最优解。

$$\mathbf{F}^t = \arg \min_{\mathbf{F}} \|\mathbf{X} - \mathbf{A}^{t-1} \mathbf{F}\|_{L_1} \quad (2.10)$$

$$\mathbf{A}^t = \arg \min_{\mathbf{A}} \|\mathbf{X} - \mathbf{A} \mathbf{F}^t\|_{L_1} \quad (2.11)$$

我们改写式(2.10)中的目标函数，

$$E(\mathbf{F}) = \|\mathbf{X} - \mathbf{A}^{t-1} \mathbf{F}\|_{L_1} = \sum_{j=1}^n |\mathbf{x}_j - \mathbf{A}^{t-1} \mathbf{f}_j| \quad (2.12)$$

其中  $\mathbf{x}_j$  是矩阵  $\mathbf{X}$  的第  $j$  列， $\mathbf{f}_j$  是  $\mathbf{F}$  的第  $j$  列。于是式(2.10)问题可以分解为  $n$  个独立的子优化问题，求解相应的  $\mathbf{f}_j$ ：

$$\mathbf{f}_j = \arg \min_{\boldsymbol{\theta}} |\mathbf{A}^{t-1} \boldsymbol{\theta} - \mathbf{x}_j| \quad (2.13)$$

同样地，(2.11)可以转化为下面的  $p$  个独立的子优化问题，

$$\mathbf{a}_i^T = \arg \min_{\boldsymbol{\theta}} |\mathbf{x}_i^T - \mathbf{F}^T \boldsymbol{\theta}| \quad (2.14)$$

其中  $\mathbf{a}_i$  为  $\mathbf{A}$  的第  $i$  行，而  $\mathbf{x}_i$  为  $\mathbf{X}$  的第  $i$  行。

使用特征值分解法或奇异值分解法时我们需要对矩阵  $\mathbf{X}$  进行缺失值插补，然后才能进行计算。在  $L_1$  主成分分析中我们不需要进行缺失值插补，在式中，遇到  $x_j$  具有缺失值的场合，我们直接舍弃相应的约束条件即可。我们改写(2.12)，

$$E(\mathbf{F}) = \sum_{i=1}^d \sum_{j=1}^n |x_{ij} - \mathbf{a}_i^T \mathbf{f}_j|$$

如果某个项  $x_{ij}$  缺失，我们直接舍弃目标函数中的对应累加项，在上述算法中对应的做法就是直接删除(2.13)的一个约束条件。

### 2.3.2 算法步骤

我们已经发现可以通过交替优化算法求解  $L_1$  范数主成分分析，下面我们更加详细地讨论该算法一些细节和具体实现步骤。

在算法开始时，首先我们需要给  $\mathbf{A}$  一个初始值  $\mathbf{A}^{(0)}$ 。对于  $\mathbf{A}$  可以采用简单随机数进行初始化，这里我们为了加快收敛速度，可以使用经过缺失值插补（这里我们使用均值插补）后通过 PCA 算法进行得到的因子载荷矩阵作为  $\mathbf{A}^{(0)}$ 。在本章后续小结的实验中我们可以发现，在含有大量缺失值和离群值的条件下，两种不同的初始化方法最终结果差异并不大。

因为目标函数  $E(\mathbf{A}, \mathbf{F}) = \|\mathbf{X} - \mathbf{A}\mathbf{F}\|_{L_1}$  在每一个交替的优化步骤中都递减，并且  $E(\mathbf{A}, \mathbf{F})$  具有下界 ( $\geq 0$ )。因此交替优化算法一定收敛。因此我们可以设定一个收敛域值来停止迭代，这里我们设置终止条件：

$$\theta(a_i^{(t)}, a_i^{(t-1)}) < \alpha$$

这里  $\theta(a, b)$  表示向量  $a$  和  $b$  的夹角；其中  $a_i$  是  $\mathbf{A}$  或者  $\mathbf{F}$  的第  $i$  列； $\alpha$  是一个很小的正数。

算法 3.1 给出了算法具体步骤：

---

算法 3.1：交替凸优化求解  $L_1$  主成分分析 (ACP, Alternate Convex Programming)

---

1. 初始化：给出  $\mathbf{A}$ ,  $\Sigma$  的初始值  $\mathbf{A}^{(0)}$ ,  $\Sigma^{(0)} = \mathbf{I}$ , (其中  $\Sigma$  为一对角矩阵,  $\mathbf{I}$  为单位矩阵)；
2. 交替凸优化：对于迭代次数  $t = 1, \dots$ , 收敛：

$$\mathbf{F}^{(t)} = \arg \min_{\mathbf{F}} \|\mathbf{X} - \mathbf{A}^{(t-1)} \Sigma^{(t-1)} \mathbf{F}^T\|_{L_1}$$

$$\mathbf{A}^{(t)} = \arg \min_{\mathbf{A}} \|\mathbf{X} - \mathbf{A} \Sigma^{(t-1)} \mathbf{F}^{T(t)}\|_{L_1}$$

$$\text{归一化: } \begin{cases} N_a = \text{diag}(\mathbf{A}^{(t)T} \mathbf{A}^{(t)}) \\ N_f = \text{diag}(\mathbf{F}^{(t)T} \mathbf{F}^{(t)}) \\ \mathbf{F}^{(t)} \leftarrow \mathbf{F}^{(t)} N_f^{-1} \\ \mathbf{A}^{(t)} \leftarrow \mathbf{A}^{(t)} N_a^{-1} \\ \Sigma^{(t)} \leftarrow N_a \Sigma^{(t-1)} N_f \end{cases}$$

3. 输出结果： $\mathbf{F} \leftarrow \mathbf{F} \Sigma^{1/2}$ ,  $\mathbf{A} \leftarrow \mathbf{A} \Sigma^{1/2}$
- 

## §2.4 稳健性实验

为了检验  $L_1$  范数主成分分析在处理含有大量离群值和缺失值的数据时的稳健性，我们进行模拟实验，来对比  $L_2$  范数主成分分析。

## 2.4.1 数据准备

为了进行模拟实验，我们首先需要随机产生一个高维低秩的矩阵来模拟高维宏观经济数据集。我们产生一个  $n$  维方阵  $M$ ，其中每一个随机元素均服从  $[-100, 100]$  的均分分布。然后我们对方阵  $M$  进行奇异值分解， $M = U\Sigma V^T$ 。假设我们需要产生的低秩矩阵的秩为  $r$ ，则

$$X = U_{(:,1:r)} \Sigma_{(1:r,1:r)} V_{(:,1:r)}^T$$

即为我们得到的模拟高维低秩矩阵。

之后我们可以设置一定比例的缺失值和离群值，首先我们在矩阵的左下角剔除部分元素。在剩下的元素中，我们随机选取一部分然后重新产生随机元素，每个元素服从  $[-2000, 2000]$  上的均匀分布。图 2.1 展示了一个  $30 \times 30$  秩为 3 的矩阵在模拟了缺失值和离群值后的情况。

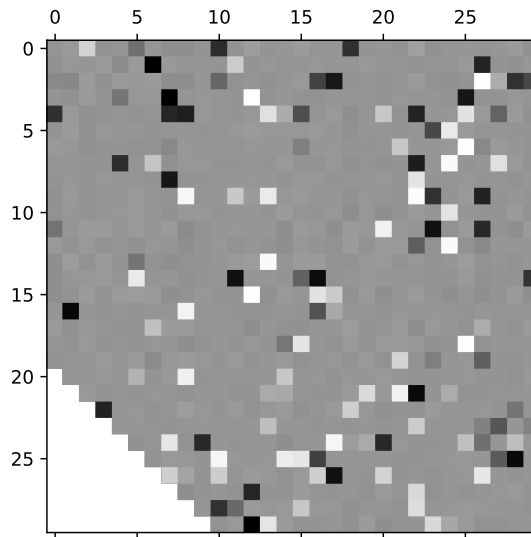


图 2.1:  $30 \times 30$  的模拟矩阵，左下角白色代表缺失值，图中灰度越深代表元素的绝对值越大，因此深黑色的点代表了离群值

基于获得的高维低秩矩阵，我们主要比较以下算法：对于  $L_1$  范数主成分分析，我们使用 IRP 算法。对于  $L_2$  范数主成分分析，采用了下面三种算法：1) 奇异值分解，该算法需要进行缺失值插补，这里我们使用均值插补法。记作 SVD；2) 基于 IRP 算法，但是更换损失函数为  $L_2$  范数，可以直接处理原始数据。记作 IRP- $L_2$ ；3) 一种基于迭代加权  $L_2$  范数的估计方法 (Black 和 Rangarajan, 1996)，通过将问题  $P_1$  损失函数加上随迭代变化的权重，可给  $L_2$  范数主成分分析带来一定的稳健性，可以直接处理原始数据。记作 IRLS(iteratively reweighted least squares)。

## 2.4.2 实验结果

令  $\hat{X} = \hat{A}_{N \times M} \hat{F}_{M \times N}$ ，我们比较因子残差  $E = \hat{X} - X$ 。观察重构残差的平方的分布情况，试验每一种算法作用在无干扰数据上和干扰后数据上的情况。

对于无干扰数据，每一种算法都具有很好的表现；对于进行了缺失值和离群值模拟的数据，我们每组试验（给定  $M$ ， $N$ ）下均重复多次取平均值，图 2.3 给出了一个典型结果。

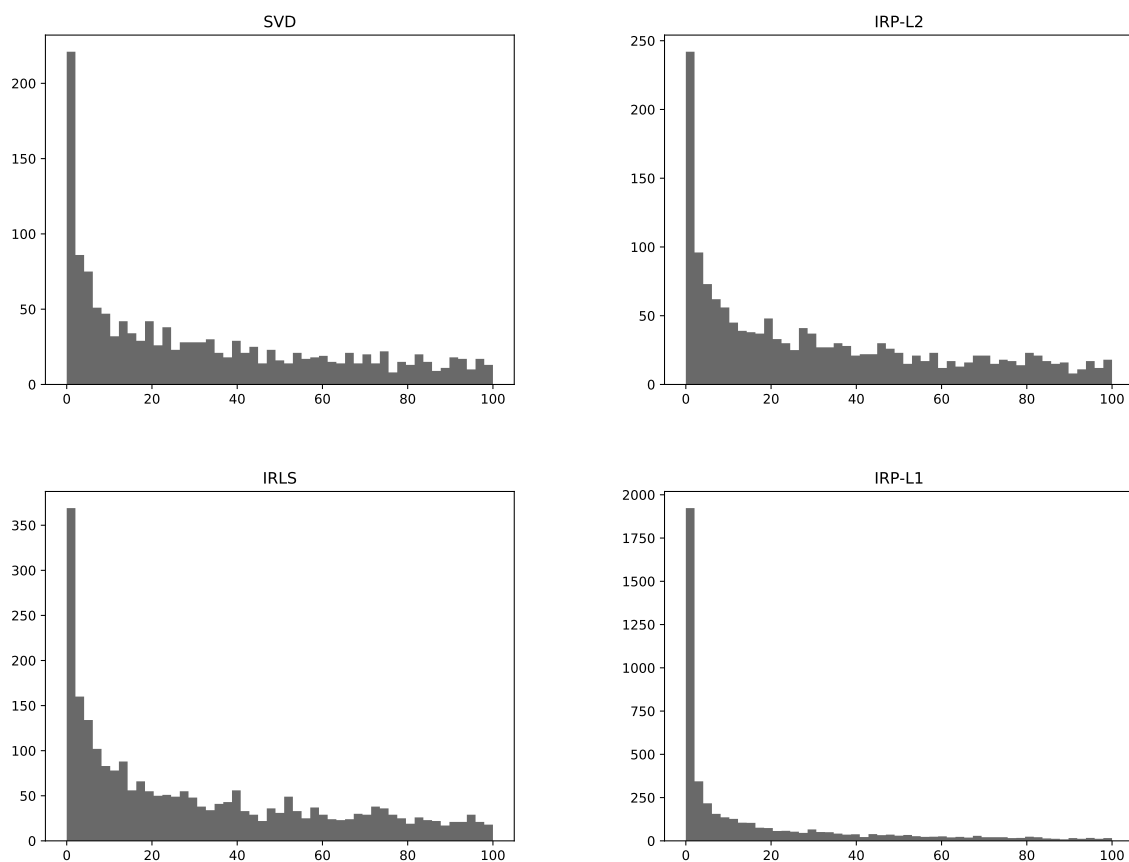


图 2.2: 图示为取  $N = 80, M = 3$ ，缺失值和离群值比例均为 10% 的情况下，多次实验中平均重构残差的平方的分布情况，横轴表示残差平方，纵轴表示频数。可以看出 IRP-L1 的重构残差表现最好。

实验表明， $L_1$  范数主成分分析相比  $L_2$  主成分分析法和其他基于  $L_2$  范数优化的方法相比具有更好的稳健性，特别适合于处理具有大量离群值的数据。

## §2.5 基于国内主要月度宏观经济数据的实证研究

上一节我们阐述了  $L_1$  范数主成分分析方法，本节我们进行基于国内主要月度宏观数据的实证研究，来探究使用  $L_1$  范数主成分分析用来代替  $L_2$  范数主成分分析进行近似因子模型估计的可行性。

数据方面将选择高维的国内主要月度宏观经济数据，该数据集包含众多的经济指标，其中许多经济指标具有重尾分布，即意味着包含较多的离群值。并且该数据集在时间序列上不完整，一些经济指标的观测值缺失。

本实证研究采用扩散指数模型进行经济预测 [31]，近年来该模型已经得到了广泛的应用。该预测模型依赖近似因子模型的因子得分估计作为关键自变量，本次研究将对  $L_2$  范数和  $L_1$  范数主成分分析下不同的因子得分估计在宏观经济指标预测中的表现。最后，将结合实证结果对  $L_1$  范数主成分分析在本场合的应用提出一些建议。

### 2.5.1 扩散指数模型

令  $y_t$  为待预测经济变量  $y$  在时间  $t$  的水平,  $\mathbf{X}_t$  为  $p$  维随机向量, 假设  $(X_t, y_t)$  服从近似因子模型并且  $\mathbf{X}_t$  和  $y_t$  具有相依性, 若  $\mathbf{X}_t, y_t$  有  $m$  维共同因子  $\mathbf{F}_t$ , 即

$$\mathbf{X}_t = \mathbf{A}\mathbf{F}_t + \mathbf{e}_t \quad (2.15)$$

则可以通过式(2.16)对  $y_{t+h}$  进行预测,

$$y_t = \beta(L)\mathbf{F}_t + \alpha(L)y_t + c + e_t \quad (2.16)$$

其中滞后算子  $\beta(L)$  反映了共同因子滞后项的影响, 而  $\alpha(L)$  表示  $y_t$  自身的滞后项的影响。

### 2.5.2 因子个数判定

本节采用 Bai 和 Ng 于 2002 年提出的确定静态因子个数的信息准则 [2], 该准则平衡了模型的拟合优度和模型简约性。在该信息准则下, 我们的因子个数需要最小化

$$IC(m) = \ln(V(\hat{\mathbf{A}}, \hat{\mathbf{F}})) + mG(p, T) \quad (2.17)$$

其中  $V(\hat{\mathbf{A}}, \hat{\mathbf{F}})$  为因子残差平方和除以  $pT$ 。而  $G(p, T)$  为一惩罚函数, 该函数使得在  $p, T \rightarrow \infty$  时  $G(p, T) \rightarrow 0$  且  $\min(p, T)G(p, T) \rightarrow \infty$ 。参考 Bai 和 Ng 文中建议, 本节实证研究中选择

$$G(p, T) = \left(\frac{p+T}{pT}\right) \ln\left(\frac{pT}{p+T}\right)$$

构造信息准则。

需要说明的是由于使用交替凸优化算法求解  $L_1$  主成分分析需要  $m$  已知, 因此我们利用  $L_2$  主成分分析确定因子个数。

### 2.5.3 数据说明

本次实证研究数据来自 EPS 数据平台——中国宏观经济数据库。

我们选择了从 1999 年 9 月至 2019 年 6 月主要公开月度宏观经济指标, 主要包括了工业 (重工业增加值、轻工业增加值、铁矿石原产量、工业企业利润总额等)、运输和邮电业 (客运总量、货运总量、邮电业务总量等)、能源 (原油生产总量、发电量、原煤产量等)、国内贸易 (烟酒消费品零售总额、日用品类零售总额、粮油食品类零售值、服装纺织品类零售值等)、固定资产投资 (第一产业固定资产投资、建筑安装工程投资、设备工具购置投资等)、房地产开发业 (住宅投资完成额、办公楼投资完成额、建筑工程投资额、安装工程投资额等)、对外经济贸易 (对外承包工程营业额、纺织品出口金额、大豆进口量、原油进口量等)、财政 (国家债务收入、企业亏损补贴、一般公共服务支出、社会保障和就业支出等)、金融 (货币供应量、股票成交额、基金份额、人民币汇率等)、保险 (财产险保费收入、人身险保费收入、财产险保费支出、保险业投资等)、就业与社会保障 (城镇新增就业人数、城镇职工基本养老保险基金收入、医疗保险基金支出等) 等共 11 个类别, 总计 120 个指标的月度时间序列。

另外, 我们还选择了一些景气指数, 它们可用来作为待预测的变量, 也可用来做预测变量形成对照组。由于 2015 年后我国不再公布先行、一致、滞后和预警指数, 并且在 2011 年后弃用国房景气指数, 我们这里选择了制造业采购经理指数、非制造业采购经理指数、消费者预期指数、和消费者景气指数四项。



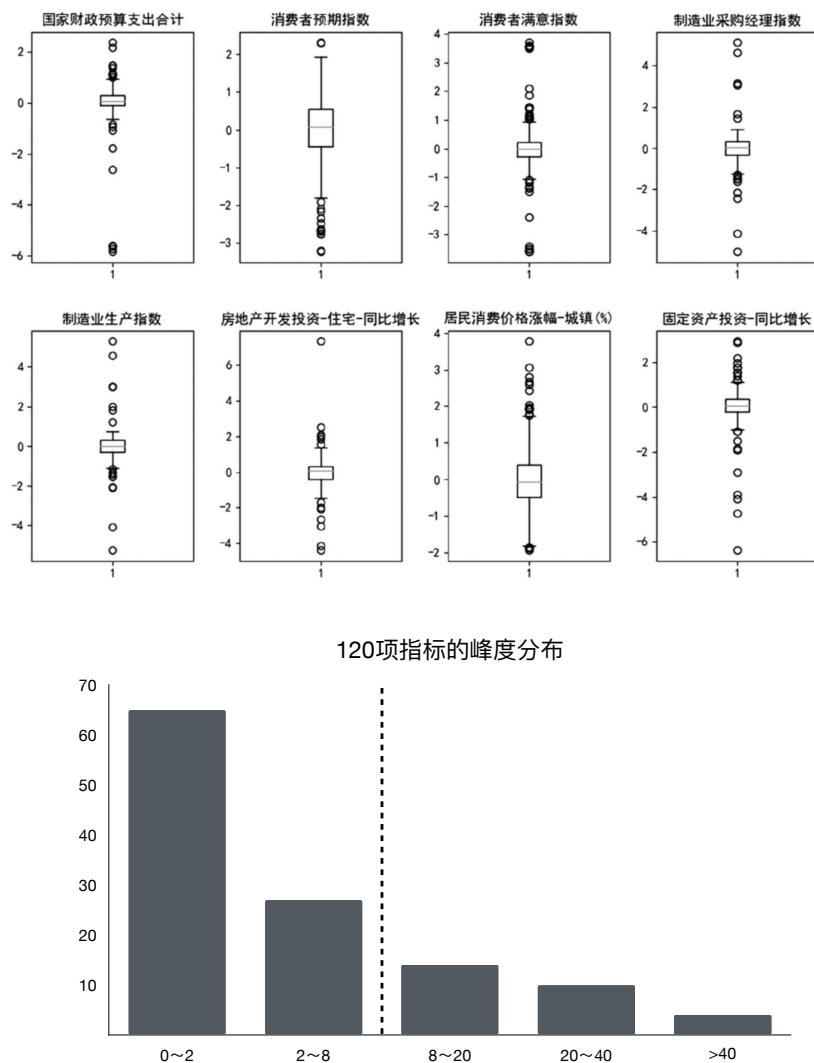


图 2.3: 部分指标的箱形图和各经济指标的峰度分布情况

由于某些指标曾多次改变统计口径、统计频率等原因，这些数据中包含许多的缺失数据。另外，在 1999~2019 年这 20 年间，中国经济经历了许数次重大外部冲击，许多经济指标中含有大量离群值（见图 3），从各指标的经验分布的峰度值来看，经济指标的分布存在分布重尾的现象。

在预处理阶段，我们对不平稳数据进行一阶差分；之后所有数据均进行标准化处理，我们不剔除任何离群值；另外，对于缺失数据的情况，在进行  $L_2$  主成分分析时对原始数据进行插补，插补方法是对已有数据拟合 ARMA 模型根据估计得到的模型填补对应缺失数据。

下面介绍两组因子的主成分估计量的获得：首先进行  $L_2$  主成分分析并根据(2.17)选择最合适的因子个数（本组数据得出最佳因子个数为 8）；接下来将  $L_2$  主成分估计的因子载荷矩阵作为交替凸优化算法  $\mathbf{A}^0$  的初始值，然后求解得到  $L_1$  因子载荷。根据(2.6)得出两组因子得分的估计：我们称为  $L_1$  因子和  $L_2$  因子。

### 2.5.4 实证结果

将得到的  $L_1$  因子和  $L_2$  因子看作  $m$  维随机向量的一组时间序列, 根据(2.16)建立预测模型。对预测结果的评价, 下面选取了三种指标: MSE、MAE 和 MPAE。

由于景气指数反映了经济中多方面特征, 可以将几种景气指数的时间序列作为预测变量, 也根据(2.16)建立预测模型。

我们把  $L_2$  因子的三种平均误差指标作为基准, 记录  $L_1$  的对应误差和它的比, 作为对照我们这里将景气指数模型的预测情况也进行展示。

表 1、表 2 和表 3 分别对应了  $h = 1, 3, 6$  情况下部分变量预测的平均误差。

	Rel MSE (L1 PCA)	Rel MSE (IM)	Rel MAE(L1 PCA)	Rel MAE (IM)	Rel MPAE(L1 PCA)	Rel MPAE (IM)
工业生产总值 *	0.81	1.49	0.87	1.67	0.43	1.55
粮食总产量 *	0.70	1.98	0.75	1.54	0.96	1.60
货币供应量 M2*	0.76	1.25	0.90	1.19	0.45	0.97
固定资产投资总额 *	0.89	1.03	0.97	0.99	0.81	1.32
房地产开发投资总额 *	0.79	0.98	0.95	1.00	0.91	1.20
社会消费品零售总额 *	0.84	1.25	0.87	1.12	0.45	1.17
出口总额	1.24	1.69	1.06	1.43	0.90	1.50
住宅新开工面积总数 *	0.89	2.40	0.85	1.98	0.45	2.22
股票流通市值 *	0.99	3.99	0.98	2.84	0.81	4.51
城乡储蓄存款余额 *	0.93	1.01	0.98	0.99	0.98	1.56

表 1: 向前一个月预测结果

	Rel MSE (L1 PCA)	Rel MSE (IM)	Rel MAE(L1 PCA)	Rel MAE (IM)	Rel MPAE(L1 PCA)	Rel MPAE (IM)
工业生产总值 *	0.85	1.16	0.95	1.22	1.32	1.93
粮食总产量 *	0.98	1.39	0.97	1.20	0.93	1.12
货币供应量 M2	1.08	1.20	1.05	1.44	0.66	1.70
固定资产投资总额 *	0.99	1.11	0.98	1.28	0.82	0.89
房地产开发投资总额 *	0.91	0.94	0.92	1.12	0.66	0.99
社会消费品零售总额 *	0.62	1.12	0.85	0.99	1.26	1.05
出口总额	1.26	1.10	1.10	1.30	1.04	1.72
住宅新开工面积总数	0.81	1.62	1.12	1.78	1.04	1.23
股票流通市值	0.99	1.98	1.00	1.50	1.04	1.91
城乡储蓄存款余额 *	0.99	1.35	0.96	1.02	0.94	1.17

表 2: 向前三个月预测结果

## §2.6 本章小结

	Rel MSE (L1 PCA)	Rel MSE (IM)	Rel MAE(L1 PCA)	Rel MAE (IM)	Rel MPAE(L1 PCA)	Rel MPAE (IM)
工业生产总产值 *	0.91	1.20	0.93	1.12	0.68	1.81
粮食总产量 *	1.00	1.30	0.94	1.18	0.79	1.77
货币供应量 M2	0.84	1.12	1.06	1.05	1.46	1.08
固定资产投资总额 *	0.88	1.43	0.94	1.29	0.93	1.85
房地产开发投资总额 *	0.90	1.01	0.91	1.12	0.73	1.07
社会消费品零售总额 *	0.64	1.39	0.79	1.20	1.30	1.34
出口总额	1.46	1.40	1.20	1.32	1.45	1.82
住宅新开工面积总数 *	0.85	1.50	0.96	1.36	0.70	1.02
股票流通市值	0.98	2.24	1.01	1.96	1.28	2.98
城乡储蓄存款余额 *	0.99	1.48	0.98	1.19	0.97	1.70

表 3: 向前六个月预测结果

从结果中可以看出,

## §2.6 本章小结

本章首先简要介绍了因子模型的基础理论, 包括了正交因子模型、动态因子模型和近似因子模型的模型假设和适用场景。近似因子模型的因子估计量可以用于宏观经济预测, 我们着重讨论了它的一种非参数估计——主成分估计量。主成分分析、奇异值分解等方法都等价于基于  $L_2$  范数优化问题, 都不具有对离群值的稳健性。针对宏观经济数据往往分布重尾、具有大量缺失和离群值的情况, 我们提出了引入  $L_1$  范数来改善估计的稳健性, 并且提出了基于  $L_1$  范数的因子模型估计方法。最后, 通过模拟实验, 验证了基于  $L_1$  范数的因子模型估计方法的稳健性。

## 第三章 最小绝对值回归的性能研究

本文的主题是稳健  $L_1$  范数主成分分析方法在宏观经济中的应用问题研究。 $L_1$  范数是最常用的几种重要范数之一，它在数值计算、统计学、运筹学、机器学习等领域有着极为重要的应用。在统计学研究中，已经基于  $L_1$  范数发展出了许多稳健统计方法，主要是利用  $L_1$  范数的良好统计性质。

$L_1$  范数稳健方法在计量经济学中一个重要应用是最小绝对值回归。最小绝对值回归最早提出是为了弥补最小二乘法的不足。使用最小二乘法建立的线性回归模型虽然具有较好的解释性，但是高斯——马尔可夫条件对模型变量的分布有正态性的要求。因此直接使用最小二乘法来建立模型，往往对原始数据的分布情况要求高。即使变量满足了正态分布，但在模型系数估计时，样本中离群值的存在对模型的系数估计效果有很大影响。

在宏观经济实证研究中，经济变量一般不能够服从正态分布，而因受到经济冲击呈现出重尾的特征。对这样的原始数据不能够通过最小二乘法直接建模，因此常常在回归中剔除异常点，使得经济变量接近正态分布，然而代价是有可能损失较多有价值的信息。

采用最小绝对值回归，是更加稳健的做法。实质上它就是中位数回归，模型的系数也具有很好的解释意义，已经广泛应用在对居民收入的研究中，并且在金融数据分析领域有着广泛的应用。在模型的拟合上，最小绝对值回归对离群值不敏感，具有相当的稳健性，因此可以很好处理重尾的宏观经济变量。

高维宏观经济数据，往往包含了维数众多的经济变量，并且样本数量往往非常大，此时如果想要直接使用最小绝对值回归，在求解时就需要解决变量维数和约束数量很高的线性规划问题，因此计算的开销很高。

最小绝对值回归在计算上的复杂性在一定程度上限制了它的应用。一直以来，普遍的加速最小绝对值回归的方法都尝试使用平滑化方法来近似最小绝对值回归的目标函数，避免直接使用原目标函数。然而，替换目标函数，可能会影响估计量的一致性从而影响最小绝对值回归模型的解释性。

但是近年来随着统计学和机器学习领域对  $L_1$  范数的研究逐渐深入，不断有新的算法尝试解决这个问题。本章将介绍最小绝对值回归的两种较新的估计算法，一种是基于聚类——迭代拆解的算法；另一种是基于替代变量的牛顿迭代方法，两种算法基于不同的出发点，都在一定程度上对解决最小绝对值回归问题做了优化。

我们在第二章中介绍了求解  $L_1$  主成分分析的交替凸优化算法，该算法简洁直观，并交替解决的子凸优化问题可以有最优解。其中(2.13)和(2.14)在交替凸优化算法中我们采用线性规划来求解，但是对于高维宏观经济数据而言，维数  $p$  和样本量  $n$  都非常大，由于交替凸优化算法中大量采用线性规划法求解，使得计算开销非常大。而在许多场合，我们常常希望快速得到近似因子用于进一步分析，这就对  $L_1$  主成分分析提出了性能上的要求。本章研究最小绝对值回归的性能优化算法，有助于我们改进交替凸优化算法的尝试，对本课题的研究有着重要意义。

### §3.1 简介

#### 3.1.1 最小绝对值回归的稳健性

$L_2$  范数最优化问题最常见的是最小二乘法。最小二乘法的优点很多，这里不赘述。尽管在求解大部分问题时用最小二乘估计求解可以得到比较令人满意的效果，但最小二乘法也存在一些局限性，比如，当收集的数据较少或者具有较多的缺失数据并且数据中夹杂有异常点时，用最小二乘法所得的结果就令人难以接受，在此情况下应用所得到到的回归方程或模型进行预测或者拟合时，则预测或拟合的精度是相当

低的, 甚至根本不能使用。正因为最小二乘法对数据中的异常值十分敏感, 当数据中具有较多离群值时, 通过最小二乘、PCA 和 SVD 方法, 得到的估计结果也会受到较大影响。

这里可以不使用最小二乘法而选择  $L_1$  范数的方法即最小一乘法, 它可以增加估计的健壮性。设  $\mathbf{X} = (X_1, X_2, \dots, X_p)^T$  为一  $p$  维随机变量,  $Y$  为响应变量,  $\beta$  为回归系数。假设我们观察到 *i.i.d.* 样本  $\mathbf{X}_{n \times p} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)^T$  和  $\mathbf{Y}_{n \times 1} = (y_1, \dots, y_n)^T$ , 我们一般使用  $\beta$  的最小二乘估计量

$$\hat{\beta} = \arg \min_{\beta} \sum_{i=1}^n \|y_i - \mathbf{x}_i^T \beta\|_{L_2} = \arg \min_{\beta} \sum_{i=1}^n (y_i - \mathbf{x}_i^T \beta)^2 \quad (3.1)$$

在稳健统计中, 我们经常使用其他的目标函数, 例如使用  $L_1$  范数来代替  $L_2$ ,

$$\hat{\beta} = \arg \min_{\beta} \sum_{i=1}^n \|y_i - \mathbf{x}_i^T \beta\|_{L_1} = \arg \min_{\beta} \sum_{i=1}^n |y_i - \mathbf{x}_i^T \beta| \quad (3.2)$$

如图所示, 通常使用  $L_1$  范数在线性回归中可以有效避免离群值造成的干扰。

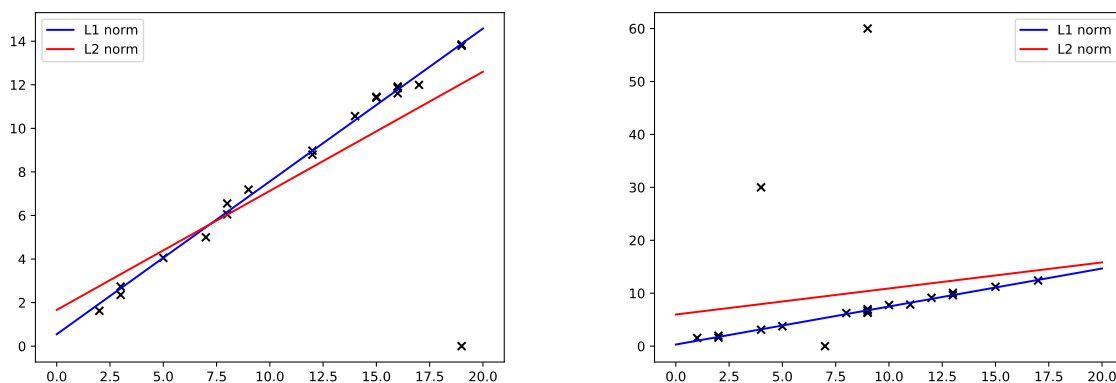


图 3.1: 如图所示, 在简单的线性模型的拟合中, 出现一个离群值就可以导致最小二乘法拟合出现明显的偏差; 而含有较多离群值时最小二乘法拟合变得很不可靠; 而采用  $L_1$  范数则具有相当稳健性。

我们使用  $L_1$  目标函数来估计  $\beta$ , 则其中  $\beta^*$  为最小绝对值回归系数,  $e$  为一随机噪声, 可以写出最小绝对值回归的一般形式

$$\begin{aligned} Y &= \mathbf{X}^T \beta^* + e \\ \beta^* &= \arg \min_{\beta \in \mathbb{R}^p} \mathbb{E}|Y - \mathbf{X}\beta|. \end{aligned} \quad (3.3)$$

### 3.1.2 最小绝对值回归的估计方法

对于(3.2), 它是一个凸优化问题, 但是其不具备显式解, 一般求它的数值解。但是其目标函数在  $\mathbf{0}$  点不可导, 因此不能直接使用使用梯度下降法, 一般来说, 该问题的全局最优解可以通过求解下面的线性规划问题得到:

$$\begin{aligned} \min_{\beta, t} \quad & 1^T t \\ \text{s.t.} \quad & -t \leq \mathbf{X}_{n \times p} \beta_{p \times 1} - \mathbf{Y}_{n \times 1} \leq t \end{aligned}$$

目前对于线性规划问题已经有了比较成熟的解决方法，主要通过单纯形法或者内点法求解，后者的时间复杂度可以控制在多项式时间，然而，一般而言，当  $n$  和  $p$  均很大时，上述线性规划问题面临很高的变量和约束维数，计算速度仍较慢。

由于  $L_1$  范数的目标函数在机器学习领域的大量使用，已经产生了一些光滑化方法，做法是用一个接近  $L_1$  的目标函数来替代它，用来替代的函数往往处处可导，因而可以使用梯度下降法求解。典型的代表就是使用 Huber's M 统计量近似  $L_1$  范数目标函数，

$$\rho(e) = \begin{cases} \frac{1}{2}e^2, & |e| \leq \gamma \\ \gamma|e| - \frac{1}{2}\gamma^2, & |e| > \gamma \end{cases}$$

其中  $\gamma$  为某一正数，该问题可以转化为一个二次规划问题求解。

## §3.2 聚类——迭代拆解算法

近年来针对最小绝对值回归的性能研究发展出了除了光滑化目标函数之外的方法，可以在不改变目标函数的情况下，通过寻求新的优化方法进行求解。这样一来，可不改变最小绝对值回归估计量的统计性质。这里介绍 Park 等人于 2016 年提出的一种基于聚类——迭代拆解算法的最小绝对值回归求解方法。

### 3.2.1 聚类——迭代拆解算法说明

聚类是一种在机器学习中常见的做法，就是按照某种给定的规则，将特征接近的样本点归类到一起。聚类——迭代拆解算法的提出受到以下事实的启发：1) 优化问题面临的数据集模庞大，其中许多的样本点在进行参数估计时的贡献是很接近的；2) 如果对相似的样本点进行聚类，提炼该聚类中的信息，避免每个样本点都进入计算，那么就会大大减小问题的规模；3) 假设在聚类后构造的新数据集上不能接近问题的最优解，那么就拆解当前的聚类，在新聚类上进行计算，聚类个数有限，因此最坏情况下相当于直接求解原问题；

采用聚类——迭代拆解算法求解某个优化问题的前提如下：1) 必须能够提出一种规则来对样本点聚类；2) 必须找到合适的聚类和拆解聚类的标准；3) 需要在聚类后构造的新数据集上明确定义新的优化问题；4) 能够判定当前解是否接近最优解。

算法 2.1 给出了任何一个聚类——迭代拆解算法的主要步骤，注意算法 2.1 必然在某处停止，因为每次不断拆解聚类，当聚类个数  $|K^t| = n$  时，相当于计算原问题，此时算法终止。

---

#### 算法 2.1：聚类——迭代拆解算法 (Aggregate and Iterative Disaggregate, AID)

---

输入：原始数据集  $\mathbf{X}_{n \times p}$ ，样本点的下标集合  $I = \{1, 2, \dots, n\}$ ，数据的特征下标集合  $J = 1, 2, \dots, p$ ，原优化问题  $P$ 。

初始化：对原始数据集  $\mathbf{X}$  聚类，然后按某种规则产生新的优化数据集  $\mathbf{X}^1$ 。

对于  $t = 1, \dots, T$ :

记  $C^t = \{C_1^t, \dots, C_K^t\}$  为聚类的集合， $K^t = 1, \dots, |K^t|$  为当前聚类的下标，

1. 根据当前聚类情况  $C^t$ ，构造新的数据集  $\mathbf{X}^t$ ，求解相应的优化问题  $P^t$ ；

2. 检查解  $s^t$  是否达到最优条件；

3. 如果不满足条件，拆解当前聚类。

---

## 3.2.2 优化最小绝对值回归

改写(3.2)的目标函数,

$$E^* = \min_{\beta \in \mathbb{R}^p} \sum_{i \in I} |y_i - \sum_{j \in J} x_{ij} \beta_j| \quad (3.4)$$

首先给出聚类方法。给定  $|K_0|$  为目标聚类个数, 初始化  $C_0 = \{C_1^0, C_2^0, \dots, C_K^0\}$ , 我们可以使用任意的聚类方法进行初始化。接下来给出如何根据聚类来产生新的数据集, 对于在任一迭代周期内产生的聚类  $C_k^t, k = 1, \dots, K^t$ , 取

$$x_{kj}^t = \frac{\sum_{i \in C_k^t} x_{ij}}{|C_k^t|}, \quad j \in J \text{ 并且 } y_k^t = \frac{\sum_{i \in C_k^t} y_i}{|C_k^t|}$$

对于每一个不同的聚类需要给出一个权重来区分信息量不同的聚类, 因此新的数据集上, 我们求解下面的问题

$$F^t = \min_{\beta^t \in \mathbb{R}^p} \sum_{k=1}^{K^t} |C_k^t| |y_k^t - \sum_{j \in J} x_{kj}^t \beta_j^t| \quad (3.5)$$

容易发现, 任何(3.5)的可行解都是(3.4)的可行解。记  $\hat{\beta}^t$  为(3.5)的解, 在每次迭代, 我们计算此时的原目标函数的取值

$$E^t = \sum_{i \in I} |y_i - \sum_{j \in J} x_{ij} \hat{\beta}_j^t| \quad (3.6)$$

接下来给出拆解聚类的准则: 设  $t$  步的聚类集合为  $C^t$ , 该步解为  $\hat{\beta}_t$ , 对于  $k = 1, \dots, K^t$ , 计算  $\theta_i = y_i - \sum_{j \in J} x_{ij} \hat{\beta}_j^t$ , 1) 若对于任意  $i \in C_k^t$ ,  $\theta_i$  有相同的符号, 那么该聚类  $C_k^t$  将保留到下次迭代, 即  $C^{t+1} \leftarrow C^{t+1} \cup C_k^t$ , 见图 3.2;

2) 若不满足上述条件, 那么根据  $\theta_i$  符号异同, 将  $C_k^t$  分成两个集合,  $C_{k+}^t = \{i \in C_k^t | \theta_i > 0\}$ ,  $C_{k-}^t = \{i \in C_k^t | \theta_i < 0\}$ , 见图 3.2b, 这两个集合在下一步形成新的聚类, 即  $C^{t+1} \leftarrow C^{t+1} \cup \{C_{k+}^t, C_{k-}^t\}$ , 见图 3.2c。

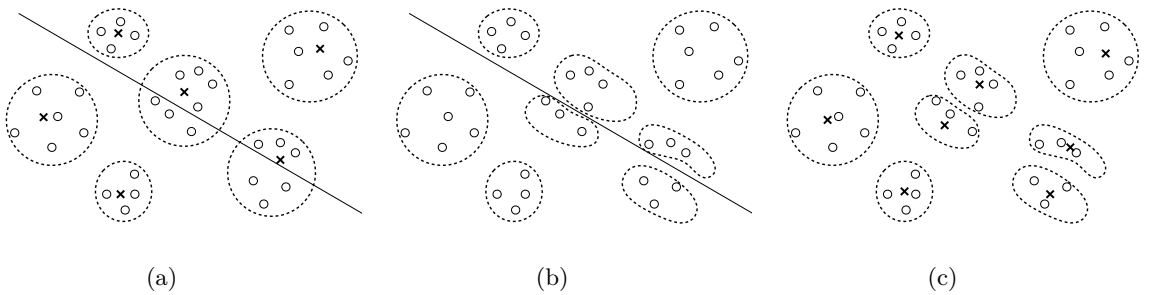


图 3.2: 聚类拆解步骤示意图

结合算法 2.1, 到这里已经给出了完成聚类——迭代拆解最小绝对值回归的所有计算步骤, 当聚类无法继续划分时, 迭代终止。

下面证明最后一次迭代的解  $\hat{\beta}_j^T$  就是(3.4)的解  $\beta^*$ ,

$$\begin{aligned}
 E^* &= \sum_{i \in I} |y_i - \sum_{j \in J} x_{ij} \beta_j^*| = \sum_{k \in K^t} \sum_{i \in C_k^t} |y_i - \sum_{j \in J} x_{ij} \beta_j^*| \\
 &\geq \sum_{k \in K^t} | \sum_{i \in C_k^t} y_i - \sum_{j \in J} x_{ij} \beta_j^* | = \sum_{k \in K^t} |C_k^t| |y_k^t - \sum_{j \in J} x_{kj} \beta_j^*| \\
 &\geq \sum_{k \in K^t} |C_k^t| |y_k^t - \sum_{j \in J} x_{kj} \hat{\beta}_j^t| = \sum_{k \in K^t} | \sum_{i \in C_k^t} y_i - \sum_{j \in J} \sum_{i \in C_k^t} x_{ij} \hat{\beta}_j^t | \\
 &= \sum_{k \in K^t} \sum_{i \in C_k^t} |y_i - \sum_{j \in J} x_{ij} \hat{\beta}_j^t| = \sum_{i \in I} |y_i - \sum_{j \in J} x_{ij} \hat{\beta}_j^t| = E^t
 \end{aligned}$$

因为  $\hat{\beta}_t$  是(3.4)的可行解, 又显然  $E^* \leq E^t$ , 这就证明了  $E^* = E^t$ , 注意到  $\sum_{k \in K^t} |C_k^t| |y_k^t - \sum_{j \in J} x_{kj} \hat{\beta}_j^t|$  就是  $F^t$ , 因此  $E^t = F^t$ 。因此最后一次迭代  $F^T$  的最优解  $\hat{\beta}_j^T$  就是原问题的最优解。

### 3.2.3 优化带有惩罚项的最小绝对值回归

前面给出的算法已经在很大程度上优化了在处理高维宏观经济变量时, 最小绝对值回归计算性能的问题。而在高维宏观经济实证研究中, 变量的维数众多, 在建立最小绝对值回归模型时, 还需要考虑到变量筛选问题。

对于(3.3), 为了进行变量的筛选, 我们需要对估计量  $\beta^*$  的维数做出约束, 一般来说, 我们通常采用以下 3 三种方法: 1)  $\|\beta\|_0 = p$ , 可以直接选择入选变量的个数; 2)  $\|\beta\|_2 < s$ , 该法又称为岭回归; 3)  $\|\beta\|_1 < s$ , 即为常用的  $L_1$  正则化方法。通过加入新的约束, 问题的形式也发生了变化。

Park 等人于 2019 年的研究中给出了进一步的结论, 对于问题

$$E^* = \min_{B \in \phi} \|Y - f(X, B)\|_{L_1} \quad (3.7)$$

其中  $Y$  为模型响应变量数据矩阵,  $X$  为解释变量数据矩阵,  $B$  为模型系数矩阵,  $\phi$  为模型系数的约束条件。  $f$  为任一目标函数, 若  $f$  满足条件

$$f(B, WX) = Wf(B, X) \quad (3.8)$$

其中  $W$  为一权重矩阵。那么就可以通过聚类——迭代拆解算法求原优化问题的最优解。

而在岭回归和 LASSO 中, 都通过给目标函数加上惩罚项来进行求解, 岭回归对应的惩罚项  $s \sum_{i=1}^p \beta_i^2$ ,  $L_1$  正则化对应的惩罚项为  $s \sum_{i=1}^p |\beta_i|$ , 容易发现, 加入惩罚项后目标函数仍然满足(3.8)。因此, 对于需要进行变量筛选的高维最小绝对值回归问题, 我们可以修改目标函数, 通过算法 2.1 解决。

## §3.3 一种基于替代变量的估计方法

聚类——迭代拆解算法是通过有效减少  $L_1$  目标函数最小化的问题规模来提升求解性能, 每一次迭代仍需要通过线性规划方法求解  $L_1$  目标函数的最小化问题。Weidong Liu 等人于 2020 年提出了一种基于替代变量的估计方法, 通过替代变量将分位数损失函数转化为  $L_2$  损失函数, 避免使用线性规划求解分位数损失, 因此可以显著提升一般分位数回归的求解性能。

考虑到最小绝对值回归为其特殊情形, 且该估计量具有良好的性质, 可以在一定程度上解决最小绝对值回归的性能问题。下面我们介绍如何将该估计应用在最小绝对值回归的场景下, 并且给出具体的计算步骤。



### 3.3.1 基于替代变量的迭代算法

考虑(3.3)，对目标函数稍作形式变换，其中  $\rho(x) = x(0.5 - \mathbb{I}[x \leq 0])$ ， $\mathbb{I}(x)$  为指示函数。

$$\beta^* = \arg \min_{\beta \in \mathbb{R}^p} \mathbb{E}|Y - \mathbf{X}^T \beta| = \arg \min_{\beta \in \mathbb{R}^p} \mathbb{E} \rho(Y - \mathbf{X}^T \beta) \quad (3.9)$$

若已知  $n$  i.i.d. 的样本  $(\mathbf{X}_i, Y_i)$  ( $1 \leq i \leq n$ )，令  $\hat{\beta}$  为  $\beta^*$  的估计量，

$$\hat{\beta} = \arg \min_{\beta \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^n \rho(Y_i - \mathbf{X}_i^T \beta). \quad (3.10)$$

一般地，我们使用牛顿迭代法求解某随机优化问题：

$$\beta^* = \arg \min_{\beta \in \mathbb{R}^p} \mathbb{E}[G(\beta; \mathbf{X}, Y)] \quad (3.11)$$

其中  $G(\beta; \mathbf{X}, Y)$  是损失函数， $\mathbf{X}$  和  $Y$  分别是  $p$  维自变量和一元响应变量， $\beta$  为回归系数。使用牛顿-拉弗森迭代来求解，

$$\tilde{\beta}_1 = \beta_0 - \mathbf{H}(\beta_0)^{-1} \mathbb{E}[g(\beta; \mathbf{X}, Y)] \quad (3.12)$$

其中  $\beta_0$  是一个初始估计， $g(\beta; \mathbf{X}, Y)$  为损失函数  $G(\beta; \mathbf{X}, Y)$  关于  $\beta$  的梯度。

$\mathbf{H}(\beta) := \partial \mathbb{E}[g(\beta; \mathbf{X}, Y)] / \partial \beta$  表示  $\mathbb{E}G(\beta; \mathbf{X}, Y)$  的海赛矩阵。特别地，我们这里考虑损失函数为  $L_1$  损失的特殊情形，

$$G(\beta; \mathbf{X}, Y) = \rho(Y - \mathbf{X}^T \beta) \quad (3.13)$$

在(3.13)的条件下， $g(\beta; \mathbf{X}, Y) = \mathbf{X}(\mathbb{I}[Y - \mathbf{X}^T \beta < 0] - 0.5)$ 。

并且  $\mathbf{H}(\beta) = \mathbb{E}(\mathbf{X} \mathbf{X}^T f(\mathbf{X}^T(\beta - \beta^*)))$ ，这里  $f(x)$  是噪声  $e$  的密度函数。当初始估计量  $\beta_0$  和  $\beta^*$  很接近时， $\mathbf{H}(\beta_0)$  就会很接近  $\mathbf{H}(\beta^*) = \Sigma f(0)$ ，这里  $\Sigma = \mathbb{E} \mathbf{X} \mathbf{X}^T$  是  $\mathbf{X}$  的协方差矩阵。使用  $\mathbf{H}(\beta^*)$  替换  $\mathbf{H}(\beta_0)$ ，可得

$$\beta_1 = \beta_0 - \mathbf{H}(\beta^*)^{-1} \mathbb{E}[g(\beta; \mathbf{X}, Y)] = \beta_0 - \Sigma^{-1} f^{-1}(0) \mathbb{E}[g(\beta_0; \mathbf{X}, Y)] \quad (3.14)$$

在  $\beta^*$  对  $\mathbb{E}[g(\beta_0; \mathbf{X}, Y)]$  进行泰勒展开，

$$\begin{aligned} \mathbb{E}[g(\beta_0; \mathbf{X}, Y)] &= \mathbf{H}(\beta^*)(\beta_0 - \beta^*) + O(|\beta_0 - \beta^*|_2^2) \\ &= \Sigma f(0)(\beta_0 - \beta^*) + O(|\beta_0 - \beta^*|_2^2) \end{aligned}$$

结合(3.14)，可以得到

$$\begin{aligned} |\beta_1 - \beta^*|_2 &= |\beta_0 - \Sigma^{-1} f^{-1}(0)(\Sigma f(0)(\beta_0 - \beta^*) + O(|\beta_0 - \beta^*|_2^2)) - \beta^*|_2 \\ &= O(|\beta_0 - \beta^*|_2^2) \end{aligned}$$

因此，如果我们得到一个  $\beta^*$  的一致估计量  $\beta_0$ ，我们就可以通过(3.14)得到偏误更小的估计。

下面我们将(3.14)转化成一个最小二乘问题。首先我们重写该式，

$$\begin{aligned} \beta_1 &= \Sigma^{-1}(\Sigma \beta_0 - f^{-1}(0) \mathbb{E}[g(\beta_0; \mathbf{X}, Y)]) \\ &= \Sigma^{-1} \mathbb{E}[\mathbf{X} \{ \mathbf{X}^T \beta_0 - f^{-1}(0)(\mathbb{I}[Y \leq \mathbf{X}^T \beta_0] - 0.5) \}] \end{aligned} \quad (3.15)$$

这里我们定义一个新的响应变量  $\tilde{Y}$ ,

$$\tilde{Y} = \mathbf{X}^T \boldsymbol{\beta}_0 - f^{-1}(0)(\mathbb{I}[Y \leq \mathbf{X}^T \boldsymbol{\beta}_0] - 0.5) \quad (3.16)$$

那么  $\boldsymbol{\beta}_1 = \boldsymbol{\Sigma}^{-1} \mathbb{E}(\mathbf{X} \tilde{Y})$  就是线性回归问题  $\tilde{Y} = \mathbf{X}^T \boldsymbol{\beta}$  的最优回归系数, 即

$$\boldsymbol{\beta}_1 = \arg \min_{\boldsymbol{\beta} \in \mathbb{R}^p} \mathbb{E}(\tilde{Y} - \mathbf{X}^T \boldsymbol{\beta})^2 \quad (3.17)$$

给定 *i.i.d.* 样本  $(\mathbf{X}_i, Y_i)$ , 构造

$$\tilde{Y}_i = \mathbf{X}_i^T \hat{\boldsymbol{\beta}}_0 - \hat{f}^{-1}(0)(\mathbb{I}[Y_i \leq \mathbf{X}_i^T \hat{\boldsymbol{\beta}}_0] - 0.5) \quad (3.18)$$

其中  $\hat{\boldsymbol{\beta}}_0$  为  $\boldsymbol{\beta}^*$  的一个初始估计,  $\hat{f}(0)$  为  $f(0)$  的一个估计

$$\hat{\boldsymbol{\beta}} = \arg \min_{\boldsymbol{\beta} \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^n (\tilde{Y}_i - \mathbf{X}_i^T \boldsymbol{\beta})^2 \quad (3.19)$$

这里我们选择某估计量作为  $\hat{\boldsymbol{\beta}}_0$ , 并且采用  $f(0)$  的核密度估计作为  $\hat{f}(0)$ ,

$$\hat{f}(0) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{Y_i - \mathbf{X}_i^T \hat{\boldsymbol{\beta}}_0}{h}\right)$$

其中  $K(x)$  为核函数,  $h \rightarrow 0$  是带宽, 本方法在每次迭代都需要调整带宽 [7]。

只要给定的初始值  $\hat{\boldsymbol{\beta}}_0$  是  $\boldsymbol{\beta}^*$  的一致估计量, 那么  $\hat{\boldsymbol{\beta}}$  就将会是一个更加接近  $\boldsymbol{\beta}^*$  的新的估计, 并且(3.19)为最小二乘问题, 其计算十分简便。

我们当然可以将  $\hat{\boldsymbol{\beta}}$  作为(3.14)的初始值, 这样继续构造替代变量进行迭代, 最终收敛到  $\boldsymbol{\beta}^*$ 。算法 2 给出了使用替代变量估计方法的计算步骤。

---

算法 2.2 使用替代变量迭代算法方法求解最小绝对值回归问题 (SVN, substitute variable newton method)

---

输入:  $Y$  和  $\mathbf{X}$  的样本  $\mathbf{Y} = (Y_1, Y_2, \dots, Y_n)$ ,  $\mathbf{X} = (\mathbf{X}_1^T, \mathbf{X}_2^T, \dots, \mathbf{X}_n^T)$ , 迭代次数  $T$ , 核函数  $K$ , 依赖于迭代次数的带宽  $h_t (t = 1, \dots, T)$

初始化: 给出初始相合估计量  $\hat{\beta}^0$ , 将样本划分成  $J$  个均等子集, 样本量均为  $m$ , 为了给出最小绝对值回归估计量的相合估计, 我们取任一子集里面的数据, 直接使用最小一乘法估计  $\hat{\beta}_{(0)}$ 。

对于  $t = 1, \dots, T$ : 取新的数据子集, 在该数据集上

1. 计算  $\hat{f}^t(0)$ ,

$$\hat{f}^t(0) = \frac{1}{mh} \sum_{i=1}^m K\left(\frac{Y_i - \mathbf{X}_i^T \hat{\beta}^{t-1}}{h_t}\right)$$

2. 计算  $\tilde{Y} = (\tilde{Y}_1, \tilde{Y}_2, \dots, \tilde{Y}_m)$ ,

$$\tilde{Y}_i = \mathbf{X}_i^T \hat{\beta}^{t-1} - \hat{f}^t(0)^{-1} (\mathbb{I}[Y_i \leq \mathbf{X}_i^T \hat{\beta}^{t-1}] - 0.5)$$

3. 计算  $\hat{\beta}^t$ ,

$$\hat{\beta}^t = \arg \min_{\beta \in \mathbb{R}^p} \frac{1}{m} \sum_{i=1}^m (\tilde{Y}_i - \mathbf{X}_i^T \beta)^2$$

输出:  $\hat{\beta}^T$

---

### 3.3.2 优化带有惩罚项的最小绝对值回归

在高维相依自变量的情形下, 我们只需在(3.17)加入惩罚项即可, 例如我们使用  $L_1$  正则化, 那么

$$\beta_{1,\lambda} = \arg \min_{\beta \in \mathbb{R}^p} \mathbb{E}(\tilde{Y} - \mathbf{X}^T \beta)^2 + \lambda |\beta| \quad (3.20)$$

对应的, 最终估计量计算如下

$$\hat{\beta} = \arg \min_{\beta \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^n (\tilde{Y}_i - \mathbf{X}_i^T \beta)^2 + \lambda |\beta| \quad (3.21)$$

(3.21)是著名的 LASSO 问题, 已经有了快速解决的算法。因此, 对于高维情形, 算法 2 仅需稍作改动。

## §3.4 模拟实验

本章前面两节介绍了聚类——迭代拆解算法和一种基于替代变量的迭代算法, 前者通过减小问题规模、逐步逼近最优解的方法来提升计算性能, 而后者通过变量替换的方法将最小绝对值回归问题转化为最小二乘问题, 在给定一个相合估计条件下逼近最优解。两种方法都可以处理带有惩罚项和不带有惩罚项的最小绝对值回归问题, 并且前文已经给出了两种算法实现的具体步骤。

本节将通过一个数值模拟实验来比较和分析两种方法的性能表现, 分析算法的收敛情况, 并对算法的解的确性进行观察。

## 3.4.1 数据准备

为测试两种算法求解一般最小绝对值回归问题的性能，设定模拟数据来自下面的模型：

$$Y_i = \mathbf{X}_i^T \boldsymbol{\beta} + e_i, i = 1, 2, \dots, n$$

其中  $\mathbf{X}_i = (1, X_{i,1}, \dots, X_{i,p-1})$  为  $p$  维随机向量。假设  $\mathbf{X}$  的各维独立并服从正态同分布， $\mathbf{e}$  为高斯噪声。我们设置不同  $p, n$  的组合用来测试算法的计算性能，并观察在不同的噪声分布下，估计结果的准确性。对于系数  $\boldsymbol{\beta}$ ，设  $l$  为其  $L_0$  范数，取

$$\boldsymbol{\beta} = (\frac{10}{l}, \frac{20}{l}, \dots, \frac{10(l-1)}{l}, 10, 0, \dots, 0)$$

算法 2.1（以下简称 SVN）的每步最佳带宽依赖于数据集的性质，参考 weidong liu 等，这里给出

$$h^t = \sqrt{\frac{s \log n}{n}} + s^{-1/2} (\frac{s^2 \log n}{10m})^{(t+1)/2},$$

核函数选取高斯核函数，其中  $s$  的理论最优取值和  $l$  有关，而本次实验中取常数。

对于聚类——迭代拆解算法，我们这里选择初始聚类方法如下：首先从原始数据中少量采样（本实验取  $\max\{0.5\%n, m\}$ ），通过最小绝对值回归给出系数估计  $\boldsymbol{\beta}^{init}$ ，对每一个原始数据点，计算其在当前模型系数下的残差，然后进行 K-means 聚类得到  $C^0$ 。

本次实验最小绝对值回归性能测试的对比基准方法是：线性规划内点法（以下简称 LP）；

我们给出了 SVN 算法和 AID 算法的 Python3 实现，数值计算基于 Numpy 包，作为基准的 LP 使用 Scipy 数值计算包的对应实现。

## 3.4.2 实验结果

我们采用最终估计值和真实值差的  $L_2$  范数来衡量准确性。在实验结果中我们着重观察算法的收敛性、算法的准确性和计算性能。

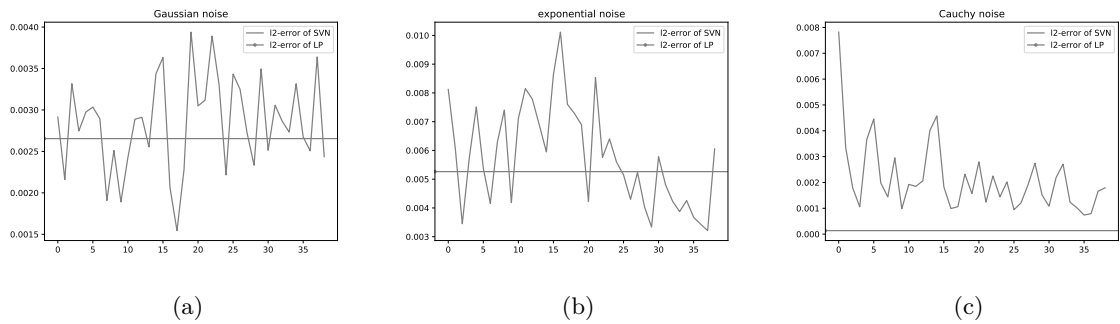
首先我们在最小绝对值回归模型下，比较三种算法的性能表现。表 4 中， $r^0$  表示 AID 算法初始聚类  $K_0/n$  的值，而  $r^T$  表示算法终止时  $K_T/n$  的取值。需要注意的是，对于 AID 算法， $T$  表示算法终止时经历的迭代次数， $Time$  表示算法终止时运算的 cpu 时间。而对于 SVN 算法， $T$  表示在当前带宽选择下使其每估计值达到稳定的迭代次数， $Time$  表示全部数据参与迭代完毕经历的 cpu 时间（处理机情况：苹果 M1 芯片 8 核，8GB 内存，OSX，CPython 解释器 3.8.2 Arm 版）。

n	p	AID				SVN		LP
		$r^0(\%)$	$r^T(\%)$	T	Time(sec)	T	Time(sec)	Time(sec)
20000	10	0.05	2.90	6	2	3	0.38	0.20
20000	50	0.15	12.00	12	13	7	6	3
20000	100	0.15	12.00	12	38	11	27	21
20000	200	0.66	21.55	9	118	8	76	126
20000	500	0.80	28.00	11	535	3	322	813
40000	10	0.05	2.90	6	3	3	0.74	0.23
40000	50	0.15	12.00	12	31	7	14	7
40000	100	0.15	12.00	12	84	8	56	48
40000	200	0.66	21.55	9	210	8	154	258
40000	500	0.80	28.00	11	1965	3	1143	3446
200000	100	0.80	11.00	11	134	3	361	496
200000	200	0.80	28.00	11	2476	3	1631	3329

表 4: 高斯噪声下三种估计算法的性能比较

从表 4 可以看出在  $n, m$  较大场合下, AID 算法和 SVN 算法均在性能上领先, 尤其是 SVN 算法具有更好的性能表现, 因为以上时间为训练全部数据需要的时间, 而在下面可以看到 SVN 算法具有相当快的收敛速度, 在后续性能比较中, 我们会将 SVN 算法的停止条件设定为相邻两次迭代的  $\hat{\beta}^t$  夹角小于某个的小正数, 能够进一步缩短 SVN 算法的时间。

接下来我们观察不同噪声分布对估计准确性的影响, 这里主要观察 SVN 算法的准确性 (注意 AID 算法的解就是 LP 最优解, 因此它的准确性是不言自明的)。图 3.3 显示了不同的噪声分布下 SVN 算法的表现, 其中横轴代表迭代的次数。可以看到, 随着迭代进行, SVN 算法估计值随样本信息不断调整, 始终保持在和 LP 解相差不大的水平上; SVN 算法仅在柯西噪声这种极端重尾场景下, 表现稍差。

图 3.3: SVN 算法在各种噪声下的表现。  $n = 20000, p = 50, s = 20$ 

主要到算法 3.2 中我们使用最小绝对值回归估计量作为初始估计。在柯西噪声下, 我们考虑了最小二乘估计量作为初始估计的情况, 来观察 SVN 算法的准确性。

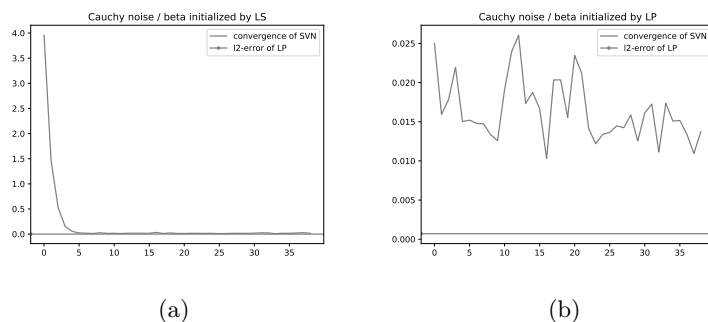


图 3.4: 使用最小二乘估计进行初始化。横轴表示迭代次数, 纵轴表示估计的误差平方和。

图 3.4表明, 选择最小二乘估计量作为初始值, SVN 算法依然能够很快收敛。这启发我们在使用最小绝对值回归时, 如果最小二乘估计量是相合的, 那么也可以通过最小二乘估计量产生初始估计, 这样一来, 在计算中可做到完全避免直接求解最小绝对值回归问题。

需要说明的是, SVN 算法的收敛速度和带宽选择有很大关系。在选择了足够小的带宽时, SVN 算法可快速收敛。图 3.5展示了 SVN 算法的收敛情况。观察到 SVN 算法在带宽较小时收敛速度表现较好, 这也符合该算法提出的理论基础。在带宽选择较小 ( $s$  较小) 时, SVN 算法能够迅速收敛, 当然, 如果我们选择了很大的带宽那么算法不能收敛。

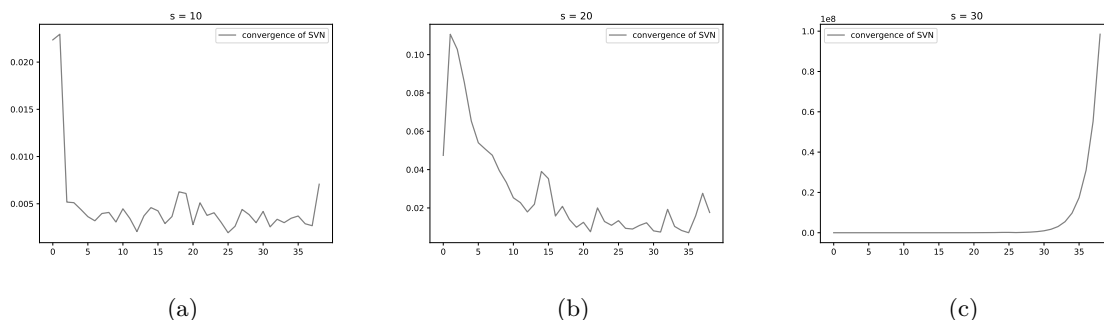


图 3.5: SVN 算法的收敛情况。横轴表示迭代次数, 纵轴表示相邻两次  $\beta$  估计之差的平方和。

然而在核密度估计中, 一个更大的带宽意味着对该点密度估计包含了更多的样本信息, 选择过很小的带宽, 可能造成对核密度估计不准, 在应用中需要进行合理选择。

下面我们采用最小二乘估计进行初始化, 并且设置终止条件, 对算法的性能作进一步的比较。这次我们将噪声设为正态噪声, 但将其中 10% 的数据添加柯西噪声, 部分性能对比结果见表 5。

s	n	p	Iteration	Time(SVN)	Time(SVN-full)	Time(LP)	Time(AID)	l2-error(SVN)	l2-error(LS)	l2-error(LP)
5	20000	10	12	0.12	0.42	0.17	2	0.0023	0.7304	0.0024
5	20000	50	8	1	7	3	13	0.0039	3.1548	0.0020
5	20000	100	14	10	28	19	41	0.0059	15.0373	0.0033
5	20000	200	27	81	130	170	124	0.0188	151.4039	0.0028
5	20000	500	40	437	437	1119	698	0.0259	245.0070	0.0026

表 5: 部分性能结果比对

### 3.4.3 结论

AID 算法和 SVN 算法都可以用来求解最小绝对值回归，并且两者在数据规模较大的情况下，都明显快于线性规划。SVN 算法虽然不能保证取得最优解，但是它可以完全避免线性规划计算，因此在计算性能上很有优势；而 AID 算法在小规模数据下不具有性能优势，然而在数据集很大时优势明显，并且 AID 算法最终的解就是对全部数据求解线性规划的结果，因此它具有更好的稳健性。

我们的结论是，在数据能够服从线性模型时，对最小绝对值回归的计算推荐使用 SVN 算法，该算法具有很快的速度。而在一般求解具有  $\min \sum_{i=1}^n |y_i - \mathbf{X}_i^T \boldsymbol{\beta}|$  形式，但是数据能否服从线性模型未知时，可以先使用最小二乘法拟合，并对模型做出检验然后进行算法选择。对于必须用线性规划来求解的形式类似问题，那么建议使用 AID 算法来进行优化。

我们没有进行实验对比带有惩罚项的情况，原因在于：惩罚项主要用来对线性模型中用于系数稀疏化处理，避免直接拟合的模型具有共线性，在实际应用中普遍使用  $L_1$  惩罚项。我们通常已经假定了数据服从线性模型，所以此时 SVN 算法适用。并且前面提到 LASSO 算法具有很高的性能，而增加额外约束的线性规划在速度上就更不具有优势，所以对于带有  $L_1$  惩罚项的最小绝对值回归，我们始终推荐使用 SVN 算法。

## §3.5 SVN-AID 算法

我们已经研究了 AID 和 SVN 两种方法，注意到 AID 算法的基础思想是具有普适性的。我们不妨再次分析使用 AID 算法解决最小绝对值回归问题的实质：为了避免在整个数据集上一次性求解，我们使用聚类的方法，希望通过尽可能少的计算来利用样本的信息。不妨将前文提到的 AID 算法称为 AID-LP 算法，它成功减少了解决最小绝对值回归问题的 LP 计算规模。

在 AID 算法求解最小绝对值回归时，假设在第  $t$  次迭代中我们得到了  $\boldsymbol{\beta}^t$ ，然后根据  $\boldsymbol{\beta}^t$  对现有的聚类进行拆解。在下一步，我们重新计算该问题，直到达到问题最优解。而  $\boldsymbol{\beta}^t$  此时已经成功分离了许多聚类，而这些聚类在下一个步骤中仍然全部需要进入计算，因为上一步的参数信息  $\boldsymbol{\beta}^t$  在下一步计算时被舍弃了。

而 SVN 算法，正是通过利用上一步的参数信息来构造代理变量简化计算。我们不妨结合两种算法的优点，将 SVN 算法和 AID 算法结合起来，进一步优化对最小绝对值回归问题的求解。

### 3.5.1 算法步骤

以下我们给出 SVN-AID 的步骤：1) 初始化：随机采样，进行最小绝对值回归，该步骤使用 LP，得到  $\hat{\boldsymbol{\beta}}^0$ 。根据各个样本点对该模型的拟合残差进行聚类。2) 记上一次迭代得出的系数估计为  $\hat{\boldsymbol{\beta}}^t$ ，对当前聚类按照规则拆解。本步骤返回被拆解的聚类组成的集合，舍弃上一步正确分类的聚类。3) 根据上一步返回的聚类集合，构造新数据集，并且按照(3.18)构造代理变量数据集，并由(3.19)得出本轮估计。

具体见算法 4.2，该算法通过舍弃正确分类的聚类，不断缩小问题规模，迭代优化上一步得出的参数估计值来进行对最优解的近似，该算法在理论上可以极大减小 AID 算法的计算量，并且在舍弃大量样本点的同时，充分利用了前面步骤得出的参数信息。

## 算法 4.2 SVN-AID 算法

输入:  $Y$  和  $X$  的样本  $Y = (Y_1, Y_2, \dots, Y_n)$ ,  $X = (X_1^T, X_2^T, \dots, X_n^T)$ , 迭代次数  $T$ , 核函数  $K$ , 依赖于迭代次数的带宽  $h_t (t = 1, \dots, T)$

初始化: 给出初始估计,  $\hat{\beta}^0$ , 对每一个样本点  $(X_i, Y_i)$  根据  $Y_i - X_i^T \hat{\beta}$  的结果进行 K-means 聚类, 得到聚类结果  $C^0 = \{C_1^0, C_2^0, \dots, C_K^0\}$ 。

对于  $t = 1, \dots, T$ :

1) 根据上一轮  $\hat{\beta}^{t-1}$ , 计算带宽  $h^t$  和  $f(\hat{0})^t = \frac{1}{K^t h^t} \sum_{i=1}^{K^t} K(Y_i - X_i^T \hat{\beta}^{t-1})$ 。

据(3.18)在当前聚类上构造替代变量  $\tilde{Y} = (\tilde{Y}_1, \dots, \tilde{Y}_{K^t})$ ; 求解

$$\hat{\beta}_t = \arg \min_{\beta \in \mathbb{R}^p} \frac{1}{K^t} \sum_{k=1}^{K^t} |C_k^t| (\tilde{Y}_k - X_k^T \beta)^2$$

2) 对于  $k = 1, \dots, K^t$ , 计算  $\theta_i = y_i - \sum_{j \in J} x_{ij} \hat{\beta}_j^t$ , 根据  $\theta_i$  符号异同, 将  $C_k^t$  分成两个集合,  $C_{k+}^t = \{i \in C_k^t | \theta_i > 0\}$ ,  $C_{k-}^t = \{i \in C_k^t | \theta_i < 0\}$ , 这两个集合在下一步形成新的聚类, 即  $C^{t+1} \leftarrow C^{t+1} \cup \{C_{k+}^t, C_{k-}^t\}$ 。

输出:  $\beta^T$

## 3.5.2 数值模拟实验

## §3.6 本章小结

本章首先简要介绍了  $L_1$  范数的概念, 列举了一些其在经济数据分析领域的应用。随后, 讨论了最小绝对值回归的稳健性和其求解方法。

最小绝对值回归已经广泛应用在高维宏观经济数据的处理中, 但是它在应用中存在一定的性能问题。本章介绍了两种可以应用于最小绝对值回归的优化手段: 聚类——迭代拆解算法和基于替代变量的估计方法。进行了数值模拟实验, 分析了其优缺点, 并给出了其在实际应用中的几点建议。

聚类——迭代拆解算法是一种非常好的思想, 它结合了机器学习中聚类的方法, 聚类的作用可以将信息进行提纯, 在经过处理后的数据集上解决原始问题对应的加权优化问题, 可以避免直接解决原始问题而耗费庞大的计算量。聚类——迭代拆解算法可以应用于解决最小绝对值回归问题, 它在一定程度上减少了线性规划的计算量, 并且在理论上具有最优解, 即它的解一定可以和直接对所有数据求解进行线性规划问题一样好。

另外, 本章介绍了另外一种巧妙的算法——一种基于替代变量的迭代算法, 该算法通过构造替代变量, 从而将原来需要通过线性规划解决的最小绝对值回归问题转化为解决替代变量的最小二乘问题。这样一来, 计算的复杂度就大大降低。该方法在实际应用中性能表现十分优秀, 但是需要注意带宽的选择。

聚类——迭代拆解算法对和最小绝对值回归目标函数相同的任何优化问题均适用; 然而 SVN 算法在其理论上需要假设线性模型成立, 需要结合我们面临的实际问题酌情使用。

最后, 我们在本章对最小绝对值回归性能优化的研究, 会对后续章节中优化  $L_1$  范数主成分分析法估计近似因子模型的求解算法有推动作用。



## 第四章 近似因子模型的 $L_1$ 范数主成分估计算法研究

本章继续讨论  $L_1$  范数主成分分析，重点在于优化前文介绍的交替凸优化算法。 $L_1$  主成分分析虽然在稳健性上有很大的优势，但是其求解需要比  $L_2$  主成分分析多得多的时间。

在第三中的研究中，已经重点讨论了优化最小绝对值回归的算法，注意到交替凸优化算法中包含了最小绝对值回归问题，因此可以充分利用得出的结论，尝试将两种算法应用于改良交替凸优化算法。之后通过一个模拟实验和对一个高频金融数据的主成分分析的测试，测试了算法的性能。

至此我们已经对问题  $P_3$  做出了较为深入的讨论，而问题  $P_4$  同样值得探究，它是  $L_1$  主成分分析的另一重要研究方向。我们将介绍一种利用贪婪策略求解  $P_4$  算法，并通过模拟实验证明算法的有效性。

在第二章，我们将交替凸优化算法用于近似因子模型的估计，并且通过实证研究论证了  $L_1$  范数主成分分析得出的因子同样适用于宏观经济预测中，并且有良好的预测表现。本章的最后，我们将求解  $P_4$  来得到的因子估计量同样在国内主要月度宏观经济数据预测场景下进行了测试，并给出了一些结论与建议。

### §4.1 改进的 $L_1$ 主成分分析交替凸优化算法

在第二章中，我们提出了求解  $L_1$  主成分分析的交替凸优化算法；交替凸优化算法的计算量主要集中在(2.13)和(2.14)，注意到在数据服从因子模型的假设下，(2.14)是一个最小绝对值回归问题。因此我们可以尝试使用第三章中基于替代变量的估计算法来代替线性规划求解(2.14)。

#### 4.1.1 基于替代变量算法的交替凸优化算法改进

假设

#### 4.1.2 基于聚类——迭代拆解算法的交替凸优化算法改进

#### 4.1.3 数值模拟

#### 4.1.4 高频金融数据的主成分分析

### §4.2 求解特征空间 $L_1$ 范数最大化的贪婪算法

我们已经在第三章中研究了  $P_3$ ，这里我们首选给出问题  $P_4$  的规范化表述：

$$P_4 : \hat{\mathbf{A}} = \arg \max_{\mathbf{A}} \|\mathbf{A}^T \mathbf{X}\|_{L_1} = \arg \max_{\mathbf{A}} \sum_{i=1}^n \sum_{k=1}^m \left| \sum_{j=1}^p a_{jk} x_{ij} \right|, \text{ 其中 } \mathbf{A}^T \mathbf{A} = \mathbf{I}_m \quad (4.1)$$

这里我们介绍一种求解  $P_4$  的算法。求解 4.1 时，由于对于不同的  $m$ ，最优  $\mathbf{a}_j^*$  会发生变化，因此想要得出 4.1 的全局最优解是非常复杂的。下面介绍 Kwak 于 2009 年提出的一种算法，它将问题分解一个个  $m = 1$  的子问题，然后用贪婪策略获得全局解。该算法直观、简洁并且易于实现。

#### 4.2.1 求解子问题

首先考虑  $m = 1$  的情况，该问题寻找(4.2)的最优解。

$$\mathbf{a}^* = \arg \max_{\mathbf{a}} \|\mathbf{a}^T \mathbf{X}\|_{L_1} = \arg \max_{\mathbf{a}} \sum_{i=1}^n |\mathbf{a}^T \mathbf{x}_i|, \text{ 其中 } \|\mathbf{a}\|_{L_2} = 1 \quad (4.2)$$

算法 4.1 给出了(4.2)的求解算法。

---

算法 4.1:  $m = 1$  时的  $P_4$  求解算法

---

初始化: 取  $\mathbf{a}$  的任意初始值;  $\mathbf{a}^0 \leftarrow \mathbf{a}^0 / \|\mathbf{a}^0\|$ 。

对于  $t = 1, \dots, T$ :

1. 对所有的  $i \in \{1, \dots, n\}$ , 若  $\mathbf{a}^{0T} \mathbf{x}_i < 0$ , 令  $p_i^t = -1$ , 否则  $p_i^t = 1$ 。
  2. 令  $t \leftarrow t + 1$  并且  $\mathbf{a}^t = \sum_{i=1}^n p_i^{t-1} \mathbf{x}_i$ ;  $\mathbf{a}^t \leftarrow \mathbf{a}^t / \|\mathbf{a}^t\|_{L_2}$ 。
  3. 检查是否收敛:
    - 1) 如果  $\mathbf{a}^t \neq \mathbf{a}^{t-1}$ , 回到步骤 2;
    - 2) 若存在某个  $i$  使得  $\mathbf{a}^T \mathbf{x}_i = 0$ , 令  $\mathbf{a}^t \leftarrow (\mathbf{a}^t + \Delta \mathbf{a}) / \|\mathbf{a}^t + \Delta \mathbf{a}\|_{L_2}$  并回到步骤 2, 这里  $\Delta \mathbf{a}$  为一个很小的随机向量;
    - 3) 若前面两个都不符合, 令  $\mathbf{a}^* = \mathbf{a}^t$ , 算法返回。
- 

按照算法 4.1 得出的  $\mathbf{a}^*$  是  $\mathbf{x}_1, \dots, \mathbf{x}_i$  的线性组合, 即  $\mathbf{a}^t \propto \sum_{i=1}^n p_i^{t-1} \mathbf{x}_i$ , 故其拥有旋转不变性。注意到的  $\mathbf{a}^*$  为(4.2)的最优解, 它可能不是(4.1)的全局最优解。 $\mathbf{a}^0$  可以任意设置, 因此可以采用  $L_2$  主成分作为其初始值, 从而提高解趋向全局最优的可能性。

#### 4.2.2 贪婪策略求解主成分

前文已经给出了获得  $m = 1$  时  $P_4$  最优解的算法, 我们可以通过贪婪策略来求解  $P_4$ 。贪婪策略是一种在问题的动态求解过程中, 对问题所处的每一个当前状态都寻求其最优解, 从而期望能够逼近全局最优解的算法设计思想。下面给出利用贪婪策略求解  $P_4$  的步骤,

---

算法 4.2: 求解特征空间  $L_1$  范数最大化的贪婪算法

---

输入:  $\mathbf{a}^0, \{\mathbf{x}_i^0 = \mathbf{x}_i\}_{i=1}^n$ 。

对于  $j = 1, \dots, m$ :

对所有的  $i \in \{1, \dots, n\}$ ,  $\mathbf{x}_i^j = \mathbf{x}_i^{j-1} - \mathbf{a}^{j-1}(\mathbf{a}^{(j-1)T} \mathbf{x}_i^{j-1})$ ,

通过算法 4.1 求解  $\mathbf{a}^j$ , 令  $\mathbf{X}^j = [\mathbf{x}_1^j, \mathbf{x}_2^j, \dots, \mathbf{x}_i^j]$ 。

---

#### 4.2.3 模拟与实证

### §4.3 本章小结

本章进一步讨论了  $L_1$  范数主成分分析的求解算法问题, 介绍了  $L_1$  范数主成分分析的另一种求解思路。

## 参考文献

- [1] Alain Baccini, Ph Besse, and A Falguerolles. A l1-norm pca and a heuristic approach. *Ordinal and symbolic data analysis*, 1(1):359–368, 1996.
- [2] Jushan Bai and Serena Ng. Determining the number of factors in approximate factor models. *Econometrica*, 70(1):191–221, 2002.
- [3] Michael J Black and Anand Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *International journal of computer vision*, 19(1):57–91, 1996.
- [4] J Paul Brooks, José H Dulá, and Edward L Boone. A pure l1-norm principal component analysis. *Computational statistics & data analysis*, 61:83–98, 2013.
- [5] Gary Chamberlain and Michael Rothschild. Arbitrage, factor structure, and mean-variance analysis on large asset markets, 1982.
- [6] Yeung Lewis Chan, James H Stock, and Mark W Watson. A dynamic factor model framework for forecast combination. *Spanish Economic Review*, 1(2):91–121, 1999.
- [7] Xi Chen, Weidong Liu, Xiaojun Mao, and Zhuoyi Yang. Distributed high-dimensional regression under a quantile loss function. *Journal of Machine Learning Research*, 21(182):1–43, 2020.
- [8] Vartan Choulakian. L1-norm projection pursuit principal component analysis. *Computational statistics & data analysis*, 50(6):1441–1451, 2006.
- [9] Chris Ding, Ding Zhou, Xiaofeng He, and Hongyuan Zha. R 1-pca: rotational invariant l 1-norm principal component analysis for robust subspace factorization. In *Proceedings of the 23rd international conference on Machine learning*, pages 281–288, 2006.
- [10] Sandra Eickmeier and Christina Ziegler. How successful are dynamic factor models at forecasting output and inflation? a meta-analytic approach. *Journal of Forecasting*, 27(3):237–265, 2008.
- [11] Anders Eriksson and Anton Van Den Hengel. Efficient computation of robust low-rank matrix approximations in the presence of missing data using the l 1 norm. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 771–778. IEEE, 2010.
- [12] Mario Forni, Marc Hallin, Marco Lippi, and Lucrezia Reichlin. The generalized dynamic-factor model: Identification and estimation. *Review of Economics and statistics*, 82(4):540–554, 2000.
- [13] Jacqueline S Galpin and Douglas M Hawkins. Methods of l1 estimation of a covariance matrix. *Computational Statistics & Data Analysis*, 5(4):305–319, 1987.
- [14] Junbin Gao. Robust l1 principal component analysis and its bayesian variational inference. *Neural computation*, 20(2):555–572, 2008.

- [15] John Geweke. The dynamic factor analysis of economic time series. *Latent variables in socio-economic models*, 1977.
- [16] Qifa Ke and Takeo Kanade. Robust subspace computation using  $l_1$  norm. 2003.
- [17] Qifa Ke and Takeo Kanade. Robust  $l_1$  norm factorization in the presence of outliers and missing data by alternative convex programming. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 739–746. IEEE, 2005.
- [18] Nojun Kwak. Principal component analysis based on  $l_1$ -norm maximization. *IEEE transactions on pattern analysis and machine intelligence*, 30(9):1672–1680, 2008.
- [19] Matteo Luciani. Large-dimensional dynamic factor models in real-time: A survey. *Available at SSRN 2511872*, 2014.
- [20] Panos P Markopoulos, George N Karystinos, and Dimitris A Pados. Optimal algorithms for  $l_1$ -subspace signal processing. *IEEE Transactions on Signal Processing*, 62(19):5046–5058, 2014.
- [21] Panos P Markopoulos, Sandipan Kundu, Shubham Chamadia, and Dimitris A Pados. Efficient  $l_1$ -norm principal-component analysis via bit flipping. *IEEE Transactions on Signal Processing*, 65(16):4252–4264, 2017.
- [22] Michael McCoy and Joel Tropp. Two proposals for robust pca using semidefinite programming. *arXiv preprint arXiv:1012.1086*, 2010.
- [23] Feiping Nie, Heng Huang, Chris Ding, Dijun Luo, and Hua Wang. Robust principal component analysis with non-greedy  $l_1$ -norm maximization. In *IJCAI proceedings-international joint conference on artificial intelligence*, volume 22, page 1433, 2011.
- [24] Young Woong Park. Optimization for  $l_1$ -norm error fitting via data aggregation. *Inform Journal on Computing*, 33(1):120–142, 2021.
- [25] Young Woong Park and Diego Klabjan. Iteratively reweighted least squares algorithms for  $l_1$ -norm principal component analysis. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 430–438. IEEE, 2016.
- [26] Thomas J Sargent, Christopher A Sims, et al. Business cycle modeling without pretending to have too much a priori economic theory. *New methods in business cycle research*, 1:145–168, 1977.
- [27] James H Stock and Mark W Watson. New indexes of coincident and leading economic indicators. *NBER macroeconomics annual*, 4:351–394, 1989.
- [28] James H Stock and Mark W Watson. Business cycle fluctuations in us macroeconomic time series. *Handbook of macroeconomics*, 1:3–64, 1999.
- [29] James H Stock and Mark W Watson. Forecasting inflation. *Journal of Monetary Economics*, 44(2):293–335, 1999.

- [30] James H Stock and Mark W Watson. Forecasting using principal components from a large number of predictors. *Journal of the American statistical association*, 97(460):1167–1179, 2002.
- [31] James H Stock and Mark W Watson. Macroeconomic forecasting using diffusion indexes. *Journal of Business & Economic Statistics*, 20(2):147–162, 2002.
- [32] Andrea Visentin, Steven Prestwich, and S Armagan Tarim. Robust principal component analysis by reverse iterative linear programming. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 593–605. Springer, 2016.
- [33] 高华川 and 张晓峒. 动态因子模型及其应用研究综述. *统计研究*, 32(12):101–109, 2015.