2020 届研究生硕士学位论文

分 类 号：　　学校代码：10269

密　　级：　　学　　号：51174404015

# 華東師範大學

EAST CHINA NORMAL UNIVERSITY

# 硕士学位论文

MASTER'S DISSERTATION

**论文题目：** 基于回归树的充分降维方法研究

院　　系：统计学院

专　　业：统计学

研 究 方 向：大数据统计

指 导 教 师：於州　教授

研　究　生：吴柏威

完 成 日 期：2020 年 6 月

2020 届研究生硕士学位论文

学校代码：10269
学　　号：51174404015

# 華東師範大學

# 基于回归树的充分降维方法研究

院　　系　统计学院
专　　业　统计学
研究方向　大数据统计
导　　师　於州　教授
研　究　生　吴柏威
完成日期　2020 年 6 月

Master Dissertation of Year 2020 University ID: 10269
Student ID: 51174404015

# Research on Sufficient Dimension Reduction Method Based on Regression Tree

Department School of Statistics

Major Statistics

Research Direction Big data statistics

Supervisor Professor Zhou Yu

Author Baiwei Wu

Date June, 2020

# 华东师范大学学位论文原创性声明

郑重声明：本人呈交的学位论文《基于回归树的充分降维方法研究》，是在华东师范大学攻读硕士/博士（请勾选）学位期间，在导师的指导下进行的研究工作及取得的研究成果．除文中已经注明引用的内容外，本论文不包含其他个人已经发表或撰写过的研究成果．对本文的研究做出重要贡献的个人和集体，均已在文中作了明确说明并表示谢意．

作者签名：吴柏威　　日　　期：2020.6.8

# 华东师范大学学位论文著作权使用声明

《基于回归树的充分降维方法研究》系本人在华东师范大学攻读学位期间在导师指导下完成的硕士/博士（请勾选）学位论文，本论文的研究成果归华东师范大学所有．本人同意华东师范大学根据相关规定保留和使用此学位论文，并向主管部门和相关机构如国家图书馆、中信所和“知网”送交学位论文的印刷版和电子版；允许学位论文进入华东师范大学图书馆及数据库被查阅、借阅；同意学校将学位论文加入全国博士、硕士学位论文共建单位数据库进行检索，将学位论文的标题和摘要汇编出版，采用影印、缩印或者其它方式合理复制学位论文．

本学位论文属于（请勾选）

（　　）1．经华东师范大学相关部门审查核定的“内部”或“涉密”学位论文 *，于____ 年____ 月____ 日解密，解密后适用上述授权．

（√）2．不保密，适用上述授权．

作者签名：吴柏威　　导师签名：

日　　期：2020.6.8　　日　　期：2020.6.8

* “涉密”学位论文应是已经华东师范大学学位评定委员会办公室或保密委员会审定过的学位论文（需附获批的《华东师范大学研究生申请学位论文“涉密”审批表》方为有效），未经上述部门审定的学位论文均为公开学位论文．此声明栏不填写的，默认为公开学位论文，均适用上述授权）．

# 吴柏威硕士学位论文答辩委员会成员名单

| 姓 名 | 职 称 | 单 位 | 备 注 |
| --- | --- | --- | --- |
| 张日权 | 教授 | 华东师范大学 | 主 席 |
| 陆智萍 | 副教授 | 华东师范大学 | |
| 方方 | 教授 | 华东师范大学 | |
| | | | |
| | | | |
| | | | |
| | | | |

# 目 录

摘　　要 v

ABSTRACT（英文摘要） vi

主要符号对照表 viii

第一章　引言 1

§1.1 研究背景 . . . . . 1

§1.2 国内外研究现状 . . . . . 1

§1.3 本文的框架与创新点 . . . . . 3

第二章　基础知识 4

§2.1 降维 . . . . . 4

§2.1.1 降维的定义与分类 . . . . . 4

§2.1.2 降维效果的评价标准 . . . . . 4

§2.1.3 充分降维 . . . . . 5

§2.1.4 估计中心降维子空间的方法 . . . . . 6

§2.1.5 结构维数 . . . . . 8

§2.1.6 多维响应变量的充分降维方法 . . . . . 8

§2.2 回归树与集成学习 . . . . . 10

§2.2.1 回归树 . . . . . 10

§2.2.2 集成学习 . . . . . 10

第三章　基于回归树的充分降维方法 15

§3.1 核心思想 . . . . . 15

§3.2 基于回归树的 SIR 方法 . . . . . 16
§3.3 基于回归树的 SAVE 方法 . . . . . 17
§3.4 基于回归树的 DR 方法 . . . . . 17
§3.5 响应变量缺失值处理方法 . . . . . 18

第四章 数值模拟与实例分析 19
§4.1 响应变量二维的情形 . . . . . 19
§4.1.1 线性模型 . . . . . 19
§4.1.2 非线性模型 . . . . . 23
§4.2 响应变量三维及以上的情形 . . . . . 25
§4.2.1 响应变量三维 . . . . . 25
§4.2.2 响应变量高维 . . . . . 29
§4.3 响应变量有缺失值 . . . . . 33
§4.4 与现有方法的比较 . . . . . 35
§4.5 实例分析 . . . . . 37

第五章 结论及展望 39
§5.1 结论 . . . . . 39
§5.2 展望 . . . . . 39

参考文献 41

致谢 44

# 插 图 目 录

图 3-1 回归树示意图 ..... 15

# 表 格 目 录

表 4-1 线性模型 $\sigma = 0.02$ 的实验结果 . . . . . 20
表 4-2 线性模型 $\sigma = 0.2, 0.5, 1, 2$ 的实验结果 . . . . . 21
表 4-3 线性模型控制树最大深度 $max\_depth = 4, 5$ 的实验结果 . . . . . 21
表 4-4 线性模型 $p = 15, 20$ 的实验结果 . . . . . 22
表 4-5 线性模型使用不同树模型的实验结果 . . . . . 22
表 4-6 不同非线性模型的实验结果—Xgboost . . . . . 24
表 4-7 不同非线性模型的实验结果—RF . . . . . 24
表 4-8 model(4-6) 在不同树模型下的实验结果 . . . . . 26
表 4-9 model(4-7) 在不同树模型下的实验结果 . . . . . 27
表 4-10 model(4-8) 在不同树模型下的实验结果 . . . . . 29
表 4-11 model(4-9) 在不同树模型下的实验结果 . . . . . 30
表 4-12 model(4-9) 在不同树模型下的实验结果 ($p$=15) . . . . . 31
表 4-13 model(4-10) 在不同树模型下的实验结果 ($p$=20) . . . . . 32
表 4-14 model(4-11) 响应变量有缺失值的实验结果-1 . . . . . 34
表 4-15 model(4-11) 响应变量有缺失值的实验结果-2 . . . . . 35
表 4-16 model(4-6) 树模型与投影法的实验结果比较 . . . . . 36
表 4-17 model(4-9) 树模型与投影法的实验结果比较 . . . . . 36
表 4-18 model(4-10) 树模型与投影法的实验结果比较 . . . . . 36
表 4-19 树模型与投影法的变异性比较 . . . . . 37
表 4-20 RF-DR 给出的前两个基方向 . . . . . 38

# 摘　　要

大数据时代的到来使人们面对的数据越来越复杂，充分降维理论对于研究这种复杂数据有着重要的意义。在响应变量多维时，传统的充分降维理论往往会面临许多难题。本文主要研究基于回归树的充分降维方法，在响应变量多维的情况下巧妙地解决了维数灾难的问题。

响应变量一维时，传统的方法通常会采用切片的方法对响应变量进行划分。但是随着维数的升高，这种切片的方法会导致切分出的许多切片内部没有任何样本点。而回归树的方法可以对多维空间进行划分，叶子结点的值正好是空间划分后的均值。基于这一思想，本文提出了基于回归树的充分降维方法。这里的回归树模型可以是梯度提升树，随机森林，Xgboost 等。对于 SIR、SAVE、DR 方法，本文给出了估计核矩阵的方法。

最后，本文通过大量的模拟与实例分析，验证了该方法在响应变量多维时的有效性。与现有的方法相比，该方法在响应变量高维时，有一定的优势。无论是线性还是非线性模型，在有一定程度噪声的情况下，本文的方法都可以较好的估计出降维方向。在样本量比较少时，随机森林的效果较好。在样本量比较大时，梯度提升树、随机森林、Xgboost 的表现相当。由于集成学习模型往往会有较多的超参数，对于超参数的设置仍是缺乏理论依据的，但一般使用默认的参数便能取得不错的效果。本文还利用集成学习模型对于缺失值的处理办法，将其用到了充分降维领域，分析了响应变量有缺失值时的充分降维效果，利用含缺失值的样本信息后，降维效果显著好于丢弃这些含缺失值的样本。

**关键词：**回归树，集成学习，充分降维，多维响应变量

# Abstract

With the advent of the era of big data, the data is becoming more and more complex. The theory of sufficient dimension reduction is of great significance for studying such complex data. In the case of multivariate responses, there exists many problems. Therefore, this paper mainly focuses on the method of sufficient dimension reduction based on regression tree, which solves the problem of dimensional disaster in the case of multivariate responses.

In the scenario of univariate response, traditional method usually use the slice method to divide the response variable. However, with the increase of the dimension, this method can easily lead to the lack of sample points in many slices. The regression tree method can divide the multi-dimensional space and the value of the leaf nodes is just the mean value after the space division. Based on this idea, this paper presents a new sufficient dimension reduction method based on regression tree, which can be GBDT, RF, Xgboost, etc. For SIR, SAVE and DR, this paper gives the method for estimating the kernel matrix.

Finally, a lot of simulations and one example is used to verify the effectiveness of the method in multivariate responses. Compared with the existing methods, the method in this paper performs better in the case of high-dimensional variables. Regardless of whether it is a linear or non-linear model, the method can better estimate the dimension reduction direction in

the presence of a certain degree of noise. When the sample size is small, RF works better. When the sample size is relatively large, the performance of GBDT, RF, and Xgboost is equivalent. Because ensemble learning models often have many hyperparameters, there is still no theoretical basis for the setting of hyperparameters, but generally using the default parameters can achieve good results. The ensemble learning model can handle missing values efficiently, so this paper applies it to sufficient dimension reduction with missing values and analyzes the effect of sufficient dimension reduction on missing values in response variables. After using sample information with missing values, the dimension reduction effect performs significantly better than just discarding these samples with missing values.

**Key Words**: regression tree, ensemble learning, sufficient dimension reduction, multivariate responses

# 主要符号对照表

| | |
|---|---|
| SIR | 切片逆回归 (Sliced Inverse Regression) |
| SAVE | 切片平均方差估计 (Sliced Average Variance Estimation) |
| DR | 方向回归 (Directional Regression) |
| GBDT | 梯度提升树 (Gradient Boosting Decision Tree) |
| RF | 随机森林 (Random Forest) |
| PR | 投影重采样法 (Projective Resampling) |
| $X_i^n$ | 变量 $X$ 的第 $n$ 个样本的第 $i$ 维 |

# 第一章　引言

## §1.1　研究背景

早在 2012 年，Viktor Mayer-Schönberger 就出版了非常具有前瞻性的著作《大数据时代》。他很早就洞见了大数据时代的发展趋势。在书中，他指出大数据时代将变革我们的工作、生活和思维方式。2019 年，央视纪录片《大数据时代》播出。它阐述了随着计算机和互联网的广泛应用，人类的数据爆炸式地生长，大数据技术对我们转型、决策、商业带来的重要影响。

随着我们观测手段的丰富，信息技术的发展，我们存储了海量高维数据。高维数据是大数据时代的一大体现，目前在各个领域随处可见。例如在各个企业中，面向用户的厂商会根据用户信息、用户行为构建每个用户的画像。对于每一个用户，它的数据维数往往是成百上千维，甚至更高。在视频、图像数据挖掘领域，一张 4k 的图片分辨率就能达到 3840×2160。在金融、生物等其它领域，如股票、期货分时图，DNA、RNA 序列也会产生非常多的高维数据。

高维数据无疑是十分复杂的，它给我们带来了很多挑战，例如“维数灾难”。举个简单的例子，在 $k$ 近邻算法中，我们在预测样本 $x$ 类别时，需要在其邻域内找到若干个训练样本。当 $x$ 的维数比较大时，往往在其附近找不到足够的样本，高维情况下的数据稀疏问题会使很多算法失效。降维是解决维数灾难的一个有效途径。在充分降维中，我们往往会采用切片方法估计中心降维子空间。但在响应变量高维时，简单的切片方法不再可行。现有的其他方法也有一定局限性。本文主要研究响应变量多维的情形下充分降维的相关问题，这对高维数据处理有重要意义。

## §1.2　国内外研究现状

在充分降维领域，Li (1991)[16] 提出了 SIR(Sliced Inverse Regression) 方法，SIR 是最经典的一种充分降维方法，但该方法有一定局限性，有时候不能找全所有的降维方向。Cook and Wersbeig(1991)[7] 提出了 SAVE(Sliced Average Variance

Estimates)，相比与 SIR，该方法能够更加全面的找出降维方向，也就是找到更加准确的降维子空间。Hsing and Carroll(1992) [11] 对切片方法进行了研究，他们证明了在切片在 $\sqrt{n}$ 到 $n$ 区间时，切片方法是渐近收敛的。Cook(1998) 提出了 PHD 方法，用于估计中心降维子空间 [4]。

在 Li 开创性地提出 SIR 方法之后，很多学者对切片方法有了进一步的研究。Hsing(1999) 首次将 $k$ 近邻方法应用到了切片逆回归方法中，在切片过程中应用了最近邻方法进行切分,而不是简单地按区间长度进行平均地切分 [10]。Bura and Cook(2001) 提出了 PIR(Parametric Inverse Regression) [1]，将参数方法运用到了切片逆回归上。SIR 方法的假设是一阶的，SAVE 方法的假设是二阶的。Yin and Cook(2003) 推广到了三阶矩，四阶矩，提出了 SAT(Sliced Average Third-Moment Estimation) 方法 [36]。Li et al.(2005) 提出了穷尽中心降维子空间概念和 CR(Contour regression) 方法 [15]。Li and Wang(2007) 在 SIR 和 SAVE 的基础上，将这两种方法的假设条件进行了组合，提出了 DR(Directional Regression) 方法 [13]。该方法的效果通常会比单一的 SIR 或 SAVE 方法更稳健。Li et al.(1999) 分析了响应变量 $Y$ 中有缺失值情形下的充分降维方法 [18]。Li et al.(2003) 首次提出了多重切片法 [17] 思想，用于解决响应变量 $Y$ 是多维的问题。但由于维数增加，这种方法的效率不高，在更高维的情况下，会变得不可用。Setodji and Cook(2004) 将 K-means 方法用到了切片逆回归中，适用于响应变量存在异常值的情形 [32] 。在响应变量低维时，该方法有时也适用。Ni et al.(2005) 将岭回归用到了切片逆回归中 [27]，在一些场景下可以更好的估计回归曲线。Li et al.(2005) 将充分降维方法用到了变量选择上 [19]。Li and Nachtsheim(2006) 研究了稀疏情况下的 SIR 方法，提出了 Sparse SIR [21]。Liu et al.(2019) 研究了基于 lasso 的 Sparse SIR [24]。Park et al.(2009) 将充分降维方法应用到了时间序列分析上 [29, 30]。Cook et al.(2007) 研究了数据集在 n<p 时的充分降维方法 [20]。Lue(2008) 研究了含缺失数据的 SAVE 方法 [25]。Li and Yin(2008) 提出了正则化的切片逆回归 [22]。Cook and Lee(1999) 研究了响应变量是 0-1 变量的充分降维方法 [6]。Chiaromonte et al.(2002) 推广到了响应变量为一般的类别变量 [3]。Li et al.(2008) 提出了投影重采样法 [14]。其核心思想是在响应变量多维时，通过投影的办法先将响应变量转化成一维。Ma et al.(2012) 研究了半参数下的充分降维方法 [26]。Cook et al.(2012) 研究了一类高维下的充分降维方法的渐近性质 [5]。Yu et al.(2016) 研究了基于边际切片逆回归的超高维变量选择方法 [37]。Tan et al.(2020) 给出了一种速率最优的稀疏 SIR 自适应估计方案 [34]。

在机器学习领域，决策树是目前最热门的方法之一。该方法的思想来源于 Quinlan，他分别在 1986 年和 1993 年提出了 ID3 算法和 C4.5 算法 [31]。Beriman(1994) 等人提出了 CART 算法，这是目前广泛应用的决策树方法。Friedman(2001) 提出了 GBDT(Gradient Boosting Decision Tree) [9]，该方法通过每次计算模型的负梯度来进

行模型迭代，得到了更好的效果。Liaw et al.(2002) 使用 Bagging 的方法，提出了 RF(Random Forest) 方法[23]，这是 Bagging 方法在决策树使用的经典之作，其随机采样与随机特征的特性使模型表现出强大的性能。Chen and Guestrin(2016) 提出了 Xgboost[2]，通过对目标函数进行二阶展开，加入正则化等，提高了树模型的精度和泛化能力。Ke et al.(2017) 提出了更高效的方法 Lightgbm[12]，通过对特征进行直方图排序，以及提高叶子结点分裂深度的优先级等，使得其工程化性能更好。

## §1.3 本文的框架与创新点

本文的研究内容是基于回归树的充分降维方法。我们提出了一种适用于响应变量多维情况下的充分降维方法，并通过数值模拟的方法验证了我们提出的方法的有效性。我们还分析了自变量维数变化，响应变量维数变化，模型噪声扰动，响应变量有缺失值时，我们方法的适用性。本文的框架如下：

第一章主要介绍了研究背景、国内外关于决策树、集成学习和充分降维的研究现状以及本文要研究的主要内容。

第二章主要介绍了充分降维的基础知识，包括什么是降维，降维效果评估方法，什么是充分降维，SIR、SAVR、DR 方法，什么是结构维数，现有的适用于多维响应变量的充分降维方法。第二章还介绍了回归树与集成学习的相关知识，包括梯度提升树，随机森林，Xgboost，为第三章提出适用于响应变量多维情况下的充分降维方法做准备。

第三章我们基于回归树的特点，提出了我们的方法，该方法适用于响应变量多维的情形。我们给出了 SIR、SAVR、DR 方法在多维响应变量时的估计方法，该方法有效地对响应变量空间进行了切分，有较好的普适性。根据部分集成学习模型的特性，我们还提出了多维响应变量部分维度数据含缺失值时的处理办法。

第四章主要是数值模拟。根据我们提出的方法，我们模拟了二维、三维以及更高维响应变量下，不同方法的降维效果。我们还模拟并分析了不同树模型，树模型参数改变、自变量、响应变量维数变化，模型噪声扰动，响应变量有缺失值时的降维效果，验证了我们方法的有效性。我们通过与现有的投影重采样法的对比，说明了我们提出的基于回归树的充分降维方法的优势。最后，我们还找了一个实际数据进行分析。

第五章给出了论文的结论，我们方法的优势以及对多维响应变量情况下充分降维方法的展望。

本文的创新之处在于将回归树，集成学习的思想运用到充分降维领域，从全新的角度，提出了一种适用于多维响应变量的充分降维方法。该方法尤其在响应变量高维时，有非常不错的表现。

# 第二章 基础知识

## §2.1 降维

### §2.1.1 降维的定义与分类

降维的数学描述如下：设 $\{x^n\}_{n=1}^N \subset R^p$ 是容量为 $N$ 的数据集。我们去找到一个降维映射：

$$F : \Omega^p \to \Omega^d$$

$$x \to y = F(x),$$

这里 $d \ll p$，称 $y$ 是 $x$ 的低维表示。

通过将原始高维属性空间上的数据映射到一个低维空间，一些在高维情况下不可行的算法变得有效可行。降维的方法有很多，从不同角度有不同的分类。基于线性变换进行降维的方法称为线性降维方法，基于非线性变换进行降维的方法称为非线性降维方法。根据是否考虑响应变量的信息，可以分为无监督降维、半监督降维和有监督降维。常见的降维方法有：线性判别分析，主成分分析法，核化线性降维，流形学习，度量学习。

### §2.1.2 降维效果的评价标准

降维的目的是尽可能地保留原始数据信息，降维可能会对原始数据信息造成损失。那么我们怎么评价降维的效果呢？

在机器学习领域，通常可以根据学习器在降维前后的性能、表现来判断降维效果的好坏，包括空间复杂度，时间复杂度，精确率，召回率等。

如果降维后的结果是二维或者三维的情形，通常可以通过可视化的手段来判断降维效果的好坏。在高维情况下，可以通过计算样本距离来判断。

Li(1991)[16] 通过计算降维方向与降维子空间的平方多重相关系数来判断估计的降维方向与降维子空间的接近程度。平方多重相关系数越大，说明估计的降维方向与真

实的降维子空间越接近。

我们也可以通过计算真实降维空间的投影矩阵和估计的降维空间的投影矩阵的距离来评估降维的效果:

$$\left\| B\left(B^TB\right)^{-1}B^T - \hat{B}\left(\hat{B}^T\hat{B}\right)^{-1}\hat{B}^T \right\|_{\mathrm{r}},$$

显然距离越小，估计的降维空间的投影矩阵越接近真实的降维空间的投影矩阵。但这种方法比较适合估计降维后维度相同的情形，不同维度下，计算出来的距离不能直接进行比较。

我们还可以通过直接计算估计的降维空间的投影矩阵与真实的降维空间的投影矩阵的迹相关系数[8]来判断。对于秩为 $d$ 的投影矩阵 $A$ 和 $B$：

$$R^2 = \frac{trace(A^TB)}{d}. \tag{2-1}$$

用（2-1）式计算出来的 $R^2 \in [0,1]$，这样可以比较直观的判断降维效果的好坏。这也是本文在仿真时用于判断降维效果的主要依据。

### §2.1.3 充分降维

对于一个回归问题，自变量是 $p$ 维的 $X$，响应变量是 $q$ 维的 $Y$，$Y = f(X) + \epsilon$。充分降维的核心思想是找到一个 $p \times d$ 的矩阵 $B$，其中 $p \leq d$，使得条件分布 $Y|X$ 和 $Y|B^TX$ 相同。这种降维方法是不损失条件分布的信息的：

$$Y \perp\!\!\!\perp X|B^TX.$$

显然，这样的矩阵 $B$ 一定是存在的，只需令 $p = d, B = I_p$。其次，这样的矩阵 $B$ 不是唯一的，当矩阵 $B$ 满足条件，矩阵 $BA$ 也满足条件，其中矩阵 $A$ 是 $d \times d$ 的满秩矩阵：

$$Y \perp\!\!\!\perp X|(BA)^TX.$$

我们定义 $\mathrm{span}(B)$ 为矩阵 $B$ 列向量展开的空间，$\mathrm{span}(B)$ 是一个降维空间。同样的，$\mathrm{span}(BA)$ 也是一个降维空间。Cook 提出了中心降维子空间的概念，当所有降维空间的交集也满足这是一个降维空间时，就称这个空间是中心降维子空间。这里我们记作 $\mathscr{S}_{Y|X}$，将 $\mathscr{S}_{Y|X}$ 的维数称作结构维数。

中心降维子空间有如下性质：

$$\mathscr{S}_{Y|AX+b} = A^{-T}\mathscr{S}_{Y|X}. \tag{2-2}$$

基于 (2-2) 式，我们可以知道自变量 $X$ 对应的降维空间，和自变量为标准化后的 $X$ 对应的降维空间是存在转换关系的。所以我们在研究相关问题时，可以首先将自变量 $X$ 做标准化处理。

### §2.1.4 估计中心降维子空间的方法

历史上，对于中心降维子空间的估计有一些方法，例如 SIR、SAVE、CR、DR 方法。我们介绍比较经典的几种方法。

#### 切片逆回归 (SIR)

Li (1991) 提出了 SIR(Sliced Inverse Regression) 方法 [16]，它是目前被广泛应用，也是最经典的一种充分降维方法。这个方法的基本假设是线性条件均值假设。

若 $\beta^{\top}\Sigma\beta$ 正定，有：

$$E\left[X - E(X)|\beta^{\top}X\right] = P_{\beta}^{\top}(\Sigma)[X - E(X)].$$

若随机变量 $X$ 是平方可积的，$\Sigma = var(X)$ 是非奇异的。有如下结论：

$$\Sigma^{-1}[E(X|Y) - E(X)] \in \mathscr{S}_{Y|X}. \tag{2-3}$$

由 (2-3) 式易知：

$$\text{span}\left(\Sigma^{-1}\,\text{cov}[E(X|Y)]\Sigma^{-1}\right) \subseteq \mathscr{S}_{Y|X}.$$

我们将 $X$ 做标准化处理，令 $Z = \Sigma^{-\frac{1}{2}}(X - E(X))$。根据中心降维子空间的性质，我们容易知道：

$$\text{span}\left(\text{cov}[E(Z|Y)]\right) \subseteq \mathscr{S}_{Y|Z}.$$

$\text{span}\left(\text{cov}[E(Z|Y)]\right)$ 非零特征值对应的特征向量构成 $\mathscr{S}_{Y|Z}$ 的一组基。对于中心降维子空间的估计，一种比较简单的方法如下：

对于数据集 $[(X^1,Y^1),(X^2,Y^2),\cdots,(X^n,Y^n)]$，这里上标 $n$ 表示第 $n$ 个样本。

Step-1 对 $X$ 作标准化，$Z=\hat{\Sigma}^{-\frac{1}{2}}(X-\hat{\mu})$ 。

Step-2 根据 $Y$ 所在区间，将该区间切成 $H$ 片，记为 $I_1,I_2,\cdots,I_H$，$p_h$ 记作 $Y$ 落入对应 $I_h$ 的概率，计算每个切片中 $Z$ 的均值得到 $\hat{m}_h,h=1,2,\cdots,H$。

Step-3 构造核矩阵的估计 $\hat{M}=\Sigma_{h=1}^{H}\hat{p}_h\hat{m}_h\hat{m}_h'$

Step-4 对矩阵 $\hat{M}$ 做特征分解，取前 $k$ 个最大的特征向量 $\nu_1,\nu_2,\cdots,\nu_k$，左乘 $\hat{\Sigma}^{-\frac{1}{2}}$ 得到，$\hat{\Sigma}^{-\frac{1}{2}}\nu_1,\hat{\Sigma}^{-\frac{1}{2}}\nu_2,\cdots,\hat{\Sigma}^{-\frac{1}{2}}\nu_k$。

再通过一步计算就能得到投影矩阵。

SIR 方法有一定局限性。设 $Y=f(X_1)+\varepsilon$。当函数 $f$ 关于 0 对称时，总有：

$$E(X_i|f(X_1)+\varepsilon)=E(-X_i|f(X_1)+\varepsilon). \tag{2-4}$$

(2-4) 式意味着，对于每个 $X_i$，有 $E(X_i|f(X_1)+\varepsilon)=0$，可以得到 $E(X|Y)=0$。在这种情况下无法找全所有的降维方向。

### 切片平均方差估计 (SAVE)

Cook and Wersbeig(1991) 提出了 SAVE(Sliced Average Variance Estimates) [7]，相比于 SIR，该方法能够更加全面的找出降维方向。SAVE 方法在 SIR 方法的基础上增加了一个假设：常数条件方差假设。

$$\mathrm{var}\left(X|\beta^{\top}X\right)=\Sigma Q_{\beta}(\Sigma),$$

其中 $Q_{\beta}(\Sigma)=I-P_{\beta}(\Sigma)$。

进而得到：

$$\mathrm{span}(\Sigma-\mathrm{var}(X|Y))\subseteq\Sigma\mathscr{S}_{Y|X}.$$

令 $Z=\hat{\Sigma}^{-\frac{1}{2}}(X-\hat{\mu})$，做标准化后有：

$$\mathrm{span}\left(I_p-\mathrm{var}(Z|Y)\right)\subseteq\mathscr{S}_{Y|Z}.$$

由此可以推出：

$$\mathrm{span}(E\left[I_p-\mathrm{Cov}(Z|Y)\right]^2)\subseteq\mathscr{S}_{Y|Z}.$$

$\mathrm{span}(E\left[I_p-\mathrm{Cov}(Z|Y)\right]^2)$ 非零特征值对应的特征向量构成 $\mathscr{S}_{Y|Z}$ 的一组基。

#### 方向回归 (DR)

Li and Wang(2007) 在 SIR 和 SAVE 的基础上,提出了 DR(Directional Regression) 方法 [13]。在线性条件均值和常数条件方差的假设下，证明了：

$$\text{span}\left\{2I_p - E\left[(Z-\tilde{Z})(Z-\tilde{Z})^\top|Y,\tilde{Y}\right]\right\} \subseteq \mathscr{S}_{Y|Z}.$$

可以推出：

$$\text{span}(E[2I_p - A(Y,\tilde{Y})]^2) \subseteq \mathscr{S}_{Y|Z}.$$

DR 方法的核矩阵可以写成这样的形式：

$$\begin{aligned} M = 2E\left[E^2\left(ZZ^T|Y\right)\right] + 2E^2\left[E(Z|Y)E\left(Z^T|Y\right)\right] \\ +2E\left[E\left(Z^T|Y\right)E(Z|Y)\right]E\left[E(Z|Y)E\left(Z^T|Y\right)\right] - 2I_p. \end{aligned}$$

DR 方法可以认为是 SIR 和 SAVE 的一种组合，切片方法同样适用于该方法。

### §2.1.5 结构维数

对中心降维子空间的估计。一是降维的基方向，二是基方向的个数，即结构维数。结构维数估计方法有序贯检验法，BIC 准则法，Bootstrap 法。这里我们简单介绍一下序贯检验法。

#### 序贯检验法

序贯检验法 [33] 是充分降维领域最常用的确定结构维数的方法。设核矩阵的样本估计 $\hat{M}$ 的特征值为 $\hat{\lambda}_1, \hat{\lambda}_2, \cdots, \hat{\lambda}_n$。该检验的零假设和备择假设为：$H_0 : d = m$ v.s $H_1 : d > m$ 。统计量为：

$$T_m = n\sum_{m+1}^{p} w\hat{\lambda}_j^k,$$

其中 $w$ 是标准化因子。当 $X$ 服从正态分布时，$T_m$ 通常会收敛到标准卡方分布。由于本文的重点是估计降维的基方向，故在仿真时假定结构维数已知。

### §2.1.6 多维响应变量的充分降维方法

多维响应变量的充分降维方法一直是一大难题。Li et al.(2003) 提出了多重切片法 [17] 的方法，但随着维数增加，这种方法的效率不高，效果不理想。Setodji and

Cook(2004) [32] 将 K-means 方法用到了切片逆回归中，他们的主要思想是对响应变量 $Y$ 做聚类。但维数比较高时，聚类难以实施，这种方法便失效了。也有一种思想是通过一定的手段，将多维响应变量转化维一维的响应变量，例如 Li et al.(2008) 提出的投影重采样法 (Projective Resampling) [14]。该方法在响应变量维数较低时降维效果不错，但在响应变量高维时效果不理想。

### 投影重采样法

投影重采样法的核心思想是通过投影的方法将多维的响应变量转化为一维响应变量。投影一次显然不能够获得足够的信息去估计中心降维子空间，因此需要多次随机的进行投影。投影重采样法的步骤如下：

Step-1 设原数据的样本量为 $n$，我们需要生成个 $m_n$ 个独立同分布的随机变量 $T_1,\cdots,T_{m_n}$。这些随机变量是单位球形 $\mathbb{S}^p=\{\mathbf{t}\in\mathbb{R}^p:\|\mathbf{t}\|=1\}$ 上的均匀分布。其中 $m_n$ 的大小需要满足 $m_n/n\to\infty$，当 $n\to\infty$，例如 $n$ 可以取 $n\log(n)$ 或者 $n^{3/2}$。球面上的均匀分布可以通过正态分布生成，$T_j=G_j/\|G_j\|$，其中 $G_1,\cdots,G_{m_n}$ 是独立同分布的 $q$ 维标准正态分布。

Step-2 对 $X$ 作标准化，$Z=\hat{\Sigma}^{-\frac{1}{2}}(X-\hat{\mu})$ 。对于每个 $T_j,j=1,\cdots,m_n$，乘上多维的响应变量 $Y$，得到 $T_j^TY,j=1,\cdots,m_n$。这样就可以得到 $m_n$ 组数据集 $(Z,T_j^TY),j=1,\cdots,m_n$。

Step-3 对每一个数据集 $(Z,T_j^TY),j=1,\cdots,m_n$。利用一维响应变量的充分降维方法，例如 SIR，可以得到核矩阵的估计 $\hat{M}_j,j=1,\cdots,m_n$。

Step-4 对 $\hat{M}_j,j=1,\cdots,m_n$ 求均值得到最终的矩阵 $\hat{M}$。

Step-5 对矩阵 $\hat{M}$ 做特征分解，取前 $k$ 个最大的特征向量 $\nu_1,\nu_2,\cdots,\nu_k$，左乘 $\hat{\Sigma}^{-\frac{1}{2}}$ 得到，$\hat{\Sigma}^{-\frac{1}{2}}\nu_1,\hat{\Sigma}^{-\frac{1}{2}}\nu_2,\cdots,\hat{\Sigma}^{-\frac{1}{2}}\nu_k$。

这里对原始的 SIR，SAVE，DR 等充分降维方法均适用。

本文在仿真时会将基于回归树的充分降维方法与基于投影重采样的充分降维方法做比较。

# §2.2 回归树与集成学习

## §2.2.1 回归树

决策树 [39] 是目前常用的一种分类与回归方法。它是一种树形结构，我们可以将一颗决策树看成一个 if-then 集合，每一个实例可以被有且只有一条决策树上的一条路径覆盖，互斥且完备是决策树路径的重要性质。CART 模型递归地分割每个特征，将输入空间划分成有限个单元，在每个单元上输出条件概率分布。

在构建决策树模型时，我们可以自定义准则对空间进行划分，例如最常见的方法，就是用平方误差最小的准则构建决策树。在寻找最优切分变量 $j$ 和最优切分点 $s$ 时，求解方法如下：

$$\min_{j,s}\left[\min_{c_1}\sum_{x_i\in R_1(j,s)}(y_i-c_1)^2+\min_{c_2}\sum_{x_i\in R_2(j,s)}(y_i-c_2)^2\right].$$

通过对空间不断进行划分，最终可以得到一颗最小二乘回归树。

## §2.2.2 集成学习

单一的决策树对于复杂数据的拟合效果往往不尽如人意。如今，集成学习在数据挖掘领域大放异彩，它通过结合多个学习器，通常可以获得远优于单个学习器的泛化性能。假设基学习器之间的错误率互相独立，则由 $Hoeffding$ 不等式可知，集成学习 [40] 的错误率为：

$$P(H(x)\neq f(x))=\sum_{k=0}^{|T/2|}\binom{T}{k}(1-\epsilon)^k\epsilon^{T-k}\leq exp(-\frac{1}{2}T(1-2\epsilon)^2). \tag{2-5}$$

由 (2-5) 式，我们可以知道随着基学习器数目的增大，集成学习的错误率将指数级下降，最终趋于 0。但实际情况中，要做到基学习器的绝对独立也是有挑战的。

集合学习常见的两类，一类是 Bagging，各个弱学习器之间独立的学习，各个学习器可以并行生成。还有一类是 Boosting，之后生成的学习器去学习残差，这种方法的学习器是串形生成的。常用的决策树集成学习方法有，梯度提升树 (Gradient Boosting Decision Tree)，随机森林 (Random Forest)，Xgboost，Lightgbm 等。

### 梯度提升树

Friedman (2001) 提出了 GBDT(Gradient Boosting Decision Tree) [9]。梯度提升树可以认为以 CART 为基学习器的一种 Boosting 方法。其模型可以简单表示为：

$$F(x)=\sum_{m=1}^{M}\gamma_m h_m(x).$$

通过对样本真实值与当前学习器残差的不断逼近，得到下一个学习器：

$$F_m(x)=F_{m-1}(x)+\gamma_m h_m(x).$$

求解时的目标函数为：

$$h_m=\arg\min_h\sum_{i=1}^{n}L\left(y_i,F_{m-1}\left(x_i\right)+\gamma_m h\left(x_i\right)\right),$$

其中 $L\left(y_i,F_{m-1}\left(x_i\right)+\gamma_m h\left(x_i\right)\right)$ 是损失函数。

当损失函数为指数损失函数时，GBDT 和 Adaboost 算法等价。当然，我们可以选择不同的损失函数。常见的损失函数有：均方差损失，绝对损失，Huber 损失，分位数损失等。Huber 损失与分位数损失健壮性较强，对于异常值多的数据，表现相对稳定且良好。

为了解决过拟合问题，可以选择较小的步长，这样拟合出来的模型通常具有较好的泛化能力。该模型在求解时一般采用梯度下降法。GBDT 可以处理不同类型的数据，包括离散值和连续值。其次，该方法对参数的调节难度较低，通过简单的参数调节，就可以获得比较高的准确率。由于其算法开销低，性能表现优秀，在数据挖掘竞赛，各类公司中有着广泛的应用。其缺点是不同树与树之间是串行计算的，相比于 Bagging 方法，在计算时间上有劣势。

### 随机森林

随机森林 [23] 是一种以决策树为基学习器，采用 Bagging 方法组合的树模型。传统的决策树构造过程中，是在所有属性中选择一个最优的属性进行分割。而随机森林的每棵树的特征通常只选择部分属性。假设数据原有 $d$ 个属性，我们可以选取其中的 $k$ 个属性来构造决策树。Breiman 提出 $k$ 的推荐值为 $\log_2 d$。随机森林的随机性还有一点，在构造每一棵决策树时，只随机地选取其中部分样本。正是因为样本的扰动和属性的扰动，这样构造出来的模型具有了良好的泛化能力。

由于随机森林采用了 Bagging 方法，不同树可以并行生成，大大降低了构造决策树需要的时间。

### Xgboost

Chen and Guestrin(2016) 提出了 Xgboost [2]，它的基本思想和 GBDT 类似，在 GBDT 基础上做了泰勒展开，并且加入了一些正则项。传统的 GBDT 以 CART 树作为基学习器，Xgboost 还支持了线性分类器。

Xgboost 的目标函数有两部分组成，一部分是损失函数，一部分是正则项。

$$Obj = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i\right) + \sum_{k=1}^{K} \Omega\left(f_k\right), \tag{2-6}$$

第一部分是损失函数，第二部分用来刻画模型复杂度。

Boosting 方法都是加法模型，设 $\hat{y}_i^{(t)}$ 是前 $t$ 棵树样本 $i$ 的预测结果，$f_t\left(x_i\right)$ 是第 $t$ 棵树的预测函数，有如下表达式：

$$\hat{y}_i^{(t)} = \sum_{k=1}^{t} f_k\left(x_i\right) = \hat{y}_i^{(t-1)} + f_t\left(x_i\right). \tag{2-7}$$

将 (2-7) 式代入 (2-6) 式可以得到：

$$\begin{aligned} Obj^{(t)} &= \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t)}\right) + \sum_{i=1}^{t} \Omega\left(f_i\right) \\ &= \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)} + f_t\left(x_i\right)\right) + \Omega\left(f_t\right) + \text{ constant }. \end{aligned} \tag{2-8}$$

定义损失函数关于 $\hat{y}^{(t-1)}$ 的一阶偏导数 $g_i = \partial_{\hat{y}^{(t-1)}} l\left(y_i, \hat{y}^{(t-1)}\right)$，二阶偏导数 $h_i = \partial^2_{\hat{y}^{(t-1)}} l\left(y_i, \hat{y}^{(t-1)}\right)$。将损失函数进行二阶泰勒展开，代入 (2-8) 式中，可以得到下式：

$$Obj^{(t)} \simeq \sum_{i=1}^{n} \left[ l\left(y_i, \hat{y}_i^{(t-1)}\right) + g_i f_t\left(x_i\right) + \frac{1}{2} h_i f_t^2\left(x_i\right) \right] + \Omega\left(f_t\right) + \text{ constant}. \tag{2-9}$$

去掉常数项后，整理得到：

$$Obj^{(t)} \simeq \sum_{i=1}^{n} \left[ g_i f_t\left(x_i\right) + \frac{1}{2} h_i f_t^2\left(x_i\right) \right] + \Omega\left(f_t\right).$$

定义树的复杂度有两部分组成，一部分是叶子节点的复杂度，另一部分是叶子节点预测值的 L2 范数。表达式如下：

$$\Omega\left(f_t\right)=\gamma T+\frac{1}{2}\lambda\sum_{j=1}^{T}w_j^2.$$

将所有属于第 $j$ 个叶子结点的样本 $x_i$，构成的集合表示如下为 $I_j=\{i|q\left(x_i\right)=j\}$。对叶子结点分组之后，(2-9) 式可以写成：

$$\begin{aligned}Obj^{(t)}&=\sum_{j=1}^{T}\left[\left(\sum_{i\in I_j}g_i\right)w_j+\frac{1}{2}\left(\sum_{i\in I_j}h_i+\lambda\right)w_j^2\right]+\gamma T\\&=\sum_{j=1}^{T}\left[G_jw_j+\frac{1}{2}\left(H_j+\lambda\right)w_j^2\right]+\gamma T,\end{aligned}$$

其中 $G_j=\sum_{i\in I_j}g_i, H_j=\sum_{i\in I_j}h_i$。

显然，$Obj$ 是关于 $w_j$ 的一元二次函数，于是我们容易得到最优的 $Obj$，和对应的 $w_j$：

$$w_j^*=-\frac{G_j}{H_j+\lambda},\quad Obj=-\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_j+\lambda}+\gamma T.$$

于是可以推出，构造决策树时，叶子结点的分裂规则：

$$Gain=\frac{1}{2}\left[\frac{G_L^2}{H_L+\lambda}+\frac{G_R^2}{H_R+\lambda}-\frac{\left(G_L+G_R\right)^2}{H_L+H_R+\lambda}\right]-\gamma.$$

当 $Gain>0$ 时，分裂结点，反之不分裂。

相比于传统的 GBDT，Xgboost 加入了 L1 和 L2 正则化，可以有效的防止过拟合；Xgboost 借鉴了随机森林中的列采样，支持随机特征，这一点也可以增加模型的泛化能力；对于样本缺失值的处理，Xgboost 也能够学习出分裂方向；在特征粒度上，Xgboost 可以并行地计算每一个特征分裂的增益，加快模型的求解速度。

由于其强大的性能，目前，Xgboost 的相关算法实现已比较成熟。例如使用 python 就可以调用工业级别的包，模型有默认的参数，也可以自己设置想要的参数。

### Lightgbm

Ke et al.(2017) 提出了更高效的方法 Lightgbm [12]。前面提到的 GBDT 和 Xgboost 的表现已经非常不错。Lightgbm 可以认为是 Xgboost 的改进版本。Xgboost 会对特征

进行预排序，这样的排序是比较耗时的。在 Lightgbm 中，采用了直方图法，可以节约很多时间，并且能够节省内存空间，在计算子节点的差时，也可以节省不少开销。Lightgbm 对树的生长是 Leaf-wise 的，虽然这在一定程度上会造成过拟合现象，但可以通过控制最大深度来解决。相比于 Xgboost 的 Level-wise，能够节省运算资源。另外，Lightgbm 在采样方法和特征合并上，做了一些处理，这些处理一定程度上增加了模型的精度。

# 第三章　基于回归树的充分降维方法

本章基于回归树的特点，给出了基于回归树的 SIR，SAVE，DR 方法。我们的思想也可以用于其他充分降维方法。

## §3.1　核心思想

首先让我们回顾充分降维中的切片方法。例如 SIR 方法中，根据 $Y$ 所在区间，将该区间切成 $H$ 片，记为 $I_1, I_2, \cdots, I_H$，$p_h$ 记作 $Y$ 落入对应 $I_h$ 的概率，计算每个切片中 $Z$ 的均值得到 $\hat{m}_h, h = 1, 2, \cdots, H$。在 SAVE 和 DR 方法中，我们会在切片后估计 $E(ZZ^T|I_h)$。在响应变量 $Y$ 是一维时，切片的方法操作简单。例如对于样本数量 $n = 1000$ 的情形，可以取切片 $H = 50$ 即可。但当响应变量 $Y$ 的维数增加时，多重切片的数量将呈指数型上升。当响应变量 $Y$ 的维数是 4 时，切片数量将是 $50^4$，这样的切片方法还会使得很多切片中不存在样本，并且会大大增加计算量。

我们再来回顾一下回归树，回归树的示意图如下。

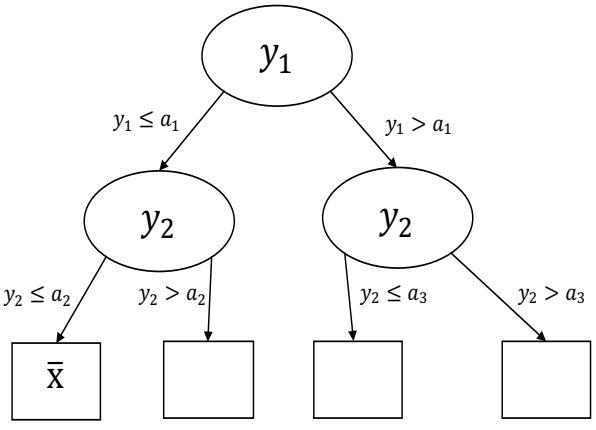


图 3-1　回归树示意图

由回归树的构造过程我们知道，通过对 $Y$ 的空间进行划分，每一个样本最终落到一个叶子结点。而叶子结点的值恰好就是用所有落入该叶子结点的均值来估计。基于此，我们想到，可以将回归树的思想用在响应变量 $Y$ 是多维情形下的充分降维方法。回归

树可以通过各种准则（不同的决策树目标函数不同会导致切分准则有差异），对响应变量 $Y$ 的空间进行切分。这样就避免了高维情况下切片遇到的维度灾难问题。

同时，我们还可以利用一些回归树方法的特性。例如，Xgboost 方法对特征缺失值进行了处理，在特征部分缺失时，不需要我们去主动处理，Xgboost 方法已经解决了这个问题。也就是说，如果我们选择 Xgboost 方法，它也能在响应变量 $Y$ 有缺失值时，充分降维方法依然有效。

## §3.2 基于回归树的 SIR 方法

对于 SIR 方法，其核心在于估计 $\mathrm{Cov}[E(Z|Y)]$。设 $X \in R^p$，$Y \in R^q$，将 $X$ 标准化得到 $Z$。我们可以将 $\mathrm{Cov}[E(Z|Y)]$ 写成这样的形式：

$$\mathrm{Cov}[E(Z|Y)] = E\left[E(Z|Y)E(Z|Y)^T\right] = E\left[f(Y)f(Y)^T\right] = M,$$

其中 $f(Y) = E(Z|Y) \in R^p$。

对于 $i = 1, 2, \cdots, p$，$f_i(Y) = E(Z_i|Y)$，可以得到 $Z_i = f_i + \epsilon_i$。有了每个 $f_i$，可以预测得到 $Z_i$，$i = 1, 2, \cdots, p$。对于样本 $X$，$Z = \hat{\Sigma}^{-\frac{1}{2}}(X - \hat{\mu})$，对数据 $[(Y^1, Z_i^1), (Y^2, Z_i^2), \cdots, (Y^n, Z_i^n)]$ 进行拟合，这里 $n$ 表示样本个数。这里我们可以采用回归树模型进行拟合得到 $\hat{f}_i$，回归树模型可以是梯度提升树，随机森林，Xgboost 等。这样我们就可以得到 $M = \mathrm{Cov}[E(Z|Y)]$ 的一个估计 $\hat{M}$：

$$\hat{M} = E_n\hat{f}(Y)\hat{f}(Y)^T = \frac{1}{n}\sum_{s=1}^{n}\hat{f}(Y^s)\hat{f}(Y^s)^T. \tag{3-1}$$

对于（3-1）中 $\hat{M}$ 的第 $i$ 行，第 $j$ 列元素，

$$\hat{M}_{ij} = E_n\hat{f}_i(Y)\hat{f}_j(Y) = \frac{1}{n}\sum_{s=1}^{n}\hat{f}_i(Y^s)\hat{f}_j(Y^s).$$

对矩阵 $\hat{M}$ 进行特征分解，得到前 $k$ 个特征向量 $\nu_1, \nu_2, \cdots, \nu_k$，左乘 $\hat{\Sigma}^{-\frac{1}{2}}$ 得到，$\hat{\Sigma}^{-\frac{1}{2}}\nu_1, \hat{\Sigma}^{-\frac{1}{2}}\nu_2, \cdots, \hat{\Sigma}^{-\frac{1}{2}}\nu_k$。

## §3.3 基于回归树的 SAVE 方法

用类似的思想，我们可以推广到 SAVE 方法。对于 SAVE 方法，其核心在于估计 $E\left[I_p-\mathrm{Cov}(Z|Y)\right]^2$。将 $E\left[I_p-\mathrm{Cov}(Z|Y)\right]^2$ 写成这样的形式：

$$E\left[I_p-\mathrm{Cov}(Z|Y)\right]^2=E\left[I_p-E(Z\cdot Z^T|Y)+E(Z|Y)E(Z|Y)^T\right]^2=M. \tag{3-2}$$

对于 (3-2) 式中的核矩阵，我们需要估计 $E(Z\cdot Z^T|Y)$ 和 $E(Z|Y)E(Z|Y)^T$。其中 $E(Z|Y)E(Z|Y)^T$ 这一项的估计方法与 SIR 方法相同。对于 $E(Z\cdot Z^T|Y)$，我们要估计 $g_{ij}(Y)=E(Z_i\cdot Z_j|Y)$，其中 $i,j=1,2,\cdots,p, Z\cdot Z^T\in R^{p\times p}$。对数据 $[(Y^1,Z_i^1Z_j^1),(Y^2,Z_i^2Z_j^2),\cdots,(Y^n,Z_i^nZ_j^n)]$ 进行拟合，这里 $n$ 表示样本个数。这里我们可以采用回归树模型进行拟合得到 $\hat{g_{ij}}$，回归树模型可以是梯度提升树，随机森林，Xgboost 等。由于对称性，对于 $p$ 维的 $X$，我们实际上只需要拟合 $p(p+1)/2$ 个 $\hat{g_{ij}}$。$M$ 的样本估计为：

$$\hat{M}=\frac{1}{n}\sum_{s=1}^{n}[I_p-\hat{G}(Y^s)+\hat{f}(Y^s)\hat{f}(Y^s)^T]^2,$$

其中 $G(Y)$ 的第 $i$ 行，第 $j$ 列元素为 $\hat{g_{ij}}(Y)$。

对矩阵 $\hat{M}$ 进行特征分解，得到前 $k$ 个特征向量 $\nu_1,\nu_2,\cdots,\nu_k$，左乘 $\hat{\Sigma}^{-\frac{1}{2}}$ 得到，$\hat{\Sigma}^{-\frac{1}{2}}\nu_1,\hat{\Sigma}^{-\frac{1}{2}}\nu_2,\cdots,\hat{\Sigma}^{-\frac{1}{2}}\nu_k$。

## §3.4 基于回归树的 DR 方法

同样的，对于 DR 方法，其核心在于估计 $E[2I_p-A(Y,\tilde{Y})]^2$，核矩阵也可以写成如下的形式：

$$\begin{aligned}E[2I_p-A(Y,\tilde{Y})]^2=&2E\left[E^2\left(ZZ^T|Y\right)\right]+2E^2\left[E(Z|Y)E\left(Z^T|Y\right)\right]\\&+2E\left[E\left(Z^T|Y\right)E(Z|Y)\right]E\left[E(Z|Y)E\left(Z^T|Y\right)\right]-2I_p=M.\end{aligned} \tag{3-3}$$

对于 (3-3) 式中的核矩阵，我们有三个部分需要估计。对于第一项 $2E[E^2(ZZ^T|Y)]$，在 SAVE 方法中，已经说明如何估计 $E[E(ZZ^T)|Y]$，我们只需要将矩阵 $G$ 变成 $2G^2$。对于第二项 $2E^2\left[E(Z|Y)E\left(Z^T|Y\right)\right]$，在 SIR 方法中，已经说明如何估计 $E\left[E(Z|Y)E\left(Z^T|Y\right)\right]$，对该结果平方乘以 2 即可。对于第三项 $2E\left[E\left(Z^T|Y\right)E(Z|Y)\right]E\left[E(Z|Y)E\left(Z^T|Y\right)\right]$，$2E\left[E\left(Z^T|Y\right)E(Z|Y)\right]$ 部分是一个实数，$E\left[E(Z|Y)E\left(Z^T|Y\right)\right]$ 部分即是 $E\left[f(Y)f(Y)^T\right]$。

综上，我们可以得到 $M$ 的样本估计为：

$$\hat{M} = \frac{1}{n}\sum_{s=1}^{n}[2\hat{G}^2(Y^s) - 2I_p] + 2[\frac{1}{n}\sum_{s=1}^{n}\hat{f}(Y^s)\hat{f}(Y^s)^T]^2 \\ +2[\frac{1}{n}\sum_{s=1}^{n}\hat{f}(Y^s)^T\hat{f}(Y^s)][\frac{1}{n}\sum_{s=1}^{n}\hat{f}(Y^s)\hat{f}(Y^s)^T],$$

其中 $G(Y)$ 的第 $i$ 行，第 $j$ 列元素为 $\hat{g_{ij}}(Y)$。

对矩阵 $\hat{M}$ 进行特征值分解，得到的前 $k$ 个特征向量 $\nu_1, \nu_2, \cdots, \nu_k$，左乘 $\hat{\Sigma}^{-\frac{1}{2}}$ 得到，$\hat{\Sigma}^{-\frac{1}{2}}\nu_1, \hat{\Sigma}^{-\frac{1}{2}}\nu_2, \cdots, \hat{\Sigma}^{-\frac{1}{2}}\nu_k$。

## §3.5 响应变量缺失值处理方法

Xgboost 方法在构造决策树，会计算叶子结点分裂后左子树的得分 $\frac{G_L^2}{H_L+\lambda}$，右子树的得分 $\frac{G_R^2}{H_R+\lambda}$，以及不分裂时的得分 $\frac{(G_L+G_R)^2}{H_L+H_R+\lambda}$。之后计算增益，根据其符号是否大于 0，决定是否进一步分裂叶子结点。

当特征完全时，对于任何一个样本，我们总可以将其分到其中一颗子树。特征出现缺失值时。一种办法是直接除去这样的样本，但这样我们会损失一些信息。Xgboost 方法在处理时，会分别计算将特征缺失样本划入左子树和右子树的得分。虽然它在一定程度上增加了一些计算量，但有效地解决了样本中存在缺失值的问题。在高维场景中，我们很难要求每个维度的信息都可以拿到，缺失值的存在很常见，我们十分有必要利用这些样本的信息。

对于上述我们提到的充分降维方法。当出现响应变量 $Y$ 有缺失值，我们可以针对性的选择能够处理缺失值问题的树模型，例如 Xgboost，Lightgbm。

# 第四章　数值模拟与实例分析

本章我们将通过数值模拟的方法，分析我们提出的方法，在响应变量 $Y$ 是多维时，不同情形下，充分降维方法的效果。包括响应变量 $Y$ 与自变量 $X$ 是线性或是非线性，自变量 $X$ 与响应变量 $Y$ 在不同维数下，噪声扰动，树模型参数不同，响应变量 $Y$ 存在缺失值时。本章假设模型的结构维数已知，$X$ 与 $\epsilon$ 独立，且均服从正态分布。最后，我们把基于回归树的充分降维方法运用到实际数据中，进行了实例分析。

## §4.1　响应变量二维的情形

### §4.1.1　线性模型

给定模型 $X$ 是 10 维的应变量，$X_i \sim N(0,1), i = 1,2,3,\cdots,10$，$X$ 每个维度满足独立同分布，$Y$ 与 $X$ 的关系如下：

$$\begin{aligned} Y_1 &= X_1 + X_2 + X_3 + \sigma\epsilon \\ Y_2 &= X_2 + X_3 + X_4 + \sigma\epsilon \end{aligned}, \tag{4-1}$$

为了增加一些扰动，增加 $Y_3, Y_4, Y_5, Y_6, Y_7$，$Y_i \sim N(0,\sigma^2), i = 3,4,5,6,7$，在仿真时 $Y$ 是 7 维的。真实的降维矩阵是：

$$\begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 1 \\ 0 & 1 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix}_{10\times 2}.$$

对应的投影矩阵是:

$$\begin{bmatrix} 0.6 & 0.2 & 0.2 & -0.4 & \cdots & 0 \\ 0.2 & 0.4 & 0.4 & 0.2 & \cdots & 0 \\ 0.2 & 0.4 & 0.4 & 0.2 & \cdots & 0 \\ -0.4 & 0.2 & 0.2 & 0.6 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}_{10\times 10}.$$

采用 Xgboost 模型，设置树模型最大深度为 3，设置 $\sigma = 0.02$，分别设置 $n = 200, 400, 600, 800, 1000$，用不同充分降维方法进行模拟。每个试验重复 50 次，求出真实投影矩阵和估计投影矩阵的迹相关系数，再求出平均值。

表 4-1 线性模型 $\sigma = 0.02$ 的实验结果

| | n=200 | n=400 | n=600 | n=800 | n=1000 |
|---|---|---|---|---|---|
| SIR | 0.997914 | 0.999314 | 0.999412 | 0.999638 | 0.999718 |
| SAVE | 0.181961 | 0.690899 | 0.983058 | 0.989588 | 0.994751 |
| DR | 0.962856 | 0.985868 | 0.992157 | 0.993679 | 0.996007 |

由表4-1我们可以看出当 $n = 200$ 时，采用 Xgboost 模型的 SIR 和 DR 方法，已经有了相当好的结果。当 $n = 600$ 时，三种方法的效果都接近 0.99。从结果看，随着 $n$ 的增大，真实投影矩阵和估计投影矩阵的迹相关系数会收敛到 1。

### 改变 $\sigma$

噪声 $\sigma$ 的大小可能会对我们的方法产生影响。于是我们模拟了在其他条件不变下，$\sigma = 0.2, 0.5, 1, 2$ 的情形。

由表4-2我们可以看出当 $\sigma = 0.2, 0.5$ 时，三种效果的方法都和 $\sigma = 0.02$ 时很接近。$\sigma = 1$ 时，效果开始有所下降，但也能得到不错的结果。即便在 $\sigma = 2$ 时，SIR 方法依然表现不错。

### 改变树的深度

由树模型的原理，我们知道，树模型的深度影响对响应变量 $Y$ 空间的划分。理论上讲，树的深度越大，空间划分越多。这一点，类似于一维场景下切片 $H$ 的大小。在一维场景时，切片数量 $H$ 越大，平均每个切片的样本个数就越少。对于表4-1的结果，

表 4-2 线性模型 $\sigma = 0.2, 0.5, 1, 2$ 的实验结果

| $\sigma = 0.2$ | n=200 | n=400 | n=600 | n=800 | n=1000 |
|---|---|---|---|---|---|
| SIR | 0.996317 | 0.998855 | 0.999137 | 0.999416 | 0.999581 |
| SAVE | 0.173197 | 0.675351 | 0.969806 | 0.990301 | 0.994242 |
| DR | 0.962427 | 0.984474 | 0.991037 | 0.992813 | 0.995435 |
| $\sigma = 0.5$ | n=200 | n=400 | n=600 | n=800 | n=1000 |
| SIR | 0.986283 | 0.996416 | 0.996853 | 0.998255 | 0.998649 |
| SAVE | 0.135462 | 0.432994 | 0.916031 | 0.975431 | 0.988465 |
| DR | 0.943416 | 0.977020 | 0.986263 | 0.988741 | 0.993559 |
| $\sigma = 1$ | n=200 | n=400 | n=600 | n=800 | n=1000 |
| SIR | 0.953293 | 0.985998 | 0.989574 | 0.993040 | 0.995565 |
| SAVE | 0.083543 | 0.194837 | 0.580867 | 0.732993 | 0.819335 |
| DR | 0.853605 | 0.939422 | 0.967493 | 0.971112 | 0.983498 |
| $\sigma = 2$ | n=200 | n=400 | n=600 | n=800 | n=1000 |
| SIR | 0.782191 | 0.947952 | 0.962658 | 0.970070 | 0.980079 |
| SAVE | 0.138783 | 0.096471 | 0.231584 | 0.437761 | 0.440965 |
| DR | 0.563789 | 0.711121 | 0.807086 | 0.829480 | 0.888178 |

我们控制了树的最大深度为 3(这是目前主流 python 包中的默认参数设置)，我们再设置最大深度为 4、5 进行重复实验。

表 4-3 线性模型控制树最大深度 $max\_depth = 4, 5$ 的实验结果

| $max\_depth = 4$ | n=200 | n=400 | n=600 | n=800 | n=1000 |
|---|---|---|---|---|---|
| SIR | 0.996944 | 0.999080 | 0.999397 | 0.999768 | 0.999792 |
| SAVE | 0.328333 | 0.400003 | 0.735002 | 0.968952 | 0.991451 |
| DR | 0.933359 | 0.981547 | 0.990807 | 0.992893 | 0.995699 |
| $max\_depth = 5$ | n=200 | n=400 | n=600 | n=800 | n=1000 |
| SIR | 0.994509 | 0.998376 | 0.999349 | 0.999641 | 0.999726 |
| SAVE | 0.937596 | 0.519530 | 0.591092 | 0.813256 | 0.931229 |
| DR | 0.791950 | 0.975663 | 0.988851 | 0.992470 | 0.995214 |

由表4-3，从收敛情况来看，设置过大的深度，反而可能会使效果变差。例如当 $n = 600$ 时，设置最大深度为 3 的效果明显好于最大深度为 4，5。在控制树的最大深度时，我们主要依据响应变量 $Y$ 的维数。

### 改变自变量的维数

自变量 $X$ 的维数显然对于充分降维的效果有影响。自变量 $X$ 的维数越大，降维的难度显然越大。原实验中，$p=10$，我们又设定了 $p=15,20$，重复进行实验。

**表 4-4** 线性模型 $p=15,20$ 的实验结果

| $p=15$ | n=200 | n=400 | n=600 | n=800 | n=1000 |
|---|---|---|---|---|---|
| SIR | 0.996494 | 0.998856 | 0.999323 | 0.999493 | 0.999623 |
| SAVE | 0.015660 | 0.071400 | 0.696228 | 0.968251 | 0.982941 |
| DR | 0.953065 | 0.978812 | 0.985627 | 0.991602 | 0.991922 |
| $p=20$ | n=200 | n=400 | n=600 | n=800 | n=1000 |
| SIR | 0.994072 | 0.998194 | 0.999120 | 0.999411 | 0.999512 |
| SAVE | 0.011818 | 0.026198 | 0.181626 | 0.620499 | 0.886964 |
| DR | 0.909884 | 0.963070 | 0.983017 | 0.987624 | 0.990612 |

对比表4-1和表4-4的结果，我们可以发现，尽管总体效果有所下降，但 SIR 和 DR 方法的表现依旧很好。SAVE 方法在样本量大时，也有比较好的降维效果。

### 不同的树模型

前面的实验，我们都采用了 Xgboost 模型。目前，我们还有很多其他树模型。例如 GBDT(梯度提升树)，RF(随机森林)，Lightgbm 等。我们试验了其他几种树模型。

**表 4-5** 线性模型使用不同树模型的实验结果

| GBDT | n=200 | n=400 | n=600 | n=800 | n=1000 |
|---|---|---|---|---|---|
| SIR | 0.997356 | 0.999387 | 0.999478 | 0.999648 | 0.999726 |
| SAVE | 0.087799 | 0.407793 | 0.914428 | 0.980101 | 0.991417 |
| DR | 0.947777 | 0.981307 | 0.989134 | 0.991445 | 0.994839 |
| RF | n=200 | n=400 | n=600 | n=800 | n=1000 |
| SIR | 0.980893 | 0.992779 | 0.993487 | 0.995532 | 0.994521 |
| SAVE | 0.708439 | 0.953769 | 0.964928 | 0.979075 | 0.979529 |
| DR | 0.897894 | 0.971070 | 0.982651 | 0.984306 | 0.987408 |

对比表4-1和表4-5，我们看到最终的结果，总体来说：RF>Xgboost>GBDT。尤其在例子中，基于 RF 的 SAVE 方法表现很好。结果也从一定程度上反映了集成学习的优越性。

由上述分析可知，噪声 $\sigma$ 的大小，自变量 $X$ 的维数，树模型的选取，树模型的参数设置，比如树的最大深度控制，对不同的充分降维方法的效果均有关系。但总的来说，我们的方法在一定程度的噪声和比较高的维数下，均能取得不错的效果。

从这个例子看，SAVE 方法的表现似乎不如 SIR 方法和 DR 方法。但实际上这与模型选择和参数设置是有关系的。下文中，我们会发现，一些例子中，SAVE 方法的表现是优于 SIR 方法的。

### §4.1.2 非线性模型

这一节，主要研究非线性模型下的充分降维效果。常见的非线性函数有：三角函数，指数函数，对数函数等。我们对 (4-1) 式的模型稍加修改，得到几种常见的非线性模型。

**例一**

三角函数

$$\begin{aligned} Y_1 &= \sin(X_1 + X_2 + X_3) + \sigma\epsilon \\ Y_2 &= X_2 + X_3 + X_4 + \sigma\epsilon \end{aligned}, \tag{4-2}$$

指数函数

$$\begin{aligned} Y_1 &= \exp(X_1 + X_2 + X_3) + \sigma\epsilon \\ Y_2 &= X_2 + X_3 + X_4 + \sigma\epsilon \end{aligned}, \tag{4-3}$$

对数函数

$$\begin{aligned} Y_1 &= \ln((X_1 + X_2 + X_3)^2 + 3) + \sigma\epsilon \\ Y_2 &= X_2 + X_3 + X_4 + \sigma\epsilon \end{aligned}, \tag{4-4}$$

带有交叉项的非线性函数

$$\begin{aligned} Y_1 &= (X_1 + X_2 + X_3)(X_2 + X_3 + X_4) + \sigma\epsilon \\ Y_2 &= X_2 + X_3 + X_4 + \sigma\epsilon \end{aligned}. \tag{4-5}$$

对于上述模型，他们的真实降维子空间与 model(4-1) 是一样的。在其他模拟条件一致的情况下，使用 Xgboost 和 RF 对这些非线性模型进行降维，计算迹相关系数。结合表4-1，表4-6、表4-6的结果，我们发现，这些模型当中，指数函数和对数函数相对更加稳健，三角函数和带有交叉项的非线性函数会比线性模型更加不容易降维。Xgboost 和 RF 的表现互有优劣，样本量较大时，Xgboost 效果较好；样本量较小时，RF 效果较好。

表 4-6 不同非线性模型的实验结果—Xgboost

| model(4-2) | n=200 | n=400 | n=600 | n=800 | n=1000 |
|---|---|---|---|---|---|
| SIR | 0.944805 | 0.972956 | 0.984620 | 0.988310 | 0.991536 |
| SAVE | 0.190583 | 0.536472 | 0.692556 | 0.815432 | 0.901567 |
| DR | 0.673122 | 0.796943 | 0.924678 | 0.927882 | 0.963291 |
| model(4-3) | n=200 | n=400 | n=600 | n=800 | n=1000 |
| SIR | 0.997837 | 0.999299 | 0.999399 | 0.999649 | 0.999680 |
| SAVE | 0.210004 | 0.748168 | 0.979108 | 0.990298 | 0.994959 |
| DR | 0.964566 | 0.985313 | 0.992065 | 0.993802 | 0.996138 |
| model(4-4) | n=200 | n=400 | n=600 | n=800 | n=1000 |
| SIR | 0.838602 | 0.960178 | 0.983557 | 0.985463 | 0.991330 |
| SAVE | 0.208837 | 0.566254 | 0.872336 | 0.920852 | 0.964136 |
| DR | 0.789727 | 0.926535 | 0.965847 | 0.972637 | 0.983970 |
| model(4-5) | n=200 | n=400 | n=600 | n=800 | n=1000 |
| SIR | 0.941141 | 0.982741 | 0.985228 | 0.985951 | 0.986376 |
| SAVE | 0.135316 | 0.503217 | 0.604756 | 0.755894 | 0.930744 |
| DR | 0.889151 | 0.940876 | 0.972060 | 0.979751 | 0.988278 |

表 4-7 不同非线性模型的实验结果—RF

| model(4-2) | n=200 | n=400 | n=600 | n=800 | n=1000 |
|---|---|---|---|---|---|
| SIR | 0.908173 | 0.966629 | 0.978602 | 0.984698 | 0.987948 |
| SAVE | 0.555225 | 0.650445 | 0.765554 | 0.815070 | 0.873247 |
| DR | 0.612342 | 0.699963 | 0.832301 | 0.901070 | 0.943060 |
| model(4-3) | n=200 | n=400 | n=600 | n=800 | n=1000 |
| SIR | 0.982690 | 0.992160 | 0.993223 | 0.995405 | 0.994498 |
| SAVE | 0.729190 | 0.953566 | 0.967011 | 0.979246 | 0.979181 |
| DR | 0.899305 | 0.969246 | 0.982507 | 0.985132 | 0.988637 |
| model(4-4) | n=200 | n=400 | n=600 | n=800 | n=1000 |
| SIR | 0.715150 | 0.855565 | 0.876231 | 0.879059 | 0.894268 |
| SAVE | 0.710436 | 0.872585 | 0.925138 | 0.932613 | 0.957497 |
| DR | 0.810698 | 0.912142 | 0.960574 | 0.963920 | 0.976405 |
| model(4-5) | n=200 | n=400 | n=600 | n=800 | n=1000 |
| SIR | 0.801550 | 0.859567 | 0.860481 | 0.856018 | 0.850217 |
| SAVE | 0.638462 | 0.824771 | 0.915094 | 0.932757 | 0.964947 |
| DR | 0.819227 | 0.933245 | 0.963533 | 0.973472 | 0.978223 |

# §4.2 响应变量三维及以上的情形

## §4.2.1 响应变量三维

**例一**

给定模型 $X$ 是 10 维的应变量，$X_i \sim N(0,1), i=1,2,3,\cdots,10$，$X$ 每个维度满足独立同分布，$Y$ 与 $X$ 的关系如下：

$$\begin{aligned} Y_1 &= X_1/[0.5+(X_2+X_3)^2]+\sigma\epsilon \\ Y_2 &= X_1(X_4+X_5+1)+\sigma\epsilon \\ Y_3 &= \sqrt{(X_4+X_5)^2}+\ln(X_4+X_5)^2+\sigma\epsilon \end{aligned}, \tag{4-6}$$

这里我们有意设置了 $Y_3$ 关于 $X$ 对称。理论上讲，SIR 方法不能找到所有的降维方向。

为了增加一些扰动，增加 $Y_4, Y_5, Y_6, Y_7$，$Y_i \sim N(0,\sigma^2), i=4,5,6,7$，在仿真时 $Y$ 是 7 维的。

这个模型真实的降维矩阵是：

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \end{bmatrix}_{10\times 3}.$$

对应的投影矩阵是:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0.5 & 0.5 & 0 & 0 & \cdots & 0 \\ 0 & 0.5 & 0.5 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0.5 & 0.5 & \cdots & 0 \\ 0 & 0 & 0 & 0.5 & 0.5 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}_{10\times 10}.$$

模拟方法相同，分别用 GBDT、RF、Xgboost 模型设置树模型最大深度为 3，设置 $\sigma = 0.02$，分别设置 $n = 200, 400, 600, 800, 1000$，用不同充分降维方法进行模拟。每个试验重复 50 次，求出真实投影矩阵和估计投影矩阵的迹相关系数，再求出平均值。

**表 4-8** model(4-6) 在不同树模型下的实验结果

| GBDT | n=200 | n=400 | n=600 | n=800 | n=1000 |
|---|---|---|---|---|---|
| SIR | 0.692930 | 0.684404 | 0.699649 | 0.679879 | 0.696290 |
| SAVE | 0.260875 | 0.464078 | 0.683536 | 0.930012 | 0.968753 |
| DR | 0.702598 | 0.704882 | 0.782502 | 0.806182 | 0.831646 |
| RF | n=200 | n=400 | n=600 | n=800 | n=1000 |
| SIR | 0.708162 | 0.702444 | 0.689632 | 0.679961 | 0.674841 |
| SAVE | 0.639975 | 0.728384 | 0.860607 | 0.919792 | 0.983714 |
| DR | 0.694791 | 0.721798 | 0.835713 | 0.868431 | 0.966601 |
| Xgb | n=200 | n=400 | n=600 | n=800 | n=1000 |
| SIR | 0.687002 | 0.682923 | 0.701712 | 0.681944 | 0.695541 |
| SAVE | 0.320270 | 0.616421 | 0.877598 | 0.948812 | 0.979151 |
| DR | 0.712475 | 0.719075 | 0.829233 | 0.854662 | 0.900469 |

和我们预期的相同，SIR 方法对于 model(4-6)，是不能找全所有的降维方向的。Xgboost 和 RF 的效果差距不大，两者均好于 GBDT，对于这个相对复杂的模型，充分降维的效果仍然不错。在 $n = 600$ 时，迹相关系数能达到 0.8 以上。相比于 SIR 和 SAVE 方法，DR 方法相对来说更加稳健这也是可以理解的。因为 DR 方法从原理上来说可以看成是 SIR 和 SAVE 方法一种组合。

**例二**

我们再来看一个三维响应变量非线性模型的例子：给定模型 $X$ 是 10 维的应变量，$X \sim N(0,1), Y$ 与 $X$ 的关系如下：

$$\begin{aligned} Y_1 &= X_1/[1+(X_2+1)^2] + \sigma\epsilon \\ Y_2 &= X_1(X_1+X_2+1) + \sigma\epsilon \\ Y_3 &= (X_1^2+X_2^2)^{1/2} + \ln(X_1^2+X_2^2)^{1/2} + \sigma\epsilon \end{aligned}, \tag{4-7}$$

为了增加一些扰动，增加 $Y_4, Y_5, Y_6, Y_7$，$Y_i \sim N(0,\sigma^2), i = 4,5,6,7$。在仿真时 $Y$ 是

7 维的。此时真实的降维矩阵是：

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix}_{10\times 2}.$$

对应的投影矩阵是:

$$\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}_{10\times 10}.$$

我们分别用 GBDT、RF、XGBoost 模型、不同的充分降维方法对不同 $n$ 下做模拟，计算真实的投影矩阵和估计的投影矩阵的迹相关系数。

**表 4-9** model(4-7) 在不同树模型下的实验结果

| GBDT | n=200 | n=400 | n=600 | n=800 | n=1000 |
|---|---|---|---|---|---|
| SIR | 0.991059 | 0.996274 | 0.997781 | 0.998335 | 0.998621 |
| SAVE | 0.037939 | 0.195383 | 0.749685 | 0.943650 | 0.978441 |
| DR | 0.932765 | 0.980948 | 0.988674 | 0.992171 | 0.994168 |
| RF | n=200 | n=400 | n=600 | n=800 | n=1000 |
| SIR | 0.966184 | 0.986982 | 0.991949 | 0.994552 | 0.995846 |
| SAVE | 0.784692 | 0.956259 | 0.976597 | 0.984986 | 0.989172 |
| DR | 0.919543 | 0.974321 | 0.984009 | 0.988425 | 0.991529 |
| Xgb | n=200 | n=400 | n=600 | n=800 | n=1000 |
| SIR | 0.990417 | 0.996079 | 0.997673 | 0.998218 | 0.998598 |
| SAVE | 0.059556 | 0.518509 | 0.932514 | 0.980007 | 0.989637 |
| DR | 0.952904 | 0.985294 | 0.991102 | 0.993701 | 0.995171 |

由表4-9的结果我们可以看到，对于复杂的非线性关系，我们提出的方法在响应变量三维时依然有效。对于 SIR，SAVE，DR 这三种方法。RF 的总体效果好于 Xgboost 好于 GBDT。

例三

我们再来看一下响应变量三维的非线性模型，给定模型 $X$ 是 10 维的应变量，$X \sim N(0,1), Y$ 与 $X$ 的关系如下：

$$\begin{aligned} Y_1 &= X_1/(3+X_2+X_3)+\sigma\epsilon \\ Y_2 &= X_1 \cdot (X_2+X_3+1)+\sigma\epsilon. \\ Y_3 &= 6(X_4+X_5)+\sigma\epsilon \end{aligned} \tag{4-8}$$

类似地，增加 $Y_4, Y_5, Y_6, Y_7$，$Y_i \sim N(0,\sigma^2), i=4,5,6,7$。在仿真时 $Y$ 是 7 维的。此时真实的降维矩阵是：

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \end{bmatrix}_{10\times 3}.$$

对应的投影矩阵是:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0.5 & 0.5 & 0 & 0 & \cdots & 0 \\ 0 & 0.5 & 0.5 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0.5 & 0.5 & \cdots & 0 \\ 0 & 0 & 0 & 0.5 & 0.5 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}_{10\times 10}.$$

同样地，分别用 GBDT、RF、Xgboost 模型、不同的充分降维方法对不同 $n$ 下做模拟，计算真实降维矩阵的投影矩阵和估计降维矩阵的投影矩阵的迹相关系数。

由表4-10的结果我们可以看到，第一，对于复杂的非线性关系，我们提出的方法在响应变量三维时依然有效。第二，对于 SIR，SAVE，DR 这三种方法。在样本量少时，RF 的总体效果好于 Xgboost 好于 GBDT，样本量大时，不同树模型差别不大，都会达到很好的效果。

表 4-10 model(4-8) 在不同树模型下的实验结果

| GBDT | n=200 | n=400 | n=600 | n=800 | n=1000 |
|---|---|---|---|---|---|
| SIR | 0.992120 | 0.997033 | 0.998272 | 0.998818 | 0.999053 |
| SAVE | 0.375815 | 0.658136 | 0.914556 | 0.980816 | 0.991220 |
| DR | 0.937236 | 0.981782 | 0.990346 | 0.993164 | 0.995136 |
| RF | n=200 | n=400 | n=600 | n=800 | n=1000 |
| SIR | 0.974176 | 0.989540 | 0.993461 | 0.995064 | 0.995427 |
| SAVE | 0.799563 | 0.949937 | 0.975354 | 0.984966 | 0.988260 |
| DR | 0.931263 | 0.975257 | 0.986855 | 0.990616 | 0.992814 |
| Xgb | n=200 | n=400 | n=600 | n=800 | n=1000 |
| SIR | 0.991709 | 0.996781 | 0.998152 | 0.998721 | 0.999008 |
| SAVE | 0.456389 | 0.835147 | 0.975850 | 0.990217 | 0.994713 |
| DR | 0.954511 | 0.985577 | 0.992301 | 0.994500 | 0.996133 |

### §4.2.2 响应变量高维

**例一**

我们继续研究响应变量更高维的情况，看看我们的方法是否还试用。给定模型 $X$ 是 10 维的应变量，$X_i \sim N(0,1), i = 1,2,3,\cdots,10$，$X$ 每个维度满足独立同分布，$Y$ 与 $X$ 的关系如下：

$$\begin{aligned} Y_1 &= X_1/(3+X_1+X_2)+\sigma\epsilon \\ Y_2 &= \exp(X_3+X_4)+\sigma\epsilon \\ Y_3 &= \sin(X_4+X_5)+\sigma\epsilon \\ Y_4 &= (X_5+X_6+2)(X_5+X_6)+\sigma\epsilon \\ Y_5 &= (X_6+X_7+2X_8)(X_2+X_3)+\sigma\epsilon \\ Y_6 &= (X_2+X_3)+\sigma\epsilon \end{aligned} \quad . \tag{4-9}$$

为了增加一些扰动，增加 $Y_7, Y_8, Y_9, Y_{10}$，$Y_i \sim N(0,\sigma^2), i = 7,8,9,10$，在仿真时 $Y$ 是 10 维的。

计算得到真实的投影矩阵是:

$$\begin{bmatrix} 1&0&0&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&0\\ 0&0&0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&0.2&0.4&0&0\\ 0&0&0&0&0&0&0.4&0.8&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0 \end{bmatrix}_{10\times 10}.$$

模拟方法相同，分别用 GBDT、RF、Xgboost 模型设置树模型最大深度为 3，设置 $\sigma = 0.02$，分别设置 $n = 200, 400, 600, 800, 1000$。每个试验重复 50 次，求出真实投影矩阵和估计投影矩阵的迹相关系数，再求出平均值。

**表 4-11** model(4-9) 在不同树模型下的实验结果

| GBDT | n=200 | n=400 | n=600 | n=800 | n=1000 |
|---|---|---|---|---|---|
| SIR | 0.930945 | 0.938459 | 0.967867 | 0.978796 | 0.985591 |
| SAVE | 0.763636 | 0.808112 | 0.844443 | 0.873843 | 0.879290 |
| DR | 0.878740 | 0.892022 | 0.942432 | 0.945863 | 0.960116 |
| RF | n=200 | n=400 | n=600 | n=800 | n=1000 |
| SIR | 0.905194 | 0.943167 | 0.974709 | 0.984346 | 0.984711 |
| SAVE | 0.854273 | 0.939468 | 0.976658 | 0.986730 | 0.990140 |
| DR | 0.857859 | 0.945149 | 0.969431 | 0.985948 | 0.987756 |
| Xgb | n=200 | n=400 | n=600 | n=800 | n=1000 |
| SIR | 0.917965 | 0.922016 | 0.956695 | 0.981124 | 0.986196 |
| SAVE | 0.773135 | 0.817401 | 0.852438 | 0.884437 | 0.890663 |
| DR | 0.877596 | 0.902792 | 0.934090 | 0.944665 | 0.958903 |

由表4-11可以看到，在响应变量 $Y$ 的维数增加到 6 维时，我们的降维方法依然有效。即使在样本数量较小时，也找到了比较接近真实的降维子空间，其中 RF 降维效果最好。

我们增加自变量 $X$ 的维数，看看降维效果是否会下降。在这里设置 $p = 15$，其余条件不变，重复实验。

表 4-12 model(4-9) 在不同树模型下的实验结果 ($p$=15)

| GBDT | n=200 | n=400 | n=600 | n=800 | n=1000 |
|---|---|---|---|---|---|
| SIR | 0.859202 | 0.912903 | 0.919719 | 0.946737 | 0.961605 |
| SAVE | 0.294126 | 0.490501 | 0.664858 | 0.762708 | 0.797126 |
| DR | 0.778405 | 0.844571 | 0.862927 | 0.871199 | 0.881314 |
| RF | n=200 | n=400 | n=600 | n=800 | n=1000 |
| SIR | 0.814924 | 0.897075 | 0.901597 | 0.953200 | 0.950685 |
| SAVE | 0.721216 | 0.834480 | 0.908355 | 0.918283 | 0.970415 |
| DR | 0.771988 | 0.847747 | 0.918055 | 0.924673 | 0.968135 |
| Xgb | n=200 | n=400 | n=600 | n=800 | n=1000 |
| SIR | 0.846425 | 0.913081 | 0.925340 | 0.948755 | 0.966407 |
| SAVE | 0.346851 | 0.623229 | 0.739957 | 0.822423 | 0.843240 |
| DR | 0.776586 | 0.849161 | 0.868507 | 0.880782 | 0.901607 |

由表4-12，我们可以看到，总体效果虽略有下降，但还是在自变量、响应变量维数高，关系复杂的情况下，还能得到不错的效果。

**例二**

我们再来研究一个更复杂的模型。给定模型 $X$ 是 20 维的应变量，$X_i \sim N(0,1), i = 1,2,3,\cdots,20$，$X$ 每个维度满足独立同分布，$Y$ 与 $X$ 的关系如下：

$$\begin{aligned}
Y_1 &= \sin(X_1)/(3+X_1+2X_2+3X_3)+\sigma\epsilon \\
Y_2 &= \exp(X_3+X_4)+\cos(X_5+X_6)\sigma\epsilon \\
Y_3 &= \sin(X_4+X_5)+\sigma\epsilon \\
Y_4 &= (X_5+X_6+2)(X_5+X_6)+\sigma\epsilon \\
Y_5 &= (X_6+X_7+2X_8)\exp(X_2+X_3)+\sigma\epsilon \\
Y_6 &= (X_2+X_3)^3+\sigma\epsilon \\
Y_7 &= \cos(X_2+X_3)+\sigma\epsilon \\
Y_8 &= \sin(X_8+X_9)+\sigma\epsilon
\end{aligned} \tag{4-10}$$

为了增加一些扰动，增加 $Y_9, Y_{10}, Y_{11}, Y_{12}$，$Y_i \sim N(0,\sigma^2), i = 9,10,11,12$，在仿真时 $Y$ 是 12 维的。

计算得到真实的投影矩阵是:

$$\begin{bmatrix} 1&0&0&0&0&0&0&0&0&\cdots&0\\ 0&1&0&0&0&0&0&0&0&\cdots&0\\ 0&0&1&0&0&0&0&0&0&\cdots&0\\ 0&0&0&1&0&0&0&0&0&\cdots&0\\ 0&0&0&0&1&0&0&0&0&\cdots&0\\ 0&0&0&0&0&1&0&0&0&\cdots&0\\ 0&0&0&0&0&0&1/3&1/3&-1/3&\cdots&0\\ 0&0&0&0&0&0&1/3&5/6&1/6&\cdots&0\\ 0&0&0&0&0&0&-1/3&1/6&5/6&\cdots&0\\ \vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&&\vdots\\ 0&0&0&0&0&0&0&0&0&\cdots&0 \end{bmatrix}_{20\times 20}.$$

分别用 GBDT、RF、Xgboost 模型设置树模型最大深度为 3，设置 $\sigma = 0.02$，分别设置 $n = 200, 400, 600, 800, 1000$，用不同充分降维方法进行模拟。每个试验重复 50 次，求出真实投影矩阵和估计投影矩阵的迹相关系数，再求出平均值。

**表 4-13** model(4-10) 在不同树模型下的实验结果 ($p$=20)

| GBDT | n=200 | n=400 | n=600 | n=800 | n=1000 |
|---|---|---|---|---|---|
| SIR | 0.855915 | 0.879850 | 0.894182 | 0.900676 | 0.920635 |
| SAVE | 0.156642 | 0.238034 | 0.487243 | 0.682461 | 0.767851 |
| DR | 0.751827 | 0.817466 | 0.839573 | 0.861835 | 0.867160 |
| RF | n=200 | n=400 | n=600 | n=800 | n=1000 |
| SIR | 0.835072 | 0.919550 | 0.945704 | 0.970883 | 0.983868 |
| SAVE | 0.670890 | 0.779205 | 0.836084 | 0.885675 | 0.929812 |
| DR | 0.736369 | 0.801642 | 0.837977 | 0.881251 | 0.911437 |
| Xgb | n=200 | n=400 | n=600 | n=800 | n=1000 |
| SIR | 0.839091 | 0.870003 | 0.887449 | 0.899859 | 0.932342 |
| SAVE | 0.194582 | 0.393330 | 0.684181 | 0.768689 | 0.825748 |
| DR | 0.765093 | 0.828877 | 0.852102 | 0.868233 | 0.877939 |

由表4-13，我们可以看到，在自变量 $X$ 达到 20 维，响应变量 $Y$ 是 8 维，响应变量 $Y$ 与自变量 $X$ 之间的关系相当复杂时，充分降维方法依然有效。

# §4.3 响应变量有缺失值

在实际场景中，数据的部分缺失是常见的现象。常见的缺失值处理方法有，用平均数，众数或者中位数替代等。本节主要模拟响应变量有缺失值的情形，主要根据部分树模型的特性，例如 Xgboost 对特征缺失处理的办法。我们通过模拟响应变量有缺失值和无缺失值的情形，对降维结果进行比较。

首先，给定模型 $X$ 是 10 维的应变量，$X_i \sim N(0,1), i = 1,2,3,\cdots,10$，$X$ 每个维度满足独立同分布，$Y$ 与 $X$ 的关系如下：

$$\begin{aligned} Y_1 &= X_1/(3+X_1+X_2)+\sigma\epsilon \\ Y_2 &= \exp(X_3+X_4)+\sigma\epsilon \\ Y_3 &= \sin(X_4+X_5)+\sigma\epsilon \\ Y_4 &= (X_5+X_6+2)(X_5+X_6)+\sigma\epsilon \\ Y_5 &= (X_6+X_7)(X_2+X_3)+\sigma\epsilon \\ Y_6 &= (X_2+X_3)+\sigma\epsilon \end{aligned} . \tag{4-11}$$

为了增加一些扰动，增加 $Y_7, Y_8, Y_9, Y_{10}$，$Y_i \sim N(0,\sigma^2), i = 7,8,9,10$，在仿真时 $Y$ 是 10 维的。

容易得到真实的投影矩阵是:

$$\begin{bmatrix} 1&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0 \end{bmatrix}_{10\times 10} .$$

采用 Xgboost 模型设置树模型最大深度为 3，设置 $\sigma = 0.02$，分别设置 $n = 200,300,400,500,600$。分别对 $Y_i$，$i = 1,2,\cdots,6$ 单独进行随机缺失，这里设定随机缺失 100 个。用不同充分降维方法进行模拟。每个试验重复 100 次，求出真实投影矩阵和估计投影矩阵的迹相关系数，再求出平均值。

表 4-14 model(4-11) 响应变量有缺失值的实验结果-1

| 未缺失 | n=200 | n=300 | n=400 | n=500 | n=600 |
|---|---|---|---|---|---|
| SIR | 0.883774 | 0.905476 | 0.920627 | 0.932497 | 0.943970 |
| SAVE | 0.758207 | 0.778351 | 0.807014 | 0.821517 | 0.840189 |
| DR | 0.860691 | 0.883716 | 0.896938 | 0.908565 | 0.916676 |
| $Y_1$ | n=200 | n=300 | n=400 | n=500 | n=600 |
| SIR | 0.877270 | 0.896020 | 0.909432 | 0.929232 | 0.938642 |
| SAVE | 0.731098 | 0.764599 | 0.798951 | 0.821178 | 0.833771 |
| DR | 0.847808 | 0.877362 | 0.885361 | 0.904207 | 0.914096 |
| $Y_2$ | n=200 | n=300 | n=400 | n=500 | n=600 |
| SIR | 0.867126 | 0.897088 | 0.915081 | 0.933318 | 0.944669 |
| SAVE | 0.769816 | 0.816141 | 0.846282 | 0.861792 | 0.876949 |
| DR | 0.847761 | 0.877603 | 0.891559 | 0.904717 | 0.913160 |
| $Y_3$ | n=200 | n=300 | n=400 | n=500 | n=600 |
| SIR | 0.874911 | 0.894892 | 0.913073 | 0.932344 | 0.936512 |
| SAVE | 0.735732 | 0.761750 | 0.800794 | 0.811554 | 0.831686 |
| DR | 0.850553 | 0.873390 | 0.884756 | 0.906039 | 0.913958 |
| $Y_4$ | n=200 | n=300 | n=400 | n=500 | n=600 |
| SIR | 0.876699 | 0.897545 | 0.906722 | 0.930975 | 0.940614 |
| SAVE | 0.753584 | 0.771708 | 0.804915 | 0.817915 | 0.833960 |
| DR | 0.849964 | 0.873349 | 0.886382 | 0.902740 | 0.913957 |
| $Y_5$ | n=200 | n=300 | n=400 | n=500 | n=600 |
| SIR | 0.860851 | 0.899057 | 0.908032 | 0.930904 | 0.932852 |
| SAVE | 0.749914 | 0.759003 | 0.793457 | 0.814391 | 0.828204 |
| DR | 0.843118 | 0.872941 | 0.883739 | 0.902619 | 0.916053 |
| $Y_6$ | n=200 | n=300 | n=400 | n=500 | n=600 |
| SIR | 0.855742 | 0.883717 | 0.902604 | 0.926069 | 0.935173 |
| SAVE | 0.758085 | 0.776084 | 0.799495 | 0.820527 | 0.831233 |
| DR | 0.843200 | 0.872182 | 0.884240 | 0.900548 | 0.914176 |

由表4-14，我们发现在响应变量 $Y_1, Y_3, Y_4, Y_5, Y_6$ 存在缺失值时，降维效果相比未缺失时，有一点下降，$Y_2$ 存在缺失值时，降维效果几乎没有变化。我们在模拟时，已经设定了 $Y_i$ 的缺失个数是 100。因此，通过对比含缺失值的样本量为 $n$ 与不含缺失值的样本量为 $n-100$ 的模拟，就能发现，在某一维度含缺失值时，Xgboost 还是能从其他维度获得一些信息，效果会优于直接丢弃这些样本。举个例子，在 $n=400$，响应变量未缺失，SAVE 方法中，迹相关系数为 0.807014，而在 $n=500$，响应变量任一维度缺失 100 个时，SAVE 方法中，迹相关系数均大于 0.807014。

从缺失比例看，在 $n=200,300,400,500,600$ 时，缺失 100 个样本等效的缺失比例分别为 $\frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \frac{1}{5}, \frac{1}{6}$。当然，我们也可以直接控制响应变量缺失比例进行模拟。对于 model(4-11)，$n=200,300,400,500,600$，设定缺失变量为 $Y_6$，设置随机缺失比例为 10%，分别用两种方式进行降维。方式一：利用所有样本；方式二：舍弃含缺失值的样本。我们得到以下结果：

**表 4-15** model(4-11) 响应变量有缺失值的实验结果-2

| | | n=200 | n=300 | n=400 | n=500 | n=600 |
|---|---|---|---|---|---|---|
| 方式一 | SIR | 0.879660 | 0.891768 | 0.909756 | 0.934401 | 0.935558 |
| | SAVE | 0.759671 | 0.770382 | 0.805959 | 0.822241 | 0.832067 |
| | DR | 0.864113 | 0.876824 | 0.887124 | 0.907370 | 0.911634 |
| 方式二 | SIR | 0.872733 | 0.891265 | 0.908866 | 0.927431 | 0.934315 |
| | SAVE | 0.756930 | 0.765304 | 0.793283 | 0.804435 | 0.827738 |
| | DR | 0.836782 | 0.862119 | 0.876631 | 0.896486 | 0.901685 |

由表4-15，我们可以看到，采用 Xgboost 方法，可以有效利用未缺失维度的信息。对于响应变量含缺失值的情形，我们不必直接丢弃这些样本。

## §4.4 与现有方法的比较

上一章中，我们已经介绍了投影重采样法 (Projective Resampling)。在投影重采样法的论文中，已经验证了该方法的效果显著好于多重切片法、基于聚类的充分降维方法。因此本文将着重跟投影重采样法的降维效果做比较。

在投影重采样法中，要求投影次数中 $m_n$ 的大小满足 $m_n/n \to \infty$，当 $n \to \infty$。这里我们取 $m_n = \lceil n\ln(n) \rceil$ 。在切片方法中，切片的数量对最终的降维效果有一定影响。这里切片的选择根据经验，SAVE 和 DR 方法的切片数量应当远小于 SIR 方法的切片数量。在 SIR 方法中，我们选择切片的数量为 $\lceil \sqrt{n} \rceil$。在 SAVE、DR 方法中，我们选择切片的数量为 $h=3,3,5,5$，在 $n=200,400,600,800$ 时。

我们对分别采用基于投影重采样的 SIR、SAVE、DR 方法进行降维，同样计算真实投影矩阵和估计投影矩阵的迹相关系数。

首先，我们选取一个响应变量三维的模型 model(4-6)。

表 4-16 model(4-6) 树模型与投影法的实验结果比较

| | n=200 | n=400 | n=600 | n=800 |
|---|---|---|---|---|
| RF-SIR | 0.708162 | 0.702444 | 0.689632 | 0.679961 |
| RF-SAVE | 0.639975 | 0.728384 | 0.860607 | 0.919792 |
| RF-DR | 0.694791 | 0.721798 | 0.835713 | 0.868431 |
| PR-SIR | 0.641712 | 0.686354 | 0.688877 | 0.704278 |
| PR-SAVE | 0.721440 | 0.712137 | 0.848562 | 0.897928 |
| PR-DR | 0.732466 | 0.715160 | 0.825181 | 0.898104 |

由表4-16，我们可以看到，在响应变量三维的情形下。我们提出的基于回归树的充分降维方法和投影重采样法降维效果比较接近。

我们方法的优势在于对响应变量空间的切分，我们比较响应变量维数较高的模型 model(4-9) 和 model(4-10)。

表 4-17 model(4-9) 树模型与投影法的实验结果比较

| | n=200 | n=400 | n=600 | n=800 |
|---|---|---|---|---|
| RF-SIR | 0.905194 | 0.943167 | 0.974709 | 0.984346 |
| RF-SAVE | 0.854273 | 0.939468 | 0.976658 | 0.986730 |
| RF-DR | 0.857859 | 0.945149 | 0.969431 | 0.985948 |
| PR-SIR | 0.866283 | 0.876021 | 0.900426 | 0.880384 |
| PR-SAVE | 0.820476 | 0.828310 | 0.840566 | 0.892807 |
| PR-DR | 0.839382 | 0.844370 | 0.867343 | 0.903786 |

表 4-18 model(4-10) 树模型与投影法的实验结果比较

| | n=200 | n=400 | n=600 | n=800 |
|---|---|---|---|---|
| RF-SIR | 0.835072 | 0.919550 | 0.945704 | 0.970883 |
| RF-SAVE | 0.670890 | 0.779205 | 0.836084 | 0.885675 |
| RF-DR | 0.736369 | 0.801642 | 0.837977 | 0.881251 |
| PR-SIR | 0.650051 | 0.686261 | 0.719681 | 0.756006 |
| PR-SAVE | 0.560101 | 0.615113 | 0.623498 | 0.643868 |
| PR-DR | 0.607166 | 0.639559 | 0.659560 | 0.683324 |

由表4-17和表4-18，我们可以看到在响应变量高维时，对于 SIR，SAVE，DR 方法，基于回归树的充分降维方法效果明显好于投影重采样法。从计算量的角度考虑，对于投

影重采样法，当样本数 $n$ 增大时，需要大量计算核矩阵，采用基于回归树的方法只需要计算一次核矩阵。这里需要指出的是，我们在设置回归树参数时，采用了默认参数。如果我们通过适当调节参数，可能可以取得更好的效果。

## §4.5 实例分析

本节，我们把基于回归树的充分降维方法运用到实际数据中。Zhu et al.(2010) [38] 在研究多维响应变量的充分降维方法时，使用了血浆视黄醇和 $\beta$-胡萝卜素水平的决定因素数据集。这里我们主要采用跟他相同的数据集，以便比较与分析。

一些研究表明，低饮食摄入量，低 $\beta$-胡萝卜素或其他类胡萝卜素与某些癌症的风险增加有关。但很少有人研究血浆中这些微量元素水平的决定因素。Nierenberg et al.(1989) [28] 调查了 315 个非黑素瘤皮肤癌患者中视黄醇和 $\beta$-胡萝卜素血浆浓度与 12 种个人特征和饮食因素之间的关系。

这个数据集中，响应变量是二维的，分别是血浆中视黄醇和 $\beta$-胡萝卜素的浓度。为了确保线性条件均值假设成立，选取 9 个自变量，分别是年龄 $X_1$，身高除以（体重的平方）$X_2$，每日消耗卡路里 $X_3$，每日消耗脂肪量 $X_4$，每日消耗纤维量 $X_5$，每周消耗酒精饮料量 $X_6$，每日消耗胆固醇量 $X_7$，每日食用 $\beta$-胡萝卜素量，$X_8$，每日食用视黄醇量 $X_9$。这里我们也去掉了第 62 行数据（参照 Zhu at al.(2010) [38]）。

这个数据集中，响应变量与自变量的相关系数很小，因此适合用充分降维方法。Zhu et al.(2010) 通过 BIC 准则法，得到了中心降维子空间的结构维数是 2。Ye and Weiss(2003) [35] 指出在不同的模型中，我们倾向于变异性最小的模型。借鉴他的思路，使用 Bootstrap 方法，去生成 $\hat{\mathscr{S}}^b_{Y|X}$，计算 $\hat{\mathscr{S}}^b_{Y|X}$ 与 $\hat{\mathscr{S}}_{Y|X}$ 之间的迹相关系数 $r^b, b=1,\cdots,B$。我们再取 $(1-r^b), b=1,\cdots,B$ 的中位数。这个中位数越小，则表示模型的变异性越小。我们分别使用基于回归树的方法 RF-SIR，RF-SAVE，RF-DR 与基于投影的方法 PR-SIR，PR-SAVE，PR-DR 去计算中位数。我们取 $B=1000$，抽样个数 200 个，树模型采用 python 中随机森林包（默认参数），PR-SIR 切片设置 15，PR-SAVE，PR-DR 切片设置 3。结果如下：

**表 4-19** 树模型与投影法的变异性比较

| 方法 | RF-SIR | RF-SAVE | RF-DR | PR-SIR | PR-SAVE | PR-DR |
|---|---|---|---|---|---|---|
| 中位数 | 0.10103 | 0.05347 | 0.04266 | 0.07985 | 0.11545 | 0.10580 |

由表4-19的结果，我们可以看到，整体上树模型的变异性小于投影重采样法。树模型在 SAVE 和 DR 的表现较好，投影重采样法在 SIR 的表现较好。

最后，我们来看一下 RF-DR 给出的前两个基方向。

表 4-20 RF-DR 给出的前两个基方向

| 系数 | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $X_5$ | $X_6$ | $X_7$ | $X_8$ | $X_9$ |
|---|---|---|---|---|---|---|---|---|---|
| 第 1 个 | -0.101 | 0.034 | -0.029 | 0.157 | 0.413 | 0.887 | -0.010 | -0.001 | 0.070 |
| 第 2 个 | 0.014 | 0.066 | 0.006 | -0.064 | -0.518 | 0.850 | -0.003 | 0.000 | 0.000 |

我们可以通过系数的大小来判断变量的贡献。由表4-20，$X_6$ 可能是最重要的因素，而 $X_8$ 可能是最不重要的因素。我们可以得出结论，人体中这些微量营养素的血浆浓度存在很大差异，这种差异大部分是由于饮食习惯造成，酒精摄入量或许是最重要的因素，这也间接说明过多的酒精摄入量可能与癌症发生有关。

# 第五章　结论及展望

## §5.1　结论

在大数据时代，充分降维方法的研究有重要意义。对于响应变量多维的充分降维方法，一直是个难题。我们提出了基于回归树的充分降维方法，通过大量的仿真实验，验证了我们提出方法的有效性。我们主要得到了下列几点结论。

第一，对于 SIR，SAVE，DR 方法，在响应变量多维时，使用回归树的办法，对响应变量空间进行切分的办法，都是有效的。相比于现有的多维响应变量下的充分降维方法，我们提出的基于回归树的充分降维方法在响应变量高维时往往能够取得更好的效果。

第二，在我们的模拟实验中，在自变量不超过 20 维、响应变量不超过 10 维时，对于线性模型，常见的非线性模型 (指数函数，三角函数，对数函数等)，噪声扰动，维度变化等对降维效果影响不大，均有着良好的表现。

第三，在样本量比较少时 ($n = 200$)，随机森林的效果较好。在样本数量比较大时 ($n > 1000$)，GBDT、RF、Xgboost 的表现差异不大。树模型的参数设置对于降维效果有一定影响，特别是在样本量较少时。通常来说，不应设置过大的树深度，尤其是 SAVE 方法。树深度的选取可以根据响应变量的维度、样本量的大小。这一点类似于传统切片方法中切片数量的设置。

第四，DR 方法的稳定性比 SIR 和 SAVE 方法好，这一点在我们的多个模拟实验中均得到了验证。

第五，在响应变量有缺失值时，可以采用 Xgboost 等可以直接处理缺失值的树模型，能够有效的利用其他维度的信息，效果显著好于直接删去样本。

第六，通过实例分析，我们发现，与现有的方法相比，我们提出的方法变异性不大。

## §5.2　展望

如今，机器学习和深度学习在各个领域有着非常亮眼的表现。但其理论性质一直是现在面临的一大难题。我们提出的方法在响应变量多维时，有着不错的效果，但在高维

情况下的理论性质也是一大挑战。我们在模拟时未考虑自变量出现异常值的情况，异常值会对均值的估计产生影响。对于超高维的数据，由于计算性能的限制，我们未作研究。对于样本稀疏的问题，本文也未考虑，这也值得深入研究。

# 参 考 文 献

[1] Bura E, Cook R D. Estimating the structural dimension of regressions via parametric inverse regression[J]. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 2001, 63(2): 393-410.

[2] Chen T, Guestrin C. Xgboost: A scalable tree boosting system[C]//Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. New York: Association for Computing Machinery, 2016: 785-794.

[3] Chiaromonte F, Cook R D, Li B. Sufficient dimensions reduction in regressions with categorical predictors[J]. Annals of Statistics, 2002, 30(2): 475-497.

[4] Cook R D. Principal Hessian directions revisited[J]. Journal of the American Statistical Association, 1998, 93(441): 84-94.

[5] Cook R D, Forzani L, Rothman A J. Estimating sufficient reductions of the predictors in abundant high-dimensional regressions[J]. Annals of Statistics, 2012, 40(1): 353-384.

[6] Cook R D, Lee H. Dimension reduction in binary response regression[J]. Journal of the American Statistical Association, 1999, 94(448): 1187-1200.

[7] Cook R D, Weisberg S. Sliced inverse regression for dimension reduction: Comment[J]. Journal of the American Statistical Association, 1991, 86(414): 328-332.

[8] Ferré L. Determining the dimension in sliced inverse regression and related methods[J]. Journal of the American Statistical Association, 1998, 93(441): 132-140.

[9] Friedman J H. Greedy function approximation: a gradient boosting machine[J]. Annals of Statistics, 2001, 29(5): 1189-1232.

[10] Hsing T. Nearest neighbor inverse regression[J]. Annals of Statistics, 1999, 27(2): 697-731.

[11] Hsing T, Carroll R J. An asymptotic theory for sliced inverse regression[J]. Annals of Statistics, 1992, 20(2): 1040-1061.

[12] Ke G, Meng Q, Finley T, et al. Lightgbm: A highly efficient gradient boosting decision tree[C]//Advances in neural information processing systems. Cambridge: MIT Press, 2017: 3146-3154.

[13] Li B, Wang S. On directional regression for dimension reduction[J]. Journal of the American Statistical Association, 2007, 102(479): 997-1008.

[14] Li B, Wen S, Zhu L. On a projective resampling method for dimension reduction with multivariate responses[J]. Journal of the American Statistical Association, 2008, 103(483): 1177-1186.

[15] Li B, Zha H, Chiaromonte F. Contour regression: a general approach to dimension reduction[J]. Annals of Statistics, 2005, 33(4): 1580-1616.

[16] Li K C. Sliced inverse regression for dimension reduction[J]. Journal of the American Statistical Association, 1991, 86(414): 316-327.

[17] Li K C, Aragon Y, Shedden K, et al. Dimension reduction for multivariate response data[J]. Journal of the American Statistical Association, 2003, 98(461): 99-109.

[18] Li K C, Wang J L, Chen C H. Dimension reduction for censored regression data[J]. Annals of Statistics, 1999, 27(1): 1-23.

[19] Li L, Cook R D, Nachtsheim C J. Model‐free variable selection[J]. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 2005, 67(2): 285-299.

[20] Li L, Cook R D, Tsai C L. Partial inverse regression[J]. Biometrika, 2007, 94(3): 615-625.

[21] Li L, Nachtsheim C J. Sparse sliced inverse regression[J]. Technometrics, 2006, 48(4): 503-510.

[22] Li L, Yin X. Sliced inverse regression with regularizations[J]. Biometrics, 2008, 64(1): 124-131.

[23] Liaw A, Wiener M. Classification and regression by randomForest[J]. R News, 2002, 2(3): 18-22.

[24] Lin Q, Zhao Z, Liu J S. Sparse sliced inverse regression via lasso[J]. Journal of the American Statistical Association, 2019, 114(528): 1-33.

[25] Lue H H. Sliced average variance estimation for censored data[J]. Communications in Statistics—Theory and Methods, 2008, 37(20): 3276-3286.

[26] Ma Y , Zhu L. A semiparametric approach to dimension reduction[J]. Journal of the American Statistical Association, 2012, 107(497):168-179.

[27] Ni L, Cook R D, Tsai C L. A note on shrinkage sliced inverse regression[J]. Biometrika, 2005, 92(1): 242-247.

[28] Nierenberg D W, Stukel T A, Baron J A, et al. Determinants of plasma levels of beta-carotene and retinol[J]. American Journal of Epidemiology, 1989, 130(3): 511-521.

[29] Park J H, Sriram T N, Yin X. Central mean subspace in time series[J]. Journal of Computational and Graphical Statistics, 2009, 18(3): 717-730.

[30] Park J H, Sriram T N, Yin X. Dimension reduction in time series[J]. Statistica Sinica, 2010, 20(2): 747-770.

[31] Quinlan J R. Induction of decision trees[J]. Machine Learning, 1986, 1(1): 81-106.

[32] Setodji C M, Cook R D. K-means inverse regression[J]. Technometrics, 2004, 46(4): 421-429.

[33] Shao Y, Cook R D, Weisberg S. Marginal tests with sliced average variance estimation[J]. Biometrika, 2007, 94(2): 285-296.

[34] Tan K, Shi L, Yu Z. Sparse SIR: Optimal rates and adaptive estimation[J]. Annals of Statistics, 2020, 48(1): 64-85.

[35] Ye Z, Weiss R E. Using the bootstrap to select one of a new class of dimension reduction methods[J]. Journal of the American Statistical Association, 2003, 98(464): 968-979.

[36] Yin X, Cook R D. Estimating central subspaces via inverse third moments[J]. Biometrika, 2003, 90(1): 113-125.

[37] Yu Z, Dong Y, Shao J. On marginal sliced inverse regression for ultrahigh dimensional model-free feature selection[J]. Annals of Statistics, 2016, 44(6): 2594-2623.

[38] Zhu L P , Zhu L X , Wen S Q. On dimension reduction in regressions with multivariate responses[J]. Statistica Sinica, 2010, 20(3):1291-1307.

[39] 李航. 统计学习方法 [M]. 北京: 清华大学出版社, 2012:55-74.

[40] 周志华. 机器学习 [M]. 北京: 清华大学出版社, 2016:171-196.

# 致　　谢

初入校园的场景还历历在目，转眼，三年的硕士研究生学习生涯即将结束。期末考试前的紧张备考，实习路上的匆匆人流，宿舍里的闲暇时光，这些都已留在我的记忆里。三年里，我收获了很多。在此，我要感谢很多人，感谢他们的默默付出。

首先，特别感谢我的导师於州老师。入学以来，老师便开始引导我学习的方向，让我逐步了解了统计学相关的最新热点。从毕业论文的选题到完成的各个阶段，得到了於州老师的细心指导和帮助。在职业规划方面，老师给我提供了实习机会，也给我了许多宝贵建议。

其次，感谢统计学院的张日权老师、汤银才老师、徐方军老师、方方老师、李育强老师、唐炎林老师等任课老师在专业课上给我的指导。感谢华东师范大学提供良好的学习环境。感谢同学们，室友们在学习和生活上的给予的帮助，让我度过了愉快、充实的三年时光。

最后，感谢我的家人，二十多年来对我生活上的支持，学习上的鼓励。

校园生活即将告一段落，但这不是人生的终点，我的生活即将翻开新的篇章。我将心怀梦想，脚踏实地，继续努力！

# 硕士学位论文

MASTER'S DISSERTATION