分类号_____ 密级 _____
U D C _____ 编号 10741

兰州财经大学
LANZHOU UNIVERSITY OF FINANCE AND ECONOMICS

硕 士 学 位 论 文

（专业学位）

论文题目 基于抽样的高维矩阵低秩逼近及应用研究

研 究 生 姓 名：任潇潇

指导教师姓名、职称：牛成英 教授

学 科、专 业 名 称：统计学 应用统计硕士

研 究 方 向：大数据分析

提 交 日 期：2022 年 5 月 30 日

独创性声明

本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

学位论文作者签名：任潇潇 签字日期：2022.5.30

导师签名：牛成英 签字日期：2022.5.30

关于论文使用授权的说明

本人完全了解学校关于保留、使用学位论文的各项规定，同意（选择“同意”/“不同意”）以下事项：

1.学校有权保留本论文的复印件和磁盘，允许论文被查阅和借阅，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文；

2.学校有权将本人的学位论文提交至清华大学“中国学术期刊（光盘版）电子杂志社”用于出版和编入CNKI《中国知识资源总库》或其他同类数据库，传播本学位论文的全部或部分内容。

学位论文作者签名：任潇潇 签字日期：2022.5.30

导师签名：牛成英 签字日期：2022.5.30

Low rank approximation of high dimensional matrix based on sampling and its application

Candidate : Ren Xiaoxiao

Supervisor: Niu Chengying

摘要

大数据时代，海量数据大多数以高维矩阵形式存在，如何对高维矩阵进行降维成为机器学习的研究热点问题。利用抽样技术降低高维数据的维度和计算复杂度已被证明是一种有效手段，但不同的抽样和矩阵重构方法在降维过程中产生的误差存在较大差异。本文从抽样的角度出发，研究高维矩阵低秩逼近的方法与误差测度，关注在提高低秩逼近精度的同时，能够降低计算复杂度。主要工作包括以下几方面：

首先，对于大规模数据集，Nyström 方法是一种较为有效的矩阵低秩逼近技术，旨在从原始数据矩阵中抽取部分列重构原始数据矩阵的低秩逼近矩阵。考虑到不同抽样方法对重构矩阵的精度有较大的影响，提出将不等概抽样 Nyström 方法与随机奇异值分解(SVD)方法相结合，进而在矩阵重构过程中提高矩阵低秩逼近精度，并有效降低计算复杂度。研究结果表明，提出的 Nyström 方法在矩阵重构中具有较高的精确度，且可以极大的降低计算复杂度。

其次，高维大数据矩阵分析中，使用少量主要成分逼近原始数据矩阵是常用方法，这些主要成分是矩阵行和列的线性组合，不易对数据的原始特征进行解释。提出将不等概抽样与自适应抽样结合的适用于 CUR 矩阵分解的抽样方法，并将该抽样方法与矩阵随机奇异值分解(SVD)方法相结合，对抽样得到的子矩阵 C 和 R 进行随机 SVD 分解，在控制计算复杂度的同时提高矩阵低秩逼近重构的精度。研究结果表明，基于不等概自适应抽样和随机 SVD 分解相结合的 CUR 矩阵分解方法在矩阵低秩逼近中具有较高的精确度和稳定性。

最后，将基于不等概抽样和随机 SVD 分解 Nyström 方法拓展运用于谱聚类，利用上市公司股票财务比率数据进行实证分析。提出基于不等概抽样 Nyström 特征提取方法，通过提取影响上市公司业绩的主要特征指标，在降低数据维度和数据计算复杂度的同时最大可能保留原始数据信息，并在选取特征变量的基础上对上市公司进行谱聚类分析。研究结果表明，按抽样比例为 20%对原数据指标进行特征提取，可以均匀包含原数据 10 大类一级指标，表示特征提取的结果具有较好的代表性。谱聚类结果分析可见，将选取的 73 家上市公司分为 4 类，通过聚类效果评价准则，得到表示聚类效果的值$R^2 = 0.72$，表明此次聚类具有良好的

效果。

将基于不等概抽样与随机SVD分解的CUR矩阵分解拓展运用于偏好特征提取，该偏好特征提取方法基于原始数据抽样，数据可解释性较高，意义明确。利用用户-电影评分数据进行实证检验，研究结果表明，利用 CUR 矩阵进行偏好特征提取算法性能较好，提取的用户或产品的特征能较好地反映原始数据特征；且随着抽样提取的列数和行数的增加，偏好特征提取的准确率呈上升趋势，压缩率呈下降趋势；将基于 CUR 矩阵分解的偏好特征提取方法与基于 SVD 分解的偏好特征提取方法相比，前者的准确度远远高于后者。

关键字： Nyström 方法 CUR 矩阵分解方法 不等概抽样 不等概自适应抽样 随机 SVD 分解 相对误差 计算复杂度

Abstract

In the era of big data, most of the massive data exist in the form of high-dimensional matrix. How to reduce the dimension of high-dimensional matrix has become a hot topic in machine learning. Sampling technique has been proved to be an effective method to reduce the dimension and computational complexity of high-dimensional data, but the errors generated by different sampling and matrix reconstruction methods are quite different in the process of dimensionality reduction. From the sampling point of view, this paper studies the method and error measure of low-rank approximation of high-dimensional matrix, focusing on improving the accuracy of low-rank approximation while reducing the computational complexity. The main work includes the following aspects:

Firstly, Nyström method is a relatively effective low-rank approximation technique for large-scale data sets, which aims to extract some columns from the original data matrix to reconstruct the low-rank approximation matrix of the original data matrix. Considering that different sampling methods have great influence on the accuracy of matrix reconstruction, a combination of unequal probability sampling Nyström method and stochastic singular value decomposition (SVD) method is proposed to improve the low-rank approximation accuracy and reduce the computational complexity in matrix reconstruction. The results show that the proposed Nyström method has high accuracy in matrix reconstruction and can greatly reduce the computational complexity.

Secondly, in high-dimensional big data matrix analysis, it is a common method to approximate the original data matrix with a small number of major components. These major components are linear combinations of matrix rows and columns, and it is difficult to explain the original characteristics of the data. Proposed to differ is sampling and adaptive sampling is suitable for the CUR sampling method of matrix decomposition, and the random sampling method and matrix singular value decomposition (SVD) method, combining the matrix C and R obtained by sampling randomly SVD decomposition, in the control of computational complexity and improve the accuracy of low rank approximation reconstruction. The results show that the CUR matrix decomposition method based on the combination of unequal probability adaptive sampling and stochastic SVD decomposition has high accuracy

and stability in low-rank approximation of matrices.

Finally, the Nyström method based on unequal probability sampling and random SVD decomposition is extended to spectral clustering, and empirical analysis is made by using the financial ratio data of listed companies.A feature extraction method based on the Nyström method of unequal-probability sampling is proposed. By extracting the main feature indexes that affect the performance of listed companies, the original data information can be retained as much as possible while reducing the data dimension and the complexity of data calculation. On the basis of selecting feature variables, spectral clustering analysis is carried out for listed companies. The results show that the sample ratio of 20% for feature extraction of the original data index can uniformly include 10 categories of first-level indicators of the original data, indicating that the results of feature extraction have good representativeness. The analysis of spectral clustering results shows that the 73 listed companies selected in this paper are divided into 4 categories, and the value $R^2 = 0.72$ representing the clustering effect is obtained through the evaluation criteria of clustering effect, indicating that the clustering has a good effect.

The CUR matrix decomposition based on unequal probability sampling and random SVD decomposition is extended to preference feature extraction, and empirical test is performed using user-movie rating data. The preference feature extraction method is based on raw data sampling, which has high explanatory value and clear meaning. The results show that the preference feature extraction algorithm based on CUR matrix has better performance, and the extracted user or product features can reflect the original data features well. With the increase of the number of sampling columns and rows, the accuracy rate of preference feature extraction increases and the compression rate decreases. The accuracy of preference feature extraction method based on CUR matrix decomposition is much higher than that based on SVD decomposition.

Keywords: Nyström method; CUR matrix decomposition method; Unequal probability sampling; Unequal probability adaptive sampling; Random SVD decomposition; Relative error; Computational complexity

目录

1 引言

1.1 研究背景与研究意义

1.1.1 研究背景

大数据时代，随着数据容量增加，维度上升，存储方式多样，如何高效利用高维数据、有效提取数据信息是当前机器学习待解决的难题。

在许多高维数据分析应用中，两个关键工具是降维和压缩。主成分分析(PCA)、聚类等众所周知的方法允许对数据进行极大的近似和压缩，但代价是难以准确解释降维和压缩结果。解决这一难题的一种方法是尝试利用数据的自我表达能力,而不是用一些抽象的基来表示。因为在许多应用中,数据都是可以自表达的，而且基于这种假设的方法在各种机器学习任务中取得了相当好的效果(Elhamifar,2013)。

在机器学习中，大多数高维数据是以矩阵形式存在的，矩阵的每一行和每一列都代表着原始数据的特征信息。研究者可抽取矩阵的行和列，通过矩阵重构来表示原始数据矩阵，实现维度缩减，即选择最具代表性的行或列来捕获初始矩阵的基本信息，为此，SVD分解、QR分解、CUR矩阵分解、Nyström等低秩逼近方法在子空间学习中的应用引起学者的高度关注(Mahoney,2009)。

1.1.2 研究意义

基于抽样的高维矩阵低秩逼近重构的目的在于降低数据维度，去除数据中的冗余信息和噪声，保留有效信息，以此简化高维数据矩阵分析的计算复杂度。而关于矩阵低秩逼近重构误差的测度，是矩阵低秩逼近效果的直观体现，误差越低，低秩逼近的效果越好。

抽样本身是一种降维的有效手段，通过选择合适的随机抽样方法抽取高维矩阵的部分行或部分列构建样本子空间，以此进行矩阵低秩逼近重构。一方面，降低数据维度，对数据进行有效压缩，节省存储空间；另一方面，高效提取高维数据的数据特征，以便更好地完成进一步的分类或识别学习。因此，基于抽样的矩

阵低秩逼近重构的研究及误差测度研究对高维数据降维具有重要的理论意义。

关于高维矩阵分析的理论研究和实际应用研究都在不断涌现，使得相关算法逐步完善。然而，对于信息变化多元化，数据日益复杂的大数据时代，现有高维矩阵分析方法仍有不足，有待进一步改进和完善。特别是高维数据降维时精度与计算复杂度的平衡问题，仍是学者们关注的重点之一。基于高维数据矩阵的研究现状，本文将从抽样的角度出发，研究高维矩阵低秩逼近重构方法，以及误差界的确定。通过理论分析与实证检验，为提高方法精度的同时降低计算复杂度寻找较好的解决方案，同时拓展高维矩阵低秩逼近的应用领域，为高维数据有效降维做出贡献。

1.2 国内外研究现状

1.2.1 国外研究现状

近几年,高维矩阵降维问题引起了许多学者的关注，就降维方法而言，国外学者首先提出通过特征提取与特征选择进行降维。特征提取是将高维特征投影到一个低维的子空间，以此进行高维数据表示，达到降维的目的;特征选择是从特征集中选择一个有代表性的子集来进行原始数据的表示。与特征提取相比，特征选择能够保留原始特征的数据信息，提高了模型的可解释性(Wang 等,2013;Saeys 等,2007)。

（1）子空间学习

主成分分析(PCA)、线性判别分析(LDA)等特征提取方法的提出，使子空间学习成为多数学者的关注点(Wold 等,1987;Yang 等,2009)，该方法也是高维数据进行降维处理的有效手段之一。但传统子空间学习存在对噪声和异常数据敏感、对数据流行结构提取不完整等问题。因此，有学者提出子空间学习可以通过高维矩阵分解进行特征选择，从而获得原始数据的低维表示。基于矩阵分解的子空间学习，一般可以分为两个过程:一是通过数学关系联合高维与低维空间，二是利用矩阵分解得到相关投影矩阵。Wang 等(2015)提出一种基于矩阵分解的特征选择方法，指出用特征子集张成的子空间可以很好表示由特征集张成的子空间。但基于矩阵分解，通过最大投影，最小冗余进行无监督特征选择，容易忽略数据局部的流形结构(Wang 等,2015)。Shang 等(2020)提出子空间学习无监督特征选择算

法(SLASR)，矩阵分解子空间学习框架,不仅充分考虑了数据局部信息，并且可以自适应学习流形结构，提高了子空间学习的性能。Rahmani(2015)提出一种对异常值具有稳健性的随机 PCA 算法，该算法是将给定的数据矩阵通过随机抽样和随机嵌入技术转化为压缩矩阵，再从压缩矩阵中提取低秩列子空间。Rahmani 和 Atia(2016)提出了一种针对高维低秩加稀疏的数据矩阵分解的新算法，该算法首先提出子空间学习，仅使用数据矩阵的部分行或列，有效降低了计算的复杂度。其次，该算法考虑到真实数据在低维子空间中非均匀分布的情况，在此基础上提出有效列抽样的分解算法，并进一步对该分解算法进行改进，通过有效信息抽样对一般均匀随机抽样方法优化。通过实证模拟得到，提出的算法可以以较少的样本，得到较为准确的分解结果。Loyola 等(2016)对于高维数据应用回归模型，介绍了智能抽样和增量函数学习算法(SSIF)。SSIF 可以精确逼近在高维输入空间上定义的函数，可用于以最优方法创建或重新采样训练模式的问题，可与任何回归工具(如神经网络、支持向量机和许多其他工具)组合。

（2）高维矩阵低秩近似

基于子空间学习，低秩近似作为高维矩阵降维的另一种有效方法引起大多数学者的重点关注。

在数值线性代数和计算机科学领域，有大量关于矩阵低秩近似的工作，其中大部分是受到 Johnson 和 Lindenstrauss(1984)著名结论的启发，该结论指出随机低维嵌入能够保留原始数据的欧几里得特性。随后，一系列将原始矩阵投影到低维子空间的投影算法被提出。SVD 分解被认为是可以得到最佳低秩逼近的方法，但随机投影算法和 SVD 分解算法都是基于整个矩阵进行存储和操作，而且对于数据压缩结果也不容易解释，甚至 SVD 分解比投影算法复杂度更高。

（3）基于抽样的低秩近似

为了降低计算复杂度，且能保留原始数据信息，基于抽样的矩阵低秩逼近方法被许多学者关注。该类方法的研究可以追溯到经典的理论结果，该结果表明，对于任意矩阵，存在 k 列的子集，可以作为原矩阵的投影低秩逼近，这个投影逼近误差相对矩阵的最优秩-k 近似是有界的(Kumar 等,2009b)。如 QR 分解算法，一种需要计算矩阵秩的确定算法，可以获得接近最优矩阵的投影误差；但此类基于抽样的投影近似算法需要遍历矩阵的每一项，以此保证低秩近似矩阵的良好性能，这需要大量的时间和空间，对于大规模矩阵也不可行。

基于抽样的高维矩阵低秩逼近的另一类方法，只针对矩阵的子集进行存储和操作，极大地降低了计算复杂度。比如 Nyström 低秩逼近和 CUR 矩阵分解方法。Nyström 低秩逼近只需从原始数据矩阵中抽选部分列重构矩阵，在最大限度提取原始数据信息的同时降低了计算复杂度，常用于对称半正定矩阵(SPSD)的低秩逼近，推广到任意高维矩阵的低秩逼近，CUR 矩阵分解是一种有效替代方法，既能通过抽样降低数据维度，简化计算复杂度，又可以将通过抽样得到的矩阵与联合矩阵的乘积作为对任意矩阵的近似重构，且能保持矩阵的稀疏性和非负性。

Nyström 方法最初是用来解决积分方程的问题，求解近似特征向量；Williams 和 Seeger(2001)将 Nyström 引入核机器学习，提高核矩阵分解速度和准确率；标准 Nyström 对所有样本赋予相同的权重，与核函数定义的特征方程有所偏差，密度加权 Nyström(DW-Nyström)通过引入概率密度函数作为权重，提高逼近精度(Zhang,2009)；Kumar(2009)提出集成 Nyström，能够生成比标准 Nyström 更精准的逼近；Wang(2013)将 CUR 矩阵分解中的技术引入 Nyström，提出一种改进 Nyström，提高了标准 Nyström 和集成 Nyström 的误差下降；针对逆矩阵的低秩逼近，Wang(2014)提出了一种改进 Nyström，进一步提升逼近精度。

CUR 矩阵分解已经被应用于许多领域，包括生物学、文本数据分析、医学图像分析等。对于存在缺失数据的矩阵，Xu(2015)提出 CUR+，计算简单，适用于低秩和满秩矩阵分解。Kevin(2016)给出了一种计算稀疏矩阵 CUR 分解的分布式算法，较大程度地提高了算法的运行速度；Li(2018)提出基于 CUR 矩阵分解，同时进行样本和特征选择的无监督学习方法，有效结合了特征和样本选择。Wang(2017)利用 CUR 矩阵分解进行缺失数据恢复；Sheheryar(2020)将 CUR 矩阵分解用于图片匹配。

（4）误差测度

关于误差测度，大多数学者都是基于范数进行相对误差或绝对误差测度。例如，John 等(2018)基于谱范数利用相对误差对基于概率的子空间学习进行了误差测度。Liu 和 Jing(2019)基于 F 范数利用绝对误差对提出的局部自适应矩阵低秩逼近误差进行了测度。Mahoney 等(2009)基于 F 范数利用绝对误差对 CUR 矩阵分解的误差进行了测度分析。Wang 等(2019)提出假设矩阵重构误差是一个低秩矩阵，使用核范数来测量该误差，并通过最小化核范数来学习参数。

1.2.2 国内研究现状

关于高维矩阵降维，近几年，国内学者相关研究成果也比较多。本文主要从子空间学习、矩阵低秩逼近、误差测度三方面进行梳理。

（1）子空间学习

邹珊(2015)提出一种可以从原始数据中提取信息的共享子空间数据表示模型，该模型利用高维特征信息，将原始数据有效地映射到低维空间中。李骜等(2020)提出基于低秩表示的判别特征子空间学习模型，有效解决了一般特征子空间学习不能保持样本局部结构和判别性，及样本具有噪声时，模型失效的两大问题；基于更新迭代，可以将基于低秩稀疏表示的子空间学习概括为基于矩阵分解和谱聚类的子空间学习(武继刚等,2021)。贺文琪等(2021)提出基于核嵌入的子空间学习算法，解决了一般降维技术无法对高维非线性数据的内部结构进行深入挖掘的问题。除了子空间学习相关算法的研究之外，更多的是应用型研究。詹永杰(2015)研究了基于独立子空间学习的表情识别问题；耿妍(2016)结合 PCA 与子空间学习对高维数据进行了分类处理；蔡雨红等(2021)提出类内低秩子空间学习，并将其应用于人脸识别；许楠(2019)将子空间学习应用到多视角聚类中。

（2）矩阵低秩近似

基于子空间学习的高维矩阵低秩近似方面的研究，刘松华等(2011)针对传统核矩阵分解，列与类别独立的假设，提出核矩阵低秩近似分解方法，避免对整个核矩阵分解运算，降低了计算复杂度；唐文俊等(2012)提出基于密度的聚类 Nyström 方法，解决了一般 Nyström 随机抽样准确性不足的问题；曾琦等(2014)指出 SVD 分解因为线性综合了全局信息，而使生成的数据稠密，难以解释，而 CUR 分解得到的矩阵稀疏，物理意义明确。雷恒鑫等(2017)基于行列联合选择，利用 CUR 矩阵分解低秩近似，分别提取用户、产品偏好特征，并通过实证得到该方法准确度和可解释性都更优良。杨美姣等(2018)将 Nyström 低秩逼近方法用于偏好特征提取，进行电影推荐。

（3）误差测度

关于矩阵低秩逼近误差测度，赵知劲等(2014)使用 KL 散度度量增量非负矩阵分解效果，分析得出使用 KL 散度节省了存储空间，且矩阵分离性能明显优于基于欧氏距离的矩阵分解效果，但该方法的计算复杂度较高；雷恒鑫等(2017)使

用误差率和压缩率测量基于行列联合选择矩阵分解的精度，压缩率指原始数据矩阵中的元素总数与低维近似矩阵总的元素数之比。段菲等(2020)指出，传统测量重构矩阵逼近效果的最小二乘误差平方函数法，需假定噪声服从零均值的高斯分布，实际应用中较难满足。因此，该作者提出，对于标准的非负矩阵分解，以样本整体的重构误差为度量单元，利用 F 范数的平方度量矩阵重构误差。此外,还有部分文献中不直接测度矩阵分解误差,而是从保留数据全局重构信息的角度出发，通过最小化子空间学习残差矩阵进行矩阵重构误差测度。

综上所述，关于高维矩阵的低秩逼近，抽样技术仍是基础又成熟的方法，基于抽样的高维矩阵低秩逼近方法研究仍是目前研究的热点问题。其中，Nyström 低秩逼近是 SPSD 矩阵低秩逼近的代表方法，推广到任意高维矩阵，CUR 矩阵分解是有效替代 Nyström 方法的低秩逼近技术。此外，在基于抽样的高维矩阵低秩逼近研究中，抽样方法与交叉矩阵低秩逼近方法的选择对整体矩阵低秩逼近的精度和计算复杂度具有重要影响，也是目前研究重要的关注点。

误差测度是矩阵低秩逼近精度测量的直观体现，分析总结相关研究文献可见，关于抽样的矩阵低秩逼近误差测度研究尽管已经取得了一些成就，但还处于探索阶段，研究文献相对匮乏，有限的相关研究文献中，常用的方法是通过矩阵范数方法来测量矩阵分解误差。而不同的抽样方法和矩阵低秩逼近方法对于误差测度具有重要影响，如何精准测度基于行列抽样的矩阵低秩逼近子空间学习过程中的误差有待进一步研究。

1.3 主要研究内容和重点解决的问题

1.3.1 主要研究内容

本文的研究内容主要包括三部分，分别是：

（1）基于行列抽样的矩阵低秩逼近研究

该部分主要研究如何基于抽样方法来进行矩阵低秩逼近，提高精度，同时降低学习计算和时间复杂度。讨论针对对称半正定矩阵的低秩逼近方法，Nyström 方法，以及任意矩阵的低秩逼近方法，CUR矩阵分解，并对两种方法进行改进；尽管已有文献已讨论了一些有效方法，但矩阵低秩逼近中主要问题是确定低秩 k，以及降低误差，提升计算效率。此外，这两种方法需要随机抽取样本列或行对大

规模的矩阵做精确近似逼近，选取样本列和样本行的抽样方法与抽样比例的不同，以及子矩阵中样本列与原矩阵对应行交叉点数据构成的交叉矩阵、联合子矩阵与原矩阵的联合矩阵的低秩逼近方法不同，重构矩阵的精度与计算复杂度也各有差异。本文拟讨论几种随机抽样方法，关注不同抽样方法下、交叉矩阵不同低秩分解方法下，抽样误差与时间复杂性以及在概率保证下的矩阵低秩分解的精度。

（2）误差界与计算复杂度的确定

从理论层面出发，讨论本文提出的矩阵低秩逼近方法与一般最佳秩逼近方法之间的误差大小，以及计算复杂度的比较，确定新提出方法的误差界和计算复杂度。

（3）数值模拟与实证检验

为检验前述理论研究结果，还需通过数值模拟和实证检验进行验证分析。数值模拟部分通过随机生成满足条件的高维数据对抽样方法及其误差与时间复杂度进行测度研究；实证分析部分首先将改进的Nyström低秩逼近方法拓展应用于谱聚类，并与一般的k-means聚类就聚类精度进行比较；其次将改进后的CUR矩阵分解拓展应用于偏好特征提取，并与一般的SVD分解偏好特征提取就准确率进行比较。

1.3.2 重点解决的问题

研究不同抽样方法抽取原始高维矩阵行列构建子矩阵、不同低秩逼近方法近似逼近交叉矩阵，及其对整体高维矩阵低秩逼近精度和计算复杂度的影响是本文研究的主要内容，也是关键部分。基于矩阵行列抽样的低秩逼近误差测度的实现可优化行列抽样方法，提高高维矩阵低秩逼近精度和降低机器学习的计算复杂度。

综合以上研究内容，本文研究思路与技术路线如图 1.1 所示。

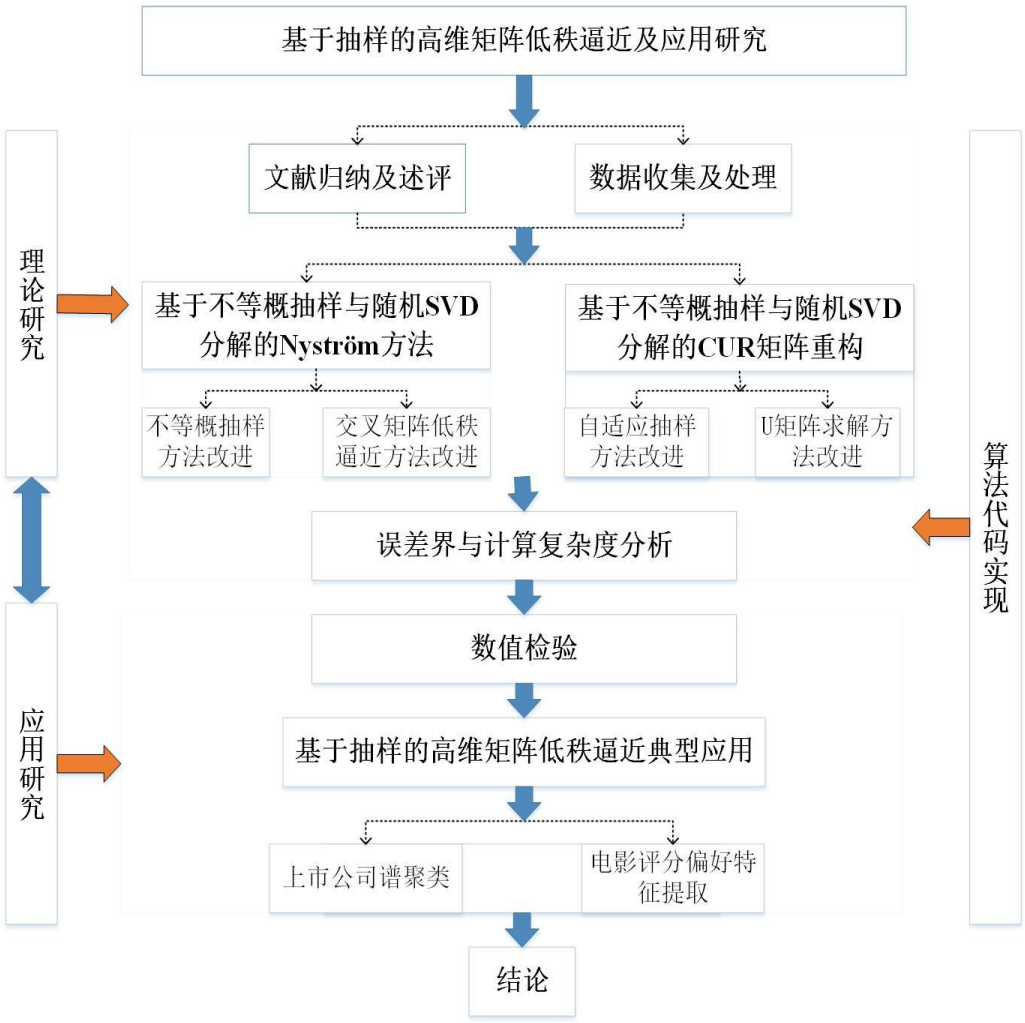


图 1.1 研究思路与技术路线

1.4 论文组织结构

本文主要从抽样的角度出发，研究基于抽样的高维矩阵低秩逼近方法以及误差界的确定，通过对抽样方法和低秩逼近方法进行改进，提高矩阵低秩逼近精度，降低计算复杂度，最后通过模拟数据进行实证检验，再将改进后的方法运用到解决实际问题中。主要安排如下：

第 1 节为引言，主要概括了高维矩阵低秩逼近的研究背景和研究意义，并介绍了近几年国内外学者对高维数据降维和压缩的研究现状，提出本文的研究内容和组织框架，以及本文的创新点。

第 2 节介绍了基于不等概抽样与随机 SVD 分解的 Nyström 方法，对已有 Nyström 矩阵低秩逼近方法进行改进，提升矩阵低秩逼近精度，降低计算复杂度，

进一步提出改进后方法的误差界，并利用模拟数据进行了实证检验。

第 3 节将 SPSD 矩阵的低秩逼近方法进行推广，介绍了基于不等概抽样和随机 SVD 分解的 CUR 矩阵分解重构方法，对现有 CUR 矩阵分解方法进行改进后，提出新的误差界，并进行了实证检验。

第 4 节将第 2 节提出的方法拓展应用，将基于不等概抽样与随机 SVD 分解的 Nyström 方法利用上市公司股票财务指标数据运用于谱聚类，并与 k-means 聚类进行聚类效果比较。

第 5 节将基于不等概抽样和随机 SVD 分解的 CUR 矩阵分解运用于偏好特征提取，利用用户电影评分数据进行实证分析，进一步与 SVD 分解偏好特征提取方法就准确率进行比较。

第 6 节对本文的研究内容进行了总结，并根据现有方法的研究不足提出了进一步的研究方向。

1.5 创新点

（1）基于低维不等概抽样思想，构造新的不等概抽样 Nyström 方法，即根据矩阵每一列不同的入样概率随机抽取部分列构建样本子集，在保证样本列选取随机性的同时，最大限度提取原始数据信息。在基于样本列的矩阵重构过程中，利用随机 SVD 分解对由样本列与对应行交叉点数据构成的矩阵进行低秩逼近，使重构后的矩阵与原矩阵之间的误差更小。

（2）在传统自适应抽样方法的基础上，提出不等概自适应抽样法。将不等概抽样的思想与自适应抽样结合，充分保留原始数据信息的同时，使样本列和样本行更具代表性；降低抽样误差的同时，提高抽样效率。其次，通过随机 SVD 分解对子矩阵$\boldsymbol{C}$和$\boldsymbol{R}$进行低秩逼近，简化时间和空间存储复杂度；再利用随机 SVD 分解低秩逼近后得到的矩阵$\boldsymbol{D}_C$和$\boldsymbol{D}_R$，联合$\boldsymbol{C}$、$\boldsymbol{R}$和原矩阵近似逼近矩阵$\boldsymbol{U}$，进一步低秩重构原始高维矩阵，提高整体 CUR 矩阵分解逼近重构的精度，且具有明确的误差界。

2 基于不等概抽样与随机 SVD 分解的 Nyström 方法

信息技术的发展为海量数据的收集和存储提供了技术支撑，进而，如何有效处理海量数据便成了当前机器学习关注的热点问题。这也给大规模数据矩阵的机器学习中，尤其是传统的算法，如支持向量机(Joachims,1998)、核主成分分析(Shi,2009)、流形学习(Chao,2013)以及聚类等方法的计算和存储带来了新的挑战。

对于大规模数据矩阵，目前有效的解决方法之一是对原矩阵进行降维处理，常用的方法包括矩阵奇异值分解(SVD)、正交三角分解(QR)等低秩逼近方法，但此类方法通常要求矩阵具有稀疏性或非负性。且传统的矩阵分解方法对线性可扩展数据是有效的，但对非线性数据便无能为力。二是基于抽样方法进行降维处理，即通过构建抽样子空间进行低秩逼近，Nyström 低秩逼近方法、CUR 矩阵分解逼近等。相比 SVD 或 QR 矩阵分解方法或 CUR 矩阵分解，Nyström 低秩逼近方法只需从原始数据矩阵中抽选部分列重构矩阵，在最大限度提取原始数据信息的同时降低了计算复杂度。其中 Nyström 低秩逼近方法常用于对称半正定矩阵(SPSD)的低秩逼近，也广泛应用于加速核算法、流形学习、图像分割以及矩阵填充等方面。

而大规模矩阵进行低秩逼近的有效方法之一的 Nyström 方法，其关键点是对矩阵列的抽取，样本列的抽样方法又对低秩逼近精度具有重要影响（Williams,2000; Fine,2001）。在使用 Nyström 方法对高维数据分析过程中，均匀抽样首先引起了学者的关注。Williams 等(2001)首先将均匀不放回抽样与 Nyström 方法融合应用到机器学习中，加速核矩阵特征分解。Kumar 等(2009)对均匀抽样和非均匀抽样下数据核矩阵的低秩逼近进行了实例分析，并对其性能做了对比分析，结果表明，均匀抽样在时间、空间和近似精确度方面都优于非均匀抽样。李毅等(2015)将均匀抽样与大数据挖掘相结合，通过误差率比较五折交叉、随机抽样和均匀抽样在数据挖掘算法中的表现，并指出均匀抽样的误差率低于其他两种抽样方法。Kumar 等(2012)认为矩阵每一列的信息随着抽样过程在不断变化，于是提出自适应抽样方法，并将其与 Nyström 方法结合，提出自适应抽样 Nyström 方法。贾洪杰等(2017)提出基于概率的增量抽样方法，通过在每次迭代中不断更新样本的抽样概率选择新的样本点，进而有效降低抽样误差。随后，这类增量抽

样方法也广泛应用于数据挖掘、谱聚类、动态优化、路径规划算法等方面。

传统的均匀抽样是在抽样和估计中都使用相等权重的抽样方法(晏振等，2016)，且每个样本的入样概率都相等。基于数据集具有相同重要性信息的思想，不需要遍历整个数据集，因此，它的采样效率一般较高，也是 Nyström 方法中较为常用的抽样方法。但对于大规模数据集，均匀抽样的样本点易集中于一个区域，且数据集重要性信息变化多样时，均匀抽样会导致样本代表性较差，而损失掉重要信息。此时自适应抽样比均匀抽样更加准确，但计算效率较低。概率增量抽样与自适应抽样法类似，也是一种动态抽样方法，相比其他抽样方法更加准确的同时，也会因为多次更新检索整个数据集，以及动态更新每一列的入样概率而使计算效率降低。且由于该方法每次抽取的样本是根据每步迭代更新的入样概率大小，便无法保证样本的随机性，使样本代表性大打折扣。

Nyström 方法需要随机抽取样本列对大规模的矩阵做精确近似逼近，选取样本列的抽样方法与抽样比例的不同，以及原矩阵中样本列与对应行交叉点数据构成的交叉矩阵的低秩逼近方法不同，都会导致重构矩阵的精度与计算复杂度存在差异。基于此问题，Kumar(2010)提出集成 Nyström 方法，该方法使用块对角矩阵逼近交叉矩阵的逆矩阵，除非交叉矩阵接近于块对角矩阵，否则这样的矩阵逼近效果特别差。Halko 等(2009)提出了一种简单但精确度较高的随机算法，即随机 SVD 分解算法，用于构造低秩近似矩阵。Mu Li(2010)将 Nyström 方法与随机 SVD 算法结合，对交叉矩阵进行随机 SVD 分解，提高低秩逼近精度。

本节基于低维不等概抽样思想，构造新的不等概抽样 Nyström 方法，即根据矩阵每一列不同的入样概率随机抽取部分列构建样本子集，在保证样本列选取随机性的同时，最大限度提取原始数据信息。与标准 SVD 分解相比，随机 SVD 分解更适用于大型数据，可以避免因计算量大而无法收敛的问题，也可以更快速求解奇异向量，降低矩阵低秩逼近误差。在基于样本列的矩阵重构过程中，利用随机 SVD 分解对由样本列与对应行交叉点数据构成的矩阵进行低秩逼近，可使重构后的矩阵与原矩阵之间的误差更小。

2.1 理论基础

2.1.1Nyström 方法

考虑对一个对称半正定矩阵$\boldsymbol{K} \in \mathbb{R}^{n\times n}$，基于从矩阵本身随机抽选的$m << n$的样本列生成一个$\boldsymbol{K}$的低秩逼近矩阵$\tilde{\boldsymbol{K}}$。假设样本列已选取，$\boldsymbol{C}$表示由这些样本列构成的$n \times m$矩阵，$\boldsymbol{W}$表示矩阵$\boldsymbol{K}$中由$m$列样本列与相对应的$m$行相交组成的$m \times m$矩阵。矩阵$\boldsymbol{K}$的行和列抽样后重新排列，使$\boldsymbol{K}$和$\boldsymbol{C}$可以写成下面的形式(Kumar,2012)：

$$\boldsymbol{K} = \begin{bmatrix} \boldsymbol{W} & \boldsymbol{K}_{21}^{T} \\ \boldsymbol{K}_{21} & \boldsymbol{K}_{22} \end{bmatrix},\ \boldsymbol{C} = \begin{bmatrix} \boldsymbol{W} \\ \boldsymbol{K}_{21} \end{bmatrix} \tag{2.1}$$

因为$\boldsymbol{K}$是对称半正定矩阵，所以$\boldsymbol{W}$也是对称半正定矩阵。$\boldsymbol{K}_{21}$的元素表示剩余点与样本点的相似矩阵，$\boldsymbol{K}_{22}$的元素表示所有剩余点的相似矩阵。当$m << n$时，$\boldsymbol{K}_{22}$一般比较大。Nystrom 方法是使用式(2.1)中的$\boldsymbol{C}$和$\boldsymbol{W}$来逼近$\boldsymbol{K}$，即：

$$\tilde{\boldsymbol{K}} = \boldsymbol{C}\boldsymbol{W}^{-1}\boldsymbol{C}^{\boldsymbol{T}} \tag{2.2}$$

其中，$\boldsymbol{W}^{-1}$表示矩阵$\boldsymbol{W}$的逆矩阵。

可以证明，随着抽取的样本列数m增加，$\tilde{\boldsymbol{K}}$收敛于$\boldsymbol{K}$(贾洪杰,2017)。

假设对称半正定矩阵$\boldsymbol{K}$可以分解为$\boldsymbol{K} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{\boldsymbol{T}}$，$\boldsymbol{\Lambda}$是对角矩阵，对角元素为矩阵$\boldsymbol{K}$的特征值，这些特征值对应的特征向量为$\boldsymbol{U}$的列。Nystrom 方法是对$\boldsymbol{W}$进行特征分解得到$\boldsymbol{U}_W$和$\boldsymbol{\Lambda}_W$。假设使用均匀不放回抽取样本列，将$\boldsymbol{W}$的 SVD 分解写为$\boldsymbol{W} = \boldsymbol{U}_W\boldsymbol{\Lambda}_W\boldsymbol{U}_W^T$，由式(2.2)可得，

$$\begin{aligned} \tilde{\boldsymbol{K}} &= \boldsymbol{C}\boldsymbol{W}^{-1}\boldsymbol{C}^{\boldsymbol{T}} \\ &= \boldsymbol{C}\boldsymbol{U}_W\boldsymbol{\Lambda}_W^{-1}\boldsymbol{U}_W^T\boldsymbol{C}^T \\ &= \left(\sqrt{\frac{n}{m}}\boldsymbol{C}\boldsymbol{U}_W\boldsymbol{\Lambda}_W^{-1}\right)\left(\frac{n}{m}\boldsymbol{\Lambda}_W\right)\left(\sqrt{\frac{n}{m}}\boldsymbol{C}\boldsymbol{U}_W\boldsymbol{\Lambda}_W^{-1}\right)^T \end{aligned}$$

所以，Nystrom 方法对矩阵$\boldsymbol{K}$生成的近似特征值和特征向量为

$$\tilde{\boldsymbol{\Lambda}} = \left(\tfrac{n}{m}\right)\boldsymbol{\Lambda}_W$$

$$\tilde{\boldsymbol{U}} = \sqrt{\tfrac{m}{n}}\boldsymbol{C}\boldsymbol{U}_W\boldsymbol{\Lambda}_W^{-1}$$

对于特别大的数据集，分解较小的交叉矩阵$\boldsymbol{W}$的计算复杂度也较高。因此，

对矩阵$\boldsymbol{W}$进行低秩逼近是降低计算复杂度的有效手段。而标准 Nyström 方法就是对矩阵$\boldsymbol{K}$生成秩为$k(k \leqslant n)$的低秩逼近，记为：

$$\tilde{\boldsymbol{K}} = \boldsymbol{C}\boldsymbol{W}_{\boldsymbol{k}}^{+}\boldsymbol{C}^{\boldsymbol{T}} \tag{2.3}$$

其中，$\boldsymbol{W}_k$是$\boldsymbol{W}$关于F范数或谱范数的秩为k最佳近似，可由$\boldsymbol{W}$的奇异值分解(SVD)得到，$\boldsymbol{W}_k^{+}$表示矩阵$\boldsymbol{W}_k$的伪逆矩阵。Nyström 方法对矩阵$\boldsymbol{K}$生成的前k个奇异值($\boldsymbol{\Sigma}_k$)和奇异向量($\boldsymbol{U}_k$)记为：

$$\tilde{\boldsymbol{\Sigma}}_k = \frac{n}{m}\boldsymbol{\Sigma}_{W,k},\ \tilde{\boldsymbol{U}}_k = \sqrt{\frac{m}{n}}\boldsymbol{C}\boldsymbol{U}_{W,k}\boldsymbol{\Sigma}_{W,k}^{+}$$

对矩阵$\boldsymbol{W}$进行 SVD 分解的时间复杂度为$O(m^3)$，其他矩阵相乘运算时间复杂度为$O(kmn)$，所以，标准 Nyström 方法总的时间复杂度为$O(m^3 + kmn)$。因为$m << n$，$k \leqslant m$，所以，该方法的复杂度低于直接对$\boldsymbol{K}$作 SVD 分解的复杂度$O(n^3)$。

2.1.2 随机 SVD 分解

在对高维矩阵做降维低秩处理时，SVD 分解一般是较佳选择。但对于大规模矩阵，精确的 SVD 分解会在计算上出现较高的复杂性。为了解决这一问题，Halko 等(2009)提出的随机 SVD 分解算法构造低秩近似矩阵，即通过一组正交基将大矩阵转换为小矩阵，通过对小矩阵的分解，近似得到大矩阵的奇异值分解。该算法将矩阵低秩近似分为两个部分：一是构造一个低维子空间捕捉矩阵的信息；二是将矩阵限制在子空间中，并对限制后的子矩阵进行矩阵分解。主要包括以下两步：

阶段一：对输入矩阵$\boldsymbol{A}$构造一个近似矩阵，可以理解为寻找一个矩阵$\boldsymbol{Q}$，要求$\boldsymbol{Q}$具有正交列，满足$\boldsymbol{A} \approx \boldsymbol{Q}\boldsymbol{Q}^{*}\boldsymbol{A}$。希望$\boldsymbol{Q}$的列尽可能的少，但能包含输入矩阵的精确近似。

阶段二：利用矩阵$\boldsymbol{Q}$，对$\boldsymbol{A}$进行标准分解(SVD、QR 等)。

以大型矩阵$\boldsymbol{A}$进行 SVD 逼近为例。两阶段随机法为 SVD 分解提供了一种更精确的计算方法，但由于输入矩阵 SVD 分解缓慢，该方法的简单计算在应用中往往不够。为此加入了过采样参数p和幂迭代参数q。而在实践中，$p = 5$或$p = 10$；

$q=1$或$q=2$时，一般会产生较优的结果。对于一些类型的随机抽样方案，当样本量较小时，SVD 低秩分解会失效，但随机 SVD 分解方法的失效概率会随过采样参数p的增加呈超指数下降。

两阶段随机 SVD 分解过程具体见算法 2.1(Halko,2010)。

算法 2.1 随机 SVD 分解

输入：给定一个$m\times n$的矩阵$\boldsymbol{A}$，奇异向量目标个数k，超参数p（$p=5$或$p=10$），幂迭代参数q（$q=1$或$q=2$），

输出：一个近似秩为$2k$的分解$\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{*}$，$\boldsymbol{U}$和$\boldsymbol{V}$是正交矩阵，$\boldsymbol{\Sigma}$是非负的对角阵。

stage1:

step1: 构造一个$n\times 2k$的 Gaussian 测试矩阵$\boldsymbol{\Omega}$。

step2: 计算矩阵$\boldsymbol{Y}=(\boldsymbol{A}\boldsymbol{A}^{*})^{q}\boldsymbol{A}\boldsymbol{\Omega}$。

step3: 构造一个矩阵$\boldsymbol{Q}$，它的列构成$\boldsymbol{Y}$值域的一组标准正交基。

stage 2:

step4: 构造$\boldsymbol{B}=\boldsymbol{Q}^{*}\boldsymbol{A}$，得到$\boldsymbol{A}$的低秩因子分解$\boldsymbol{A}\approx\boldsymbol{Q}\boldsymbol{B}$。

step5: 对低秩矩阵$\boldsymbol{B}$进行 SVD 分解：$\boldsymbol{B}=\tilde{\boldsymbol{U}}\boldsymbol{\Sigma}\boldsymbol{V}^{*}$

step6: 令$\boldsymbol{U}=\boldsymbol{Q}\tilde{\boldsymbol{U}}$

step7: 计算矩阵$\boldsymbol{A}$的近似奇异值分解。

随机 SVD 分解在对输入矩阵低秩逼近过程中至少需要遍历一次输入矩阵的列，因此计算复杂度稍高于 SVD 分解，但其基本原理简单，实用性更强，精度更高。

2.2 基于不等概率抽样与随机 SVD 分解的低秩逼近的矩阵重构

2.2.1 不等概矩阵列抽样

从高维大数据点构成的矩阵中随机抽取的列子集可以作为独立的样本(贾洪杰,2017)，考虑到数据点信息差异性，入样概率可以从矩阵的列范数与矩阵F范数决定的概率分布中得到。由于矩阵列范数不尽相同，利用不等概率抽样方法抽取样本列，在保证抽样的随机性的同时，能充分体现数据列反应信息的差异。对于矩阵$\boldsymbol{K}$，可以一次性从中选择m列构造矩阵$\boldsymbol{C}$，为体现样本列之间数据信息的差异性，第i列的抽样概率可定义为：

$$P_i = \frac{||\boldsymbol{K}^{(i)}||_2^2}{||\boldsymbol{K}||_F^2} \tag{2.4}$$

其中，$\boldsymbol{K}^{(i)}$表示矩阵$\boldsymbol{K}$的第i列，$||.||_2$表示第i列向量的2−范数，$||.||_F$表示矩阵的F范数。

基于抽样产生的大小为m的样本列，与相应矩阵$\boldsymbol{K}$的m行交点构成交叉矩阵$\boldsymbol{W} \in \mathbb{R}^{m\times m}$。

2.2.2 交叉矩阵的随机 SVD 分解

通过随机 SVD 方法对$\boldsymbol{W}$进行低秩逼近处理，主要包括两步：一是构造一个能捕捉到矩阵$\boldsymbol{W}$“行为”的随机矩阵$\boldsymbol{\Omega}$；二是对$\boldsymbol{W}$进行限制，并利用约化矩阵进行 SVD 分解。具体的计算见算法 2.2(Halko,2010)。

算法 2.2 交叉矩阵随机 SVD 分解

输入： 对称矩阵$\boldsymbol{W} \in \mathbb{R}^{m\times m}$，秩$r$,过采样参数$p$,幂参数$q$。

输出： 特征向量矩阵$\boldsymbol{U}_w$，$\boldsymbol{W}$的低秩近似矩阵$\boldsymbol{W}_k$。

step1： 构造一个$m\times(r+p)$的标准 Gaussian 随机矩阵$\boldsymbol{\Omega}$；

step2： 计算矩阵$\boldsymbol{Z} = \boldsymbol{W}\boldsymbol{\Omega}$和$\boldsymbol{Y} = \boldsymbol{W}^{q-1}\boldsymbol{Z}$；

step3： 通过 QR 分解得到一个正交矩阵$\boldsymbol{Q}$，使$\boldsymbol{Y} = \boldsymbol{Q}\boldsymbol{Q}^T\boldsymbol{Y}$；

step4： 计算约化矩阵$\boldsymbol{F}$,$\boldsymbol{F} = \boldsymbol{Q}^T\boldsymbol{W}\boldsymbol{Q}$；

step5： 对$\boldsymbol{F}$进行 SVD 分解得到$\boldsymbol{F} = \boldsymbol{U}_F\boldsymbol{\Lambda}_F\boldsymbol{U}_F^T$；

step6： 计算矩阵$\boldsymbol{U}_W = \boldsymbol{Q}\boldsymbol{U}_F$；

step7： 求$\boldsymbol{W}$的低秩近似$\boldsymbol{W}_k \backsimeq \boldsymbol{U}_W\boldsymbol{\Lambda}_F\boldsymbol{U}_W^T \backsimeq \boldsymbol{Q}\boldsymbol{F}\boldsymbol{Q}^T = (\boldsymbol{Q}\boldsymbol{V}_F)\boldsymbol{\Lambda}_F(\boldsymbol{Q}\boldsymbol{V}_F)^T$ (2.5)

算法 2.2 计算矩阵$\boldsymbol{Z}$和$\boldsymbol{Y}$的时间复杂度为$O(m^2k)$，QR 分解的复杂度为$O(mk)$，计算矩阵$\boldsymbol{F}$的复杂度为$O(mk^2)$，SVD 分解的复杂度为$O(k^3)$。所以，该算法总的时间复杂度为$O(m^2k+k^3)$，为m的二次型。

2.2.3 低秩逼近的矩阵重构

根据 Nyström 方法，利用随机 SVD 分解方法对交叉矩阵进行低秩为k的逼近后，需要对低秩逼近矩阵$\boldsymbol{W}_k$求解伪逆矩阵，可以得到矩阵$\boldsymbol{W}_k^+$，从而可以根据$\tilde{\boldsymbol{K}} = \boldsymbol{C}\boldsymbol{W}_{\boldsymbol{k}}^+\boldsymbol{C}^{\boldsymbol{T}}$得到原数据矩阵$\boldsymbol{K}$的低秩逼近重构矩阵。

根据矩阵每一列的不同入样概率进行抽样，可以保证抽选样本列的代表性，降低样本不均衡造成的抽样误差。为保证抽样的随机性，且为避免抽取到重复列，本文主要讨论不放回不等概抽样 Nyström 方法。具体的抽样重构过程如算法 2.3 所示。

算法 2.3 不等概抽样 Nyström 算法

输入： $n \times n$ SPSD 矩阵（$\boldsymbol{K}$），要抽取的样本列数(m)。
输出： 子矩阵$\boldsymbol{C}$、重构矩阵$\tilde{\boldsymbol{K}}$以及矩阵低秩逼近误差ϵ。
Step1：计算列抽样概率$P_i = \dfrac{||K^{(i)}||_2^2}{||K||_F^2}$；
Step2：根据P_i选择样本列，构造样本子矩阵$\boldsymbol{C}$；
Step3：构造交叉矩阵$\boldsymbol{W}$；
Step4：对$\boldsymbol{W}$进行随机 SVD 分解，求低秩近似矩阵$\boldsymbol{W}_k$；
Step5：计算$\boldsymbol{W}_k^+$；
Step6：计算重构矩阵$\tilde{\boldsymbol{K}} = \boldsymbol{C}\boldsymbol{W}_{\boldsymbol{k}}^+\boldsymbol{C}^{\boldsymbol{T}}$；
Step7：计算矩阵重构相对误差$\epsilon = \frac{||\boldsymbol{K}-\tilde{\boldsymbol{K}}||_F}{||\boldsymbol{K}||_F}$或绝对误差$\epsilon = ||\boldsymbol{K} - \tilde{\boldsymbol{K}}||_F$。

2.3 误差分析

为了研究本文提出的低秩逼近方法的误差，首先介绍相关定义与引理。

对矩阵$\boldsymbol{K}$进行 SVD 分解可以写为$\boldsymbol{K} = \sum_{i=1}^{r} \sigma_i \boldsymbol{\mu}_i \boldsymbol{\nu}_i^T$，$r$为矩阵的秩，$\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_r \geqslant 0$称为奇异值，$\mu_i$和$\nu_i$构成矩阵列向量的正交基，分别表示左奇异向量和右奇异向量。对于$1 \leqslant i \leqslant r$，$\mu_i^T \boldsymbol{K} = \sigma_i \boldsymbol{\nu}_i^T$和$\boldsymbol{K}\boldsymbol{\nu}_{(i)} = \sigma_i \boldsymbol{\mu}_i$。

$||\boldsymbol{K}||_F^2 = \sum_{i=1}^{m}\sum_{j=1}^{n} k_{ij}^2$。

根据 Eckart 提出的关于矩阵低秩逼近的定理(Golub,1987)，对秩为k的所有矩阵$\hat{\boldsymbol{K}}_k$，求使$||\boldsymbol{K} - \hat{\boldsymbol{K}}_k||_F$最小化的最优解，得到矩阵$\boldsymbol{K}_k$，即为矩阵$\boldsymbol{K}$的最佳秩$k$逼近。$\boldsymbol{K}_k = \sum_{i=1}^{k} \boldsymbol{K}\boldsymbol{\mu}^{(i)}\boldsymbol{\nu}^{(i)T}$，可以得到，

$$||\boldsymbol{K}_k||_F^2 = \sum_{i=1}^{k} \sigma_i^2,\ ||\boldsymbol{K} - \boldsymbol{K}_k||_F^2 = \sum_{i=k+1}^{r} \sigma_i^2 \tag{2.6}$$

定义$\boldsymbol{K} \in \mathbb{R}^{n\times n}$为一个列采样矩阵，如果矩阵$\boldsymbol{K}$的第$i$列在第$j$次抽样中被抽中，则$\boldsymbol{S}_{ij}=1$，否则，$\boldsymbol{S}_{ij}=0$。$\boldsymbol{C}=\boldsymbol{KSD}$，包含了由$\boldsymbol{D}$标准化后的$\boldsymbol{K}$的样本列，$\boldsymbol{D}$表示进行缩放的对角矩阵。$\boldsymbol{K}$是 SPSD 矩阵，可以写为$\boldsymbol{K}=\boldsymbol{X}^T\boldsymbol{X}$，$\boldsymbol{H}=\boldsymbol{XSD}$包含了由$\boldsymbol{D}$标准化后$\boldsymbol{X}$的样本列。进一步可以得到$\boldsymbol{C}=\boldsymbol{X}^T\boldsymbol{H}$，$\boldsymbol{W}=\boldsymbol{H}^T\boldsymbol{H}$。

为证明本文提出的定理，现引入引理 2.1。

引理 2.1(Li,2014)：在算法 2.2 中，根据矩阵$\boldsymbol{Q}$生成的低秩逼近矩阵关于F范数的误差期望为：

$$E||\boldsymbol{W}-\boldsymbol{QQ}^T\boldsymbol{W}||_F \leqslant (1+k/p-1)^{1/2}\left(\sum_{i>k}\sigma_i(\boldsymbol{W})\right)^{1/2} \tag{2.7}$$

其中，$\sigma_i(\boldsymbol{W})$表示$\boldsymbol{W}$的奇异值，$k \leqslant i \leqslant m$。

定理 2.1：假设矩阵$\boldsymbol{K} \in \mathbb{R}^{n\times n}$，为 SPSD 矩阵，从$\{1,\cdots,n\}$列中以抽样概率为$P_i=\frac{||\boldsymbol{K}^{(i)}||_2^2}{||\boldsymbol{K}||_F}$随机无放回地选择$m$（$1 \leqslant m \leqslant n$）构成子矩阵$\boldsymbol{C}$，令$\boldsymbol{CC^T}$为$\boldsymbol{KK^T}$的近似逼近，则，

$$E||\boldsymbol{KK}^T-\boldsymbol{CC}^T||_F \leqslant \sqrt{\tfrac{1}{m}}||\boldsymbol{K}||_F \tag{2.8}$$

定理 2.1 的证明由文献（Drineas,2006）中定理 2 的证明可得。

定理 2.2：假设矩阵$\boldsymbol{K} \in \mathbb{R}^{n\times n}$，为 SPSD 矩阵，利用本文提出的基于不等概抽样和随机 SVD 分解的 Nyström 方法，对高维 SPSD 矩阵$\boldsymbol{K}$进行低秩近似重构，重构后的矩阵关于F范数的低秩逼近误差存在以下不等式：

$$E(||\boldsymbol{K}-\tilde{\boldsymbol{K}}||_F) \leqslant \xi||\boldsymbol{K}-\boldsymbol{K}_k||_F+(n+2\xi\sqrt{\frac{1}{m}})\boldsymbol{K}_{ii}^* \tag{2.9}$$

其中，$\xi=\frac{2(k+p)}{\sqrt{p-1}}$，$p$表示随机 SVD 分解中的超参数，$\boldsymbol{K}_{ii}^*=max\boldsymbol{K}_{ii}$。

证明：根据式(2.3)和式(2.5)，可以得到：

$$\begin{aligned}\tilde{\boldsymbol{K}} &= \boldsymbol{C}\boldsymbol{W}_k^{+}\boldsymbol{C}^T = \boldsymbol{C}\boldsymbol{U}_W\boldsymbol{\Lambda}_F^{+}\boldsymbol{U}_W^T\boldsymbol{C}^T = \boldsymbol{C}(\boldsymbol{Q}\boldsymbol{F}\boldsymbol{Q}^T)^{+}\boldsymbol{C}^T \\ &= \boldsymbol{C}(\boldsymbol{Q}\boldsymbol{Q}^T\boldsymbol{W}\boldsymbol{Q}\boldsymbol{Q}^T)^{+}\boldsymbol{C}^T \\ &= \boldsymbol{C}\boldsymbol{Q}(\boldsymbol{Q}^T\boldsymbol{W}\boldsymbol{Q})^{+}\boldsymbol{Q}^T\boldsymbol{C}^T \\ &= \boldsymbol{X}^T\boldsymbol{H}\boldsymbol{Q}(\boldsymbol{Q}^T\boldsymbol{H}^T\boldsymbol{H}\boldsymbol{Q})^{+}\boldsymbol{Q}^T\boldsymbol{H}^T\boldsymbol{X} \\ &= \boldsymbol{X}^T\boldsymbol{P}_{HQ}\boldsymbol{X} \\ &= \boldsymbol{X}^T\boldsymbol{U}_R\boldsymbol{U}_R^T\boldsymbol{X}\end{aligned} \tag{2.10}$$

所以，

$$||\boldsymbol{K} - \tilde{\boldsymbol{K}}||_F = ||\boldsymbol{X}^T\boldsymbol{X} - \boldsymbol{X}^T\boldsymbol{U}_R\boldsymbol{U}_R^T\boldsymbol{X}||_F$$

其中，$\boldsymbol{U}_R$是$\boldsymbol{R}$的正交基，$\boldsymbol{R} = \boldsymbol{H}\boldsymbol{Q}$，$\boldsymbol{Q}$为算法 2.2 中 QR 分解得到的矩阵。

因为$\boldsymbol{I} - \boldsymbol{U}_R\boldsymbol{U}_R^T$是正交投影，也是半正定矩阵。所以，对于任意的向量$\mu$，

$$\mu^T\boldsymbol{X}^T(\boldsymbol{I} - \boldsymbol{U}_R\boldsymbol{U}_{\boldsymbol{R}}^T)\boldsymbol{X}\mu = (\boldsymbol{X}\mu)^T(\boldsymbol{I} - \boldsymbol{U}_R\boldsymbol{U}_R^T)(\boldsymbol{X}\mu) \geqslant 0$$

则，$\boldsymbol{X}^T\boldsymbol{X} - \boldsymbol{X}^T\boldsymbol{U}_R\boldsymbol{U}_R^T\boldsymbol{X}$也是半正定的。

所以，$||\boldsymbol{K} - \tilde{\boldsymbol{K}}||_F \leqslant Tr(\boldsymbol{X}^T\boldsymbol{X} - \boldsymbol{X}^T\boldsymbol{U}_R\boldsymbol{U}_R^T\boldsymbol{X}) = ||\boldsymbol{X}||_F^2 - ||\boldsymbol{U}_R^T\boldsymbol{X}||_F^2$

根据(Li M 等,2014)中的引理 4 和引理 5，可以得到，

$$\begin{aligned}||\boldsymbol{X}||_F^2 - ||\boldsymbol{U}_R^T\boldsymbol{X}||_F^2 \leqslant & \sum_{i>k+p}\sigma_i^2(\boldsymbol{X}) + 2\sqrt{k+p}||\boldsymbol{X}\boldsymbol{X}^T - \boldsymbol{H}\boldsymbol{H}^T||_F \\ & + 2\sqrt{k+p}||\boldsymbol{H}\boldsymbol{H}^T - \boldsymbol{R}\boldsymbol{R}^T||_F\end{aligned}$$

令$\boldsymbol{P} = \boldsymbol{I} - \boldsymbol{Q}\boldsymbol{Q}^T$为一组正交投影，则

$$\begin{aligned}||\boldsymbol{H}\boldsymbol{H}^T - \boldsymbol{R}\boldsymbol{R}^T||_F^2 &= ||\boldsymbol{H}\boldsymbol{H}^T - \boldsymbol{H}\boldsymbol{Q}\boldsymbol{Q}^T\boldsymbol{H}^T||_F^2 \\ &= Tr(\boldsymbol{H}\boldsymbol{P}\boldsymbol{H}^T\boldsymbol{H}\boldsymbol{P}\boldsymbol{H}^T) \\ &= |||\boldsymbol{P}\boldsymbol{H}^T\boldsymbol{H}\boldsymbol{P}||_F \\ &\leqslant ||\boldsymbol{P}\boldsymbol{H}^T\boldsymbol{H}||_F \\ &= ||\boldsymbol{H}^T\boldsymbol{H} - \boldsymbol{Q}\boldsymbol{Q}^T\boldsymbol{H}^T\boldsymbol{H}||_F^2 \\ &= ||\boldsymbol{W} - \boldsymbol{Q}\boldsymbol{Q}^T\boldsymbol{W}||_F^2\end{aligned}$$

根据引理 2.1 和式(2.6)，令$\alpha = \sqrt{1 + \frac{k}{p-1}}$，

$$
\begin{aligned}
E||\boldsymbol{H}\boldsymbol{H}^T - \boldsymbol{R}\boldsymbol{R}^T||_F^2 &= E||\boldsymbol{W} - \boldsymbol{Q}\boldsymbol{Q}^T\boldsymbol{W}||_F^2 \\
&\leqslant \alpha\left[\sum_{i>k}\sigma_i^2(\boldsymbol{W})\right]^{1/2} = \alpha\left[\sum_{i>k}\sigma_i^2(\boldsymbol{H}\boldsymbol{H}^T)\right]^{1/2} \\
&\leqslant \alpha\left[\sum_{i>k}\sigma_i^2(\boldsymbol{X}\boldsymbol{X}^T)\right]^{1/2} + \alpha||\boldsymbol{X}\boldsymbol{X}^T - \boldsymbol{H}\boldsymbol{H}^T||_F \\
&= \alpha||\boldsymbol{K} - \boldsymbol{K}_k||_F + \alpha||\boldsymbol{X}\boldsymbol{X}^T - \boldsymbol{H}\boldsymbol{H}^T||_F
\end{aligned}
$$

综上，根据定理 2.1，令$\xi = \frac{2(k+p)}{\sqrt{p-1}}$，$\boldsymbol{K}_{ii}^* = max\boldsymbol{K}_{ii}$

$$
\begin{aligned}
&E(||\boldsymbol{K} - \tilde{\boldsymbol{K}}||_F) \\
&\leqslant \sum_{i>k+p}\sigma_i(\boldsymbol{K}) + 2\sqrt{k+p}E_H||\boldsymbol{X}\boldsymbol{X}^T - \boldsymbol{H}\boldsymbol{H}^T||_F + 2\sqrt{k+p}||\boldsymbol{H}\boldsymbol{H}^T - \boldsymbol{R}\boldsymbol{R}^T||_F \\
&\leqslant \sum_{i>k+p}\sigma_i(\boldsymbol{K}) + 2\alpha\sqrt{k+p}||\boldsymbol{K} - \boldsymbol{K}_k||_F + 2(1+\alpha)\sqrt{k+p}E||\boldsymbol{X}\boldsymbol{X}^T - \boldsymbol{H}\boldsymbol{H}^T||_F \\
&\leqslant \sum_{i>k+p}\sigma_i(\boldsymbol{K}) + 2\alpha\sqrt{k+p}||\boldsymbol{K} - \boldsymbol{K}_k||_F + 2(1+\alpha)\sqrt{k+p}\sqrt{\frac{1}{m}}(max||\boldsymbol{K}^{(i)}||^2) \\
&= \sum_{i>k+p}\sigma_i(\boldsymbol{K}) + \frac{2(k+p)}{\sqrt{p-1}}||\boldsymbol{K} - \boldsymbol{K}_k||_F + \frac{4(k+p)}{\sqrt{p-1}}\sqrt{\frac{1}{m}}(max||\boldsymbol{K}^{(i)}||^2) \\
&\leqslant \xi||\boldsymbol{K} - \boldsymbol{K}_k||_F + 2\xi\sqrt{\frac{1}{m}}(max_i||\boldsymbol{K}^{(i)}||^2 + n\boldsymbol{K}_{ii}^* \\
&\leqslant \xi||\boldsymbol{K} - \boldsymbol{K}_k||_F + 2\xi\sqrt{\frac{1}{m}}\boldsymbol{K}_{ii}^* + n\boldsymbol{K}_{ii}^* \\
&= \xi||\boldsymbol{K} - \boldsymbol{K}_k||_F + (n + 2\xi\sqrt{\frac{1}{m}})\boldsymbol{K}_{ii}^*
\end{aligned}
$$

证毕。

2.4 计算复杂度分析

本部分讨论本章提出的不等概抽样方法结合随机 SVD 分解的 Nyström 低秩逼近方法计算复杂度，并与均匀抽样 Nyström 方法、概率增量抽样 Nyström 方法的计算复杂度进行比较。

本节中，n表示数据矩阵的总列数，m表示抽样的样本列数，k表示所期望的低秩，M表示原数据矩阵的相似矩阵中非零点的数目。标准 Nyström 方法的计算复杂度为$O(kmn + m^3)$，$O(kmn)$表示在 Nyström 方法中使用均匀抽样构造样本

矩阵的计算复杂度，现用本文中应用的不等概抽样替换均匀抽样，不等概抽样的计算复杂度为$O(min\{m,n\}k^2)$(贾洪杰,2017)；$O(m^3)$表示 SVD 分解的计算复杂度，现用本文中的随机 SVD 分解替换 SVD 分解，即用$O(m^2k+k^3)$替换$O(m^3)$，则算法整体的计算复杂度为$O(min\{m,n\}k^2+m^2k+k^3)$。

同样地，均匀抽样结合随机 SVD 分解的 Nyström 方法的计算复杂度为$O(kmn+m^2k+k^3)$。概率增量抽样的计算复杂度为$O(Mm+nm^2)$(贾洪杰,2017)，结合随机 SVD 分解的 Nyström 方法计算复杂度为$O(Mm+nm^2+m^2k+k^3)$。

将均匀抽样 Nyström 方法、概率增量抽样 Nyström 方法和本文提出的方法的计算复杂度进行对比，见表 2.1。

表 2.1 三种方法计算复杂度对比

方法	计算复杂度
均匀抽样 Nyström 方法	$O(kmn+m^2k+k^3)$
概率增量抽样 Nyström 方法	$O(Mm+nm^2+m^2k+k^3)$
提出的方法	$O(min\{m,n\}k^2+m^2k+k^3)$

由表 2.1 可得，概率增量抽样结合随机 SVD 分解的 Nyström 方法的计算复杂度较高，因为$k \leqslant m$，所以不等概抽样结合随机 SVD 分解的 Nyström 方法的计算复杂度小于或近似于均匀抽样结合随机 SVD 分解的 Nyström 方法。

2.5 数值检验

2.5.1 模拟数据生成

为验证本节提出的不等概抽样结合随机 SVD 的 Nyström 方法的有效性，随机生成两组两类别样本行为 200，维度为 200 的数据集，分别记为 data1 和 data2。data1 服从形状参数$\alpha=1$，位置参数$\beta=1$的伽马分布，data2 服从均值$\mu=0$，标准差$\sigma=1$的正态分布。将 data1 和 data2 中心标准化后，通过相关运算转换为$\boldsymbol{K}\in\mathbb{R}^{200\times200}$的 SPSD 矩阵。

2.5.2 方法模拟与精度比较

本节通过低秩逼近矩阵重构的相对误差对方法精度做出对比。令k等于交叉矩阵的秩，分别比较基于不等概抽样、均匀抽样和概率增量抽样的 Nyström 方法在标准 SVD 分解和随机 SVD 分解下的低秩逼近重构误差，具体如表 2.2 所示。相对误差数值越小，说明方法精度越高。

表 2.2 随机模拟数据低秩逼近相对误差

样本比例	数据集	Nyström 的抽样方法					
		均匀抽样		概率增量抽样		不等概抽样	
		标准 SVD	随机 SVD	标准 SVD	随机 SVD	标准 SVD	随机 SVD
5%	data1	12.27	11.94	-	-	10.17	**9.08**
	data2	22.25	13.28	19.14	11.17	16.84	**9.48**
10%	data1	11.84	11.26	3.55	10.54	9.22	**8.56**
	data2	20.21	11.56	18.49	9.31	14.13	**8.49**
20%	data1	10.57	9.38	17.73	8.87	9.09	**8.33**
	data2	16.68	11.19	14.13	7.93	9.48	**6.09**
30%	data 1	9.73	8.59	3.40	8.15	8.72	**7.91**
	data2	14.13	9.94	10.86	7.07	6.51	**5.38**
40%	data 1	8.68	7.98	4.85	7.34	8.46	**6.81**
	data 2	11.44	6.94	10.74	6.51	5.45	**2.51**
50%	data 1	8.39	**6.57**	3.25	5.30	7.95	**5.05**
	data 2	7.02	5.99	6.04	5.86	3.38	**2.38**

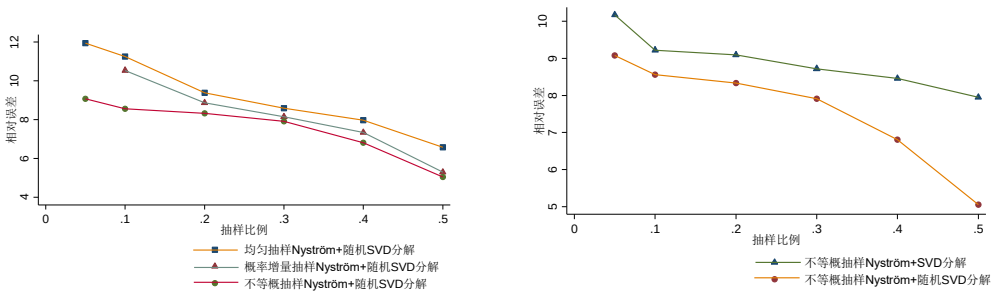
表 2.2 结果显示，通过基于抽样进行低秩逼近相对误差的平均值来看，对于 data1，在均匀抽样 Nyström 方法下低秩逼近误差随着抽样比例增加而平稳降低；随机 SVD 分解方法下的平均误差均低于 SVD 分解方法下的误差；概率增量抽样 Nyström 方法下，抽样比例为 5%时，该方法因为样本量较少，抽样方法无法迭代，无法低秩逼近；在 SVD 分解方法下，低秩逼近误差变化较不稳定，抽样比例为 20%时，误差为 17.73，远远高于该抽样比例时随机 SVD 下的误差；而在不等概抽样 Nyström 方法下，随机 SVD 分解方法下的误差均值皆低于 SVD 分解下的误差。

对于 data2，使用均匀抽样 Nyström 方法低秩逼近时，随机 SVD 分解方法下的误差均低于 SVD 分解下的值，且在较小的抽样比例时两方法下误差值相差较大。抽样比例为 50%时，SVD 分解下的误差值为 7.02，随机 SVD 的误差值为 5.99，

相比其他抽样比例下的值较为相近。使用概率增量抽样 Nyström 方法时与均匀抽样 Nyström 方法类似，在较小的抽样比例下两种分解方法的低秩逼近误差值相差较大。比如抽样比例为 5%时，SVD 分解下的误差值为 19.14，随机 SVD 分解下的值为 11.17，但整体上随机 SVD 分解下的误差值仍是低于 SVD 分解下的误差值。但对于不等概 Nyström 方法，该数据集的低秩逼近误差值在 SVD 分解方法下的降低速度高于随机 SVD 下的值同时精度低于后者。

总的来说，对于两个模拟数据集，本节提出的不等概抽样结合随机 SVD 分解的 Nyström 方法的误差值在所选抽样比例下，均低于均匀抽样、概率增量抽样结合随机 SVD 分解的 Nyström 方法。且随着抽样比例增加，该方法的误差降低，精度逐渐升高。

进一步，为了说明不同的抽样方法对 Nyström 低秩逼近误差的影响，绘制了数据集 data1 和 data2 在不等概抽样 Nyström 方法、均匀抽样 Nyström 方法和概率增量抽样 Nyström 方法下，利用随机 SVD 分解方法对交叉矩阵低秩近似时，三种低秩逼近重构方法相对误差随着抽样比例增加的变化折线图（图 2.1、图 2.2、a）。为了明显区分 SVD 分解方法和随机 SVD 分解方法对低秩逼近的影响效果，将两类数据在不等概抽样 Nyström 方法中，分别使用 SVD 分解和随机 SVD 分解后的低秩逼近误差随着抽样比例变化的趋势可视化（图 2.1、图 2.2，b）。

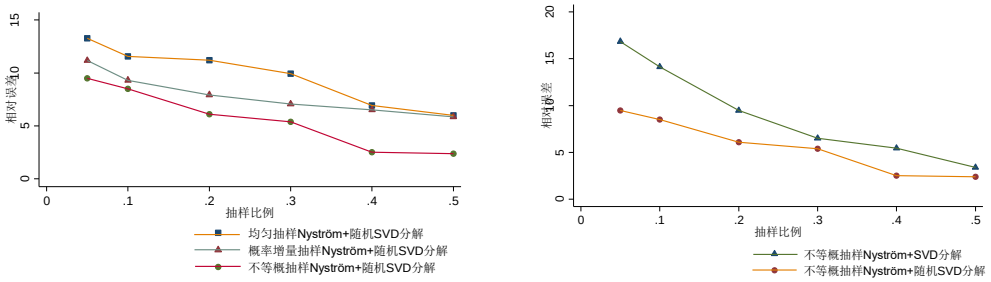


a(1) b(1)

图 2.1 data1 相对误差对比

a(1)显示，对于 data1，三种抽样方法结合随机 SVD 分解的 Nyström 低秩逼近重构误差随着抽样比例增加都呈递减趋势。抽样比例小于 20%时，三种方法误差相差较大，20%至 40%时，三种方法误差值较为接近，抽样比例为 50%时，不等概抽样结合随机 SVD 分解的 Nyström 的误差接近概率增量抽样 Nyström 误差。b(1)显示，不等概抽样结合随机 SVD 分解的 Nyström 低秩逼近重构误差远远低

于该抽样方法结合 SVD 分解时的 Nyström 误差。



a(2) b(2)

图 2.2 data2 相对误差对比

a(2)显示，对于 data2，抽样比例小于 10%时，三种抽样方法结合随机 SVD 分解的 Nyström 低秩逼近误差相近，随着抽样比例增加，不等概抽样结合随机 SVD 分解的 Nyström 低秩逼近重构误差明显低于其他两种方法，且误差值降低幅度也高于其他两种方法。b(2)显示，对于 data2，抽样比例为 30%和 50%时，在同样的抽样方法下，两种低秩分解方法对整体低秩逼近重构误差影响效果相近，其他抽样比例下，随机 SVD 分解误差远远低于 SVD 分解。

2.5.3 方法模拟运行时间比较

本部分从方法的运行时间出发，比较不等概抽样 Nyström 方法、均匀抽样 Nyström 方法和概率增量抽样 Nyström 方法在不同采样比例下的优良性，运行时间越少，算法复杂度越低。如表 2.3 所示。

表 2.3 算法关于模拟数据的运行时间（s）

样本比例	数据集	Nyström 抽样方法					
		概率增量抽样		均匀抽样		不等概抽样	
		标准 SVD	随机 SVD	标准 SVD	随机 SVD	标准 SVD	随机 SVD
5%	data1	-	-	2.14	2.11	1.92	2.00
	data2	5.32	8.23	2.08	2.81	2.07	2.73
10%	data1	2.24	3.49	2.06	2.24	1.92	2.05
	data2	6.33	9.56	2.12	3.18	2.03	2.78
20%	data1	4.23	5.24	2.20	2.31	2.08	2.20
	data2	9.39	11.42	2.29	2.93	2.20	2.63
30%	data 1	5.25	6.02	2.29	2.50	2.42	2.36
	data2	11.43	12.50	2.47	2.97	2.39	3.00
40%	data 1	5.25	8.44	2.61	2.97	2.77	3.02
	data 2	12.40	15.64	2.74	3.60	2.58	3.17
50%	data 1	8.28	8.93	2.92	3.39	2.97	3.36
	data 2	16.5	17.10	3.17	3.38	2.89	3.31

由表 2.3 可得， data1 抽样比例为 5%、10%、20%时，不等概抽样结合 SVD 分解的 Nyström 方法运行时间较少，分别为 1.92s、1.92s、2.08s。抽样比例为 30%，40%，50%时，均匀抽样结合 SVD 分解的 Nyström 方法运行时间较少，分别为 2.29s，2.61s 和 2.92s。

对于 data2，在所有的抽样比例下，不等概抽样结合 SVD 分解的 Nyström 方法运行时间最少，远远低于概率增量抽样 Nyström 方法运行时间。

整体来看，三类数据使用不等概抽样 Nyström 的运行时间，少于或接近均匀抽样 Nyström 运行时间，皆低于概率增量抽样 Nyström 运行时间。此外，三种抽样方法结合 SVD 分解的 Nyström 方法运行时间少于结合随机 SVD 时的运行时间，且随着抽样比例增加，三种方法在两种不同低秩分解方法下的运行时间都呈上升趋势增加。

2.6 小结

Nyström 方法是一种通过抽样进行矩阵低秩逼近降维的有效手段，而以抽样子空间为基础的矩阵低秩逼近技术其关键点是样本列的选取。本节基于不等概抽样方法，构造了不等概抽样 Nyström 方法。构建样本子空间时，利用不等概抽样提取样本列，可以充分考虑到矩阵每列信息的不同，使样本更具有代表性、随机性、且能充分保留原数据有效信息。此外，在矩阵重构部分结合随机 SVD 分解，可加速求解交叉矩阵的奇异向量，降低矩阵重构误差。且在样本列较少时，随机 SVD 分解可通过调整过采样参数降低算法失效概率。研究结果显示，基于不等概抽样和随机 SVD 分解的 Nyström 方法不仅保证了抽出样本的有效性，而且在提高矩阵低秩逼近精度的同时有效降低了计算复杂度。

通过模拟数据和实证数据对本文提出的方法与均匀抽样和概率增量抽样结合随机 SVD 分解的 Nyström 方法进行对比分析。总体来说，相比均匀抽样和概率增量抽样结合随机 SVD 分解的 Nyström 方法，本节提出的不等概抽样结合随机 SVD 分解的 Nyström 方法精度更高，计算复杂度低于或近似于均匀抽样结合随机 SVD 分解的 Nyström 方法，远远低于概率增量抽样结合随机 SVD 分解的 Nyström 方法。

3 基于不等概自适应抽样和随机 SVD 分解的 CUR 矩阵重构

将 SPSD 矩阵推广到任意高维矩阵的低秩逼近，CUR 矩阵分解是一种有效替代方法。CUR 矩阵分解相比 SVD 分解、QR 分解和 Nyström 低秩逼近方法，既能通过矩阵行列抽样降低数据维度，简化计算复杂度，又可以将通过抽样得到的矩阵对原始数据矩阵进行近似重构，且能保持矩阵的稀疏性和非负性。

类似于 Nyström 低秩逼近列抽样，在 CUR 矩阵分解中，矩阵列或行的抽样方法选择，对于矩阵低秩逼近精度具有重要影响。其中，均匀抽样是较为常用的抽样方法，也是较早运用到矩阵低秩逼近技术中的抽样方法(Williams,2001; Kumar,2009)。对于信息列分布不均匀的数据，均匀抽样容易忽略重要信息，因此该抽样方法不适合包含较多信息列的数据；这时，通过计算对角线元素权重(对角抽样)或矩阵列的范数(列范数抽样)进行非均匀抽样的抽样方法逐渐出现(Drineas,2005; Kannan,2006)。进一步，对于一些由核函数构造的矩阵，根据 Leverage 得分进行抽样较为合适(Drineas,2008)；Wang(2016)明确提出 Leverage 得分抽样算法，用于加速 SPSD 矩阵和 CUR 矩阵分解的低秩逼近速度。为了使抽样的精确度更高， Deshpande 等(2006)首次提出理论基础较强的自适应抽样方法的思想，并证明了抽样误差随着迭代次数增加呈指数下降；Kumar 等(2012)认为矩阵每一列的信息随着抽样过程在不断变化，于是提出新的自适应抽样方法，并将其与 Nyström 方法结合，提出自适应 Nyström 抽样方法；Wang 等(2013)将自适应抽样用于 SPSD 矩阵逼近和 CUR 矩阵分解，提出该抽样方法下矩阵低秩逼近关于 F 范数存在的误差界。

均匀抽样基于原始数据信息具有一致性重要的思想，不需遍历数据集，所以抽样效率较高；但一般大规模数据矩阵的列的信息重要性不尽相同，均匀抽样有可能会使样本集中于同一个区域，从而降低样本的代表性，损失重要信息。对角抽样、列范数抽样等非均匀抽样方法考虑了矩阵列的不同重要性，一般通过放回抽样简化分析复杂度，相比均匀抽样需要更多的计算成本和存储空间，效率较低。Leverage 得分抽样方法基于矩阵每一列的 Leverage 得分抽取矩阵列，在此过程中需要对矩阵进行 SVD 分解，计算复杂度较高，在实际大规模数据应用中，

Leverage 得分无法有效计算(Kumar,2012)。此外，以上抽样方法都是随机抽样，选取的样本都具有随机性，该优良性更适用于 SPSD 矩阵的低秩逼近；而对于其他任意矩阵，CUR 矩阵分解需要同时选取矩阵的部分行和部分列，为保证算法的有效性，考虑每步抽取的行数与列数相同，此时随机抽样不适用于 CUR 矩阵分解；因此相比其他抽样方法，自适应抽样更加有效，可通过迭代更新入样概率，降低抽样误差。

通过抽样得到列和行的子矩阵$\boldsymbol{C}$和$\boldsymbol{R}$后，需要考虑能够联合$\boldsymbol{C}$、$\boldsymbol{R}$和原矩阵的联合矩阵$\boldsymbol{U}$的计算。Stewart(1999)提出一种基于 QR 分解的称为稀疏行列近似的 CUR 矩阵分解方法，通过对原矩阵$\boldsymbol{K}$进行 QR 分解得到一组列，构成矩阵$\boldsymbol{C}$，并计算一个上三角矩阵$\boldsymbol{T}_C$使这些列正交，同样的方法可以找到行矩阵$\boldsymbol{R}$和$\boldsymbol{T}_R$，则$\boldsymbol{U}=(\boldsymbol{T}_C^T\boldsymbol{T}_C)^{-1}\boldsymbol{C}^T\boldsymbol{K}\boldsymbol{R}^T(\boldsymbol{T}_R^T\boldsymbol{T}_R)^{-1}$；该方法在抽取的列数和行数较大时计算复杂度较高，且没有明确的误差界。Wang(2016)利用 Leverage 得分抽样对$\boldsymbol{C}$和$\boldsymbol{R}$进行二次抽样，通过二次抽样后的矩阵近似计算矩阵$\boldsymbol{U}$，以此加速 CUR 矩阵分解 Wang(2016)；该方法会因为二次抽样损失大量原始数据信息，使低秩逼近误差增大。而其他多数学者大多数通过对$\boldsymbol{C}$和$\boldsymbol{R}$求伪逆矩阵，将其与原矩阵的乘积作为矩阵$\boldsymbol{U}$的近似矩阵(Li,2010)。

Halko 等(2009)提出了一种简单但精确度较高的随机算法，即随机 SVD 分解算法，用于构造低秩近似矩阵。Mu Li(2010)将 Nyström 方法与随机 SVD 算法结合，对交叉矩阵进行随机 SVD 分解，提高矩阵低秩逼近精度。

本节首先在传统自适应抽样方法的基础上，提出不等概自适应抽样法。将不等概抽样的思想与自适应抽样结合，充分保留原始数据信息的同时，使样本列和样本行更具代表性；降低抽样误差的同时，提高抽样效率。其次，通过随机 SVD 分解对子矩阵$\boldsymbol{C}$和$\boldsymbol{R}$进行低秩逼近，简化时间和空间存储复杂度；再利用随机 SVD 分解低秩逼近后得到的矩阵$\boldsymbol{D}_C$和$\boldsymbol{D}_R$，联合$\boldsymbol{C}$、$\boldsymbol{R}$和原矩阵近似逼近矩阵$\boldsymbol{U}$，进一步低秩重构原始高维矩阵，此过程可降低由于二次抽样引起的信息损失，提高整体 CUR 矩阵分解逼近重构的精度，且具有明确的误差界。

3.1 方法介绍

3.1.1CUR 矩阵分解

CUR 矩阵分解是一种较有效的矩阵低秩逼近方法。给定一个矩阵$\boldsymbol{K} \in \mathbb{R}^{m\times n}$，对其进行 CUR 矩阵分解，即指从$\boldsymbol{K}$中选取$c(c<n)$列构成矩阵$\boldsymbol{C} \in \mathbb{R}^{m\times c}$，选取$r(r<m)$行构成矩阵$\boldsymbol{R} \in \mathbb{R}^{r\times n}$，$\boldsymbol{U} \in \mathbb{R}^{c\times r}$满足$\boldsymbol{K}=\boldsymbol{CUR}+\boldsymbol{E}$，$\boldsymbol{E}$为残差矩阵，即$\boldsymbol{E}=\boldsymbol{K}-\boldsymbol{CUR}$。在该算法中，$\boldsymbol{C}$和$\boldsymbol{R}$包含了$\boldsymbol{K}$的原始列和原始行，保证了特征选择和数据解释的准确性。从计算的角度出发，该方法的挑战性在于如何高效构造矩阵$\boldsymbol{C}$、$\boldsymbol{R}$和$\boldsymbol{U}$，使得$||\boldsymbol{K}-\boldsymbol{CUR}||_F^2$的值最小。而由于$\boldsymbol{C}$和$\boldsymbol{R}$是抽样构成，所以一般通过寻找最优的$\boldsymbol{U}$矩阵，使$||\boldsymbol{K}-\boldsymbol{CUR}||_F^2$值达到最小，即

$$\boldsymbol{U}^* = \underbrace{argmin}_{\boldsymbol{U}} ||\boldsymbol{K}-\boldsymbol{CUR}||_F^2 \tag{3.1}$$

通常，$\boldsymbol{U}^*=\boldsymbol{C}^+\boldsymbol{K}\boldsymbol{R}^+$，(Wang 等,2013;Boutsidis 等,2017)。

其中，$\boldsymbol{C}^+=\boldsymbol{V}_C\boldsymbol{\Sigma}_C^{-1}\boldsymbol{U}_C^T, \boldsymbol{R}^+=\boldsymbol{V}_R\boldsymbol{\Sigma}_R^{-1}\boldsymbol{U}_R^T$分别表示矩阵$\boldsymbol{C}$、$\boldsymbol{R}$的伪逆矩阵，$\boldsymbol{U}_C$、$\boldsymbol{\Sigma}_C$、$\boldsymbol{V}_C$分别表示矩阵$\boldsymbol{C}$的左奇异向量、奇异值和右奇异向量，$\boldsymbol{U}_R$、$\boldsymbol{\Sigma}_R$和$\boldsymbol{V}_R$分别表示矩阵$\boldsymbol{R}$的左奇异向量、奇异值和右奇异向量。

在 CUR 矩阵分解中，计算伪逆矩阵的时间复杂度为$O(mc^2+nr^2)$，矩阵相乘的时间复杂度为$O(mn\cdot min\{c,r\})$，小于 SVD 分解的复杂度$O(n^3)$。

3.1.2 自适应抽样

对于一个任意矩阵$\boldsymbol{K} \in \mathbb{R}^{m\times n}$，自适应抽样是一种较为有效的抽样方法，可以通过多次迭代更新抽样降低只进行一轮抽样的误差。一般的自适应抽样为两轮抽样，以抽取矩阵列为例进行具体说明：在进行第一轮抽样后，抽出c_1列，得到原始矩阵的列子集构成的矩阵，记为矩阵$\boldsymbol{C_1} \in \mathbb{R}^{m\times c_1}$，计算得到其伪逆矩阵$\boldsymbol{C}_1^+$，进一步得到第一轮抽样后的残差矩阵$\boldsymbol{A}=\boldsymbol{K}-\boldsymbol{C_1}\boldsymbol{C}_1^+\boldsymbol{K}$，再根据$p_i=\frac{||a_i||_2^2}{||\boldsymbol{A}||_F^2}$计算第二轮抽样的第$i$列入样概率，并以此概率从矩阵$\boldsymbol{K}$中抽出$c_2$列。该抽样方法可以抽取到包含较多信息的样本列，抽样误差随着抽样次数增加呈指数下降。

3.1.3 CUR 矩阵分解中的自适应抽样

将自适应抽样可以进行有效列抽样的优良性扩展到矩阵行抽样，应用于CUR 矩阵分解，提升矩阵分解重构精度。基本思想是：给定一个矩阵$\boldsymbol{K} \in \mathbb{R}^{m\times n}$，通过任意抽样方法抽出$c$列构造矩阵$\boldsymbol{C} \in \mathbb{R}^{m\times c}$，满足$rank(\boldsymbol{C}) = rank(\boldsymbol{CC^{+}K}) = r(r \leqslant c \leqslant n)$，抽出$r_1$行构造矩阵$\boldsymbol{R}_1 \in \mathbb{R}^{r_1\times n}$，此过程为第一轮抽样；再计算残差矩阵$\boldsymbol{A} = \boldsymbol{K} - \boldsymbol{KR_1^{+}R_1}$，根据$p_i = \frac{||a_i||_2^2}{||\boldsymbol{A}||_F}$计算第二轮抽样中第$i$行的入样概率，进一步根据入样概率$p_i$，从矩阵$\boldsymbol{K}$中抽出$r_2$行，构造矩阵$\boldsymbol{R}_2 \in \mathbb{R}^{r_2\times n}$，因此，就有矩阵$\boldsymbol{R} = [\boldsymbol{R_1^T}, \boldsymbol{R_2^T}]^T \in \mathbb{R}^{(r_1+r_2)\times n}$。

基于该抽样的 CUR 矩阵分解低秩逼近的误差关于 F 范数的期望有（Wang,2013），

$$\mathbb{E}||\boldsymbol{K} - \boldsymbol{CC^{+}KR^{+}R}||_F^2 \leqslant ||\boldsymbol{K} - \boldsymbol{CC^{+}K}||_F^2 + \frac{r}{r_2}||\boldsymbol{K} - \boldsymbol{KR_1^{+}R_1}||_F^2$$

其中，$\boldsymbol{C}^{+}$为矩阵$\boldsymbol{C}$的伪逆矩阵，$\boldsymbol{R}^{+}$为矩阵$\boldsymbol{R}$的伪逆矩阵，$||\cdot||$为矩阵的 F 范数。

3.2 基于不等概自适应抽样的 CUR 矩阵分解

3.2.1 不等概抽样

一般的自适应抽样结合均匀抽样等随机抽样方法抽取矩阵列或行，虽然能保证样本具有随机性，但不能保证抽样的有效性和稳定性，不适用于 CUR 矩阵分解算法。因此，本文结合不等概抽样的思想，充分考虑矩阵每列每行信息差异性，将矩阵列向量、行向量与矩阵的范数之比作为第i列、第j行的入样概率，即

$$P_i = \frac{||\boldsymbol{K}^{(i)}||_2^2}{||\boldsymbol{K}||_F^2}; P_j = \frac{||\boldsymbol{K}^{(j)}||_2^2}{||\boldsymbol{K}||_F^2} \tag{3.2}$$

根据抽出信息最大化原则，通过以上入样概率P_i对矩阵$\boldsymbol{K}$进行列抽样，通过P_j对矩阵$\boldsymbol{K}$进行两轮行抽样，抽出入样概率值较大的某些列构造列子集矩阵，并将两轮抽样抽出的行子集合并，构造行子集矩阵。

3.2.2 不等概自适应抽样

针对 CUR 矩阵分解需要同时从原始数据矩阵中抽取c列和r行，构造能有效代表原矩阵信息的子矩阵$\boldsymbol{C}$和$\boldsymbol{R}$的算法特征，且满足在保证抽取“好”的样本列时，也能保证抽取 “好”的样本行，本文进行两轮抽样。首先假设抽取的列数c与第一轮抽样中的行数r_1相同，第二轮抽样的行数为$r_2 = 0.5r_1$；引入一个迭加参数a，在秩k固定时，通过迭加参数的变化，使抽样的列数和行数变动，且能通过调整a的取值范围使总的样本列数和行数控制在总体范围内，即$c \leqslant n, r = r_1 + r_2 \leqslant m$，从而保证抽样的有效性和一致性，具体的 CUR 矩阵分解的抽样过程如算法 3.1 所示。

算法 3.1 不等概自适应抽样

输入：任意矩阵$\boldsymbol{K} \in \mathbb{R}^{m\times n}$，目标秩$k$，迭加参数$a$。

输出：子矩阵$\boldsymbol{C} \in \mathbb{R}^{m\times c}$；$\boldsymbol{R} \in \mathbb{R}^{(r_1+r_2)\times n}$。

Step1：计算矩阵$\boldsymbol{K}$每一列的入样概率，$p_i = \frac{||k_i||_2^2}{||\boldsymbol{K}||_F}, i = 1, \cdots, n$ ；

Step2：计算矩阵$\boldsymbol{K}$每一行的入样概率，$p_j = \frac{||k_j||_2^2}{||\boldsymbol{K}||_F}, j = 1, \cdots, m$ ；

Step3：根据p_i从$\boldsymbol{K}$中抽出入样概率较大的c列，构造矩阵$\boldsymbol{C} \in \mathbb{R}^{m\times c}$，$c = ak$；

Step4：根据p_j从$\boldsymbol{K}$中抽出入样概率较大的r_1行，构造矩阵$\boldsymbol{R}_1 \in \mathbb{R}^{r_1\times n}$，$r_1 = c$；

Step5：计算残差矩阵$\boldsymbol{A} = \boldsymbol{K} - \boldsymbol{K}\boldsymbol{R}_1^{+}\boldsymbol{R}_1$；

Step6：计算残差矩阵$\boldsymbol{A}$每一行的入样概率$p_l = \frac{||a_l||_2^2}{||\boldsymbol{A}||_F}, l = 1, \cdots, m - r_1$；

Step7：根据p_l从$\boldsymbol{K}$抽取入样概率较大的r_2行，构造矩阵$\boldsymbol{R}_1 \in \mathbb{R}^{r_2\times n}$，$r_2 = 0.5r_1$；

Step8：合并$\boldsymbol{R}_1, \boldsymbol{R}_2$，得到矩阵$\boldsymbol{R} = [\boldsymbol{R}_1, \boldsymbol{R}_2]$。

算法 3.1 中，计算矩阵$\boldsymbol{R}_1$的伪逆矩阵的计算复杂度为$O(nr^2)$，矩阵相乘的计算复杂度为$O(mn \cdot min\{c, r\})$。

3.2.3 抽样子矩阵的随机 SVD 分解

为使矩阵分解算法具有更高的精确度，本文利用随机 SVD 分解方法对矩阵$\boldsymbol{C}$和矩阵$\boldsymbol{R}^T$进行低秩逼近，以矩阵$\boldsymbol{C}$的随机 SVD 分解步骤为例进行具体说明，具体见算法 3.2。

算法 3.2	矩阵$\boldsymbol{C}$随机 SVD 分解
输入：	矩阵$\boldsymbol{C}\in\mathbb{R}^{m\times c}$，秩$r$,目标低秩$k$。
输出：	特征向量矩阵$\boldsymbol{U}_{DC}$ ，低秩逼近矩阵$\boldsymbol{D}_C$
step1：	构造一个$c\times k$的标准 Gaussian 随机矩阵$\boldsymbol{\Omega}$;
step2：	计算一个新的矩阵，$\boldsymbol{Z}\in\mathbb{R}^{m\times k}$，$\boldsymbol{Z}=\boldsymbol{C\Omega}$;
step3：	对$\boldsymbol{Z}$进行 QR 分解得到一个正交矩阵$\boldsymbol{Q}$，使$\boldsymbol{Y}=\boldsymbol{QQ}^T\boldsymbol{C}$;
step4：	计算约化矩阵$\boldsymbol{F}$,$\boldsymbol{F}=\boldsymbol{Q}^T\boldsymbol{Y}$;
step5：	对$\boldsymbol{F}$进行 SVD 分解得到$\boldsymbol{F}=\boldsymbol{U}_F\boldsymbol{\Lambda}_F\boldsymbol{V}_F$;
step6：	计算左奇异向量矩阵$\boldsymbol{U}_{DC}=\boldsymbol{QU}_F$;
step7：	求$\boldsymbol{C}$的低秩近似矩阵$\boldsymbol{C}\backsimeq\boldsymbol{D}_C=(\boldsymbol{QU}_F)\boldsymbol{\Lambda}_F\boldsymbol{V}_F^T$ 。

其中，$k=tr$，$t\in(0,1)$。矩阵$\boldsymbol{R}^T$的随机 SVD 分解类似算法 3.2，低秩逼近矩阵记为$\boldsymbol{D}_R$。算法 3.2 计算矩阵$\boldsymbol{Z}$的时间复杂度为$O(m^2k)$，QR 分解的复杂度为$O(mk)$，计算矩阵$\boldsymbol{F}$的复杂度为$O(mk^2)$，SVD 分解的复杂度为$O(k^3)$。所以，该算法总的时间复杂度为$O(m^2k+k^3)$，为m的二次型。

3.2.4 矩阵 U 的近似逼近

根据上述随机 SVD 分解后得到的矩阵$\boldsymbol{D}_C$和$\boldsymbol{D}_R$，本节希望通过以下目标函数近似逼近矩阵$\boldsymbol{U}$，提高 CUR 矩阵分解精度的同时不增加计算复杂度。

$$\begin{aligned}\tilde{\boldsymbol{U}}&=\underbrace{argmin}_{\boldsymbol{U}}||\boldsymbol{K}-\boldsymbol{D}_C^T\boldsymbol{K}\boldsymbol{D}_R||_F^2+||\boldsymbol{CUR}-(\boldsymbol{D}_C^T\boldsymbol{C})\boldsymbol{U}(\boldsymbol{RD}_R)||_F^2\\&\geqslant\underbrace{argmin}_{\boldsymbol{U}}||\boldsymbol{D}_C^T\boldsymbol{K}\boldsymbol{D}_R-(\boldsymbol{D}_C^T\boldsymbol{C})\boldsymbol{U}(\boldsymbol{RD}_R)||_F^2\end{aligned}$$

通过迭代逼近，可以找到，

$$\tilde{\boldsymbol{U}}=(\boldsymbol{D}_C^T\boldsymbol{C})^+(\boldsymbol{D}_C^T\boldsymbol{K}\boldsymbol{D}_R)(\boldsymbol{RD}_R)^+ \tag{3.3}$$

证明：$||\boldsymbol{D}_C^T\boldsymbol{K}\boldsymbol{D}_R-(\boldsymbol{D}_C^T\boldsymbol{C})\boldsymbol{U}(\boldsymbol{RD}_R)||_F^2$

$$\leqslant||\boldsymbol{D}_C^T\boldsymbol{K}\boldsymbol{D}_R||_F^2-||(\boldsymbol{D}_C^T\boldsymbol{C})\boldsymbol{U}(\boldsymbol{RD}_R)||_F^2\leqslant||(\boldsymbol{D}_C^T\boldsymbol{C})\boldsymbol{U}(\boldsymbol{RD}_R)||_F^2$$

$$=Tr\left([(\boldsymbol{D}_C^T\boldsymbol{C})\boldsymbol{U}(\boldsymbol{RD}_R)]^T[(\boldsymbol{D}_C^T\boldsymbol{C})\boldsymbol{U}(\boldsymbol{RD}_R)]\right)$$

$$=Tr\left([(\boldsymbol{D}_C^T\boldsymbol{C})\boldsymbol{C}^+\boldsymbol{K}\boldsymbol{R}^+(\boldsymbol{RD}_R)]^T[(\boldsymbol{D}_C^T\boldsymbol{C})\boldsymbol{C}^+\boldsymbol{K}\boldsymbol{R}^+(\boldsymbol{RD}_R)]\right)$$

$$=Tr\left([\boldsymbol{D}_R^T\boldsymbol{R}^T(\boldsymbol{R}^+)^T\boldsymbol{K}^T(\boldsymbol{C}^+)^T\boldsymbol{C}^T\boldsymbol{D}_C\boldsymbol{D}_C^T\boldsymbol{C}\boldsymbol{C}^+\boldsymbol{K}\boldsymbol{R}^+\boldsymbol{RD}_R]\right)$$

令 $\boldsymbol{C}=\boldsymbol{U}_C\boldsymbol{\Sigma}_C\boldsymbol{V}_C^T$ ， $\boldsymbol{R}=\boldsymbol{U}_R\boldsymbol{\Sigma}_R\boldsymbol{V}_R^T$ ， $\boldsymbol{D}_C=\boldsymbol{U}_{DC}\boldsymbol{\Sigma}_{DC}\boldsymbol{V}_{DC}^T$ ，$\boldsymbol{D}_R=\boldsymbol{U}_{DR}\boldsymbol{\Sigma}_{DR}\boldsymbol{V}_{DR}^T$, $\boldsymbol{C}^+=\boldsymbol{V}_C\boldsymbol{\Sigma}_C^{-1}\boldsymbol{U}_C^T$， $\boldsymbol{R}^+=\boldsymbol{V}_R\boldsymbol{\Sigma}_R^{-1}\boldsymbol{U}_R^T$，且矩阵的迹具有性质①$Tr(\boldsymbol{K})=Tr(\boldsymbol{K}^T)$, ②$Tr(\boldsymbol{KM})=Tr(\boldsymbol{MK})$.

上式可以写为，

$$Tr\{(\boldsymbol{U}_{D_R}\boldsymbol{\Sigma}_{D_R}\boldsymbol{V}_{D_R})^T(\boldsymbol{U}_R\boldsymbol{\Sigma}_R\boldsymbol{V}_R)^T(\boldsymbol{V}_R\boldsymbol{\Sigma}_R^{-1}\boldsymbol{U}_R^T)^T\boldsymbol{K}^T(\boldsymbol{V}_C\boldsymbol{\Sigma}_C^{-1}\boldsymbol{U}_C^T)^T(\boldsymbol{U}_C\boldsymbol{\Sigma}_C\boldsymbol{V}_C^T)^T\boldsymbol{U}_{D_C}\boldsymbol{\Sigma}_{D_C}$$

$$\boldsymbol{V}_{D_C}^T(\boldsymbol{U}_{D_C}\boldsymbol{\Sigma}_{D_C}\boldsymbol{V}_{D_C}^T)^T\boldsymbol{U}_C\boldsymbol{\Sigma}_C\boldsymbol{V}_C^T\boldsymbol{V}_C\boldsymbol{\Sigma}_C^{-1}\boldsymbol{U}_C^T\boldsymbol{K}\boldsymbol{V}_R\boldsymbol{\Sigma}_R^{-1}\boldsymbol{U}_R^T\boldsymbol{U}_R\boldsymbol{\Sigma}_R\boldsymbol{V}_R^T\boldsymbol{U}_{D_R}\boldsymbol{\Sigma}_{DR}\boldsymbol{V}_{D_R}^T\}$$

$$=Tr\{\boldsymbol{V}_{D_R}^T\boldsymbol{\Sigma}_{D_R}^T\boldsymbol{V}_{D_R}^T\boldsymbol{D}_R(\boldsymbol{R}^+)^T\boldsymbol{R}^+\boldsymbol{R}(\boldsymbol{C}^+)^T\boldsymbol{C}^T\boldsymbol{C}\boldsymbol{C}^+\boldsymbol{D}_C\boldsymbol{D}_C^T\boldsymbol{K}^T\boldsymbol{K}\}$$

$$\geqslant Tr\{\boldsymbol{D}_C^T(\boldsymbol{D}_C^T)^+\boldsymbol{C}^+\boldsymbol{K}\boldsymbol{D}_R\boldsymbol{D}_R^+\boldsymbol{R}^+\}=||(\boldsymbol{D}_C^T\boldsymbol{C})^+(\boldsymbol{D}_C^T\boldsymbol{K}\boldsymbol{D}_R)(\boldsymbol{R}\boldsymbol{D}_R)^+||_F^2$$

即，$||\boldsymbol{D}_C^T\boldsymbol{K}\boldsymbol{D}_R-(\boldsymbol{D}_C^T\boldsymbol{C})\boldsymbol{U}(\boldsymbol{R}\boldsymbol{D}_R)||_F^2\geqslant||(\boldsymbol{D}_C^T\boldsymbol{C})^+(\boldsymbol{D}_C^T\boldsymbol{K}\boldsymbol{D}_R)(\boldsymbol{R}\boldsymbol{D}_R)^+||_F^2$

则，$\tilde{\boldsymbol{U}}=(\boldsymbol{D}_C^T\boldsymbol{C})^+(\boldsymbol{D}_C^T\boldsymbol{K}\boldsymbol{D}_R)(\boldsymbol{R}\boldsymbol{D}_R)^+$。

3.2.5 基于不等概自适应抽样和随机 SVD 分解的 CUR 矩阵低秩重构

经过对原矩阵抽样，得到可表示原矩阵信息的子矩阵$\boldsymbol{C}$和$\boldsymbol{R}$，及通过相关运算得到联合矩阵$\boldsymbol{U}$后，就可根据$\hat{\boldsymbol{K}}=\boldsymbol{CUR}$对高维矩阵进行低秩逼近重构，重构后的矩阵表达式见式(3.4)。

$$\tilde{\boldsymbol{K}}=\boldsymbol{CUR}=\boldsymbol{C}(\boldsymbol{D}_C^T\boldsymbol{C})^+(\boldsymbol{D}_C^T\boldsymbol{K}\boldsymbol{D}_R)(\boldsymbol{R}\boldsymbol{D}_R)^+\boldsymbol{R} \tag{3.4}$$

为了简便运算，避免抽取到重复的样本列或行，本文的讨论主要基于不放回抽样。具体的抽样分解重构过程如算法 3.3 所示。

算法 3.3	基于自适应抽样的 CUR 矩阵分解重构
输入：	矩阵$\boldsymbol{K}\in\mathbb{R}^{m\times n}$，目标低秩$k$，迭代参数$a$。
输出：	低秩重构矩阵$\tilde{\boldsymbol{K}}$ ，重构误差ϵ。
step1：	根据算法 3.1 得到矩阵$\boldsymbol{C}$、$\boldsymbol{R}$;
step2：	根据算法 3.2 对矩阵$\boldsymbol{C}$、$\boldsymbol{R}^T$进行随机 SVD 分解得到$\boldsymbol{D}_C$和$\boldsymbol{D}_R$;
step3：	进行矩阵重构，$\tilde{\boldsymbol{K}}=\boldsymbol{CUR}=\boldsymbol{C}(\boldsymbol{D}_C^T\boldsymbol{C})^+(\boldsymbol{D}_C^T\boldsymbol{K}\boldsymbol{D}_R)(\boldsymbol{R}\boldsymbol{D}_R)^+\boldsymbol{R}$;
step4：	计算矩阵重构相对误差$\epsilon=\frac{\|\boldsymbol{K}-\tilde{\boldsymbol{K}}\|_F}{\|\boldsymbol{K}\|_F}$。

3.3 误差界分析

为了研究本节提出的 CUR 矩阵分解低秩重构方法的误差界，首先引出引理 3.1 和引理 3.2。

引理 3.1(Wang,2013)**:** 令$\boldsymbol{K}$表示任意给定的$m \times n$矩阵，低秩参数k是小于m和n的任意正实数，$\epsilon \in (0,1)$为误差参数，令$\boldsymbol{C} \in \mathbb{R}^{m\times c}$和$\boldsymbol{R} \in \mathbb{R}^{m\times r}$表示由$\boldsymbol{K}$的部分列和部分行构成的矩阵，$c$和$r$都大于$4k\epsilon^{-1}(1+o(1))$，则有以下不等式成立，

$$\mathbb{E}||\boldsymbol{K}-\boldsymbol{C}\boldsymbol{C}^{+}\boldsymbol{K}\boldsymbol{R}^{+}\boldsymbol{R}||_F^2 \leqslant (1+\epsilon)||\boldsymbol{K}-\boldsymbol{A}\boldsymbol{K}_k||_F^2 \tag{3.5}$$

引理 3.2 (Li M,2010)：给定矩阵$\boldsymbol{W} \in \mathbb{R}^{m\times m}$，利用随机 SVD 分解生成随机高斯矩阵对$\boldsymbol{W}$进行低秩处理，$||\boldsymbol{W}-\boldsymbol{Q}\boldsymbol{Q}^T\boldsymbol{W}||_F$的期望为

$$\mathbb{E}_Q||\boldsymbol{W}-\boldsymbol{Q}\boldsymbol{Q}^T\boldsymbol{W}||_F \leqslant (1+k/4)^{1/2}(\sum_{i>k}\sigma_i^2(\boldsymbol{W}))^{1/2} \tag{3.6}$$

其中，$\sigma_i(\boldsymbol{W})$表示$\boldsymbol{W}$的奇异值，$k \leqslant i \leqslant m$。

定理 3.1：对于任意给定的矩阵$\boldsymbol{K} \in \mathbb{R}^{m\times n}$，从中抽取部分列和行构造矩阵$\boldsymbol{C} \in \mathbb{R}^{m\times c}$和$\boldsymbol{R} \in \mathbb{R}^{m\times r}$，构造$\boldsymbol{U}^* = \boldsymbol{C}^{+}\boldsymbol{K}\boldsymbol{R}^{+}$；再利用随机 SVD 分解算法对两子矩阵进行降秩处理得到矩阵$\boldsymbol{D}_C$和$\boldsymbol{D}_R$，构造$\tilde{\boldsymbol{U}} = (\boldsymbol{D}_C^T\boldsymbol{C})^{+}(\boldsymbol{D}_C^T\boldsymbol{K}\boldsymbol{D}_R)(\boldsymbol{R}\boldsymbol{D}_R)^{+}$，则$\boldsymbol{C}\boldsymbol{U}^*\boldsymbol{R}-\boldsymbol{C}\tilde{\boldsymbol{U}}\boldsymbol{R}$关于 F 范数存在以下不等式：

$$||\boldsymbol{C}\boldsymbol{U}^*\boldsymbol{R}-\boldsymbol{C}\tilde{\boldsymbol{U}}\boldsymbol{R}||_F^2 \leqslant ||\boldsymbol{Q}\boldsymbol{Q}^T\boldsymbol{K}-\boldsymbol{K}||_F^2 \tag{3.7}$$

证明：矩阵$\boldsymbol{C}$和矩阵$\boldsymbol{R}$的 SVD 分解可以表示为

$$\boldsymbol{C} = \boldsymbol{U}_C\boldsymbol{\Sigma}_C\boldsymbol{V}_C^T,\ \boldsymbol{R} = \boldsymbol{U}_R\boldsymbol{\Sigma}_R\boldsymbol{V}_R^T,$$

随机 SVD 分解表示为

$$\boldsymbol{C} \backsimeq \boldsymbol{D}_C = (\boldsymbol{Q}_1\boldsymbol{U}_{F1})\boldsymbol{\Lambda}_{F1}\boldsymbol{V}_{F1}^T,\ \boldsymbol{C} \backsimeq \boldsymbol{D}_C = (\boldsymbol{Q}_1\boldsymbol{U}_{F1})\boldsymbol{\Lambda}_{F1}\boldsymbol{V}_{F1}^T$$

则$\boldsymbol{C}\boldsymbol{U}^*\boldsymbol{R} = \boldsymbol{C}(\boldsymbol{C}^{+}\boldsymbol{K}\boldsymbol{R}^{+})\boldsymbol{R} = \boldsymbol{U}_C\boldsymbol{U}_C^T\boldsymbol{K}\boldsymbol{V}_R\boldsymbol{V}_R^T$

令$\boldsymbol{Z}_1 = \boldsymbol{U}_C^T\boldsymbol{K}\boldsymbol{V}_R$，则，$\boldsymbol{C}\boldsymbol{U}^*\boldsymbol{R} = \boldsymbol{U}_C\boldsymbol{Z}_1\boldsymbol{V}_R^T$

$$\begin{aligned}
\tilde{\boldsymbol{U}} &= (\boldsymbol{D}_C^T\boldsymbol{C})^{+}(\boldsymbol{D}_C^T\boldsymbol{K}\boldsymbol{D}_R)(\boldsymbol{R}\boldsymbol{D}_R)^{+} \\
&= (\boldsymbol{D}_C^T\boldsymbol{U}_C\boldsymbol{\Sigma}_C\boldsymbol{V}_C^T)^{+}(\boldsymbol{D}_C^T\boldsymbol{K}\boldsymbol{D}_R)(\boldsymbol{U}_R\boldsymbol{\Sigma}_R\boldsymbol{V}_R^T\boldsymbol{D}_R)^{+} \\
&= (\boldsymbol{\Sigma}_C\boldsymbol{V}_C^T)^{+}(\boldsymbol{D}_C^T\boldsymbol{U}_C)^{+}(\boldsymbol{D}_C^T\boldsymbol{K}\boldsymbol{D}_R)(\boldsymbol{V}_R^T\boldsymbol{D}_R)^{+}(\boldsymbol{U}_R\boldsymbol{\Sigma}_R)^{+}
\end{aligned}$$

令$\boldsymbol{Z}_2=(\boldsymbol{D}_C^T\boldsymbol{U}_C)^+(\boldsymbol{D}_C^T\boldsymbol{K}\boldsymbol{D}_R)(\boldsymbol{V}_R^T\boldsymbol{D}_R)^+$, 则，

$$\tilde{\boldsymbol{U}}=(\boldsymbol{\Sigma}_C\boldsymbol{V}_C^T)^+\boldsymbol{Z}_2(\boldsymbol{U}_R\boldsymbol{\Sigma}_R)^+,$$

$$\boldsymbol{C}\tilde{\boldsymbol{U}}\boldsymbol{R}=\boldsymbol{U}_C\boldsymbol{\Sigma}_C\boldsymbol{V}_C^T(\boldsymbol{\Sigma}_C\boldsymbol{V}_C^T)^+\boldsymbol{Z}_2(\boldsymbol{U}_R\boldsymbol{\Sigma}_R)^+\boldsymbol{U}_R\boldsymbol{\Sigma}_R\boldsymbol{V}_R^T=\boldsymbol{U}_C\boldsymbol{Z}_2\boldsymbol{V}_R^T$$

即，$\boldsymbol{C}\boldsymbol{U}^*\boldsymbol{R}=\boldsymbol{U}_C\boldsymbol{Z}_1\boldsymbol{V}_R^T$，$\boldsymbol{C}\tilde{\boldsymbol{U}}\boldsymbol{R}=\boldsymbol{U}_C\boldsymbol{Z}_2\boldsymbol{V}_R^T$

$$||\boldsymbol{K}-\boldsymbol{C}\tilde{\boldsymbol{U}}\boldsymbol{R}||_F^2$$
$$=||\boldsymbol{K}-\boldsymbol{U}_C\boldsymbol{Z}_2\boldsymbol{V}_R^T||_F^2=||\boldsymbol{K}-\boldsymbol{U}_C\boldsymbol{Z}_1\boldsymbol{V}_R^T+\boldsymbol{U}_C\boldsymbol{Z}_1\boldsymbol{V}_R^T-\boldsymbol{U}_C\boldsymbol{Z}_2\boldsymbol{V}_R^T||_F^2$$
$$=||(\boldsymbol{K}-\boldsymbol{U}_C\boldsymbol{Z}_1\boldsymbol{V}_R^T)+\boldsymbol{U}_C(\boldsymbol{Z}_1-\boldsymbol{Z}_2)\boldsymbol{V}_R^T||_F^2$$
$$=||\boldsymbol{K}-\boldsymbol{U}_C(\boldsymbol{U}_C^T\boldsymbol{K}\boldsymbol{V}_R)\boldsymbol{V}_R^T+\boldsymbol{U}_C(\boldsymbol{Z}_1-\boldsymbol{Z}_2)\boldsymbol{V}_R^T||_F^2$$
$$=||\boldsymbol{K}-(\boldsymbol{U}_C\boldsymbol{U}_C^T\boldsymbol{K})\boldsymbol{V}_R\boldsymbol{V}_R^T+\boldsymbol{U}_C(\boldsymbol{Z}_1-\boldsymbol{Z}_2)\boldsymbol{V}_R^T||_F^2$$
$$=||(\boldsymbol{I}-\boldsymbol{U}_C\boldsymbol{U}_C^T)\boldsymbol{K}+\boldsymbol{U}_C\boldsymbol{U}_C^T\boldsymbol{K}(\boldsymbol{I}-\boldsymbol{V}_R\boldsymbol{V}_R^T)+\boldsymbol{U}_C(\boldsymbol{Z}_1-\boldsymbol{Z}_2)\boldsymbol{V}_R^T||_F^2$$
$$\leqslant||(\boldsymbol{I}-\boldsymbol{U}_C\boldsymbol{U}_C^T)\boldsymbol{K}||_F^2+||\boldsymbol{U}_C\boldsymbol{U}_C^T\boldsymbol{K}(\boldsymbol{I}-\boldsymbol{V}_R\boldsymbol{V}_R^T)+\boldsymbol{U}_C(\boldsymbol{Z}_1-\boldsymbol{Z}_2)\boldsymbol{V}_R^T||_F^2$$
$$\leqslant||(\boldsymbol{I}-\boldsymbol{U}_C\boldsymbol{U}_C^T)\boldsymbol{K}||_F^2+||\boldsymbol{U}_C\boldsymbol{U}_C^T\boldsymbol{K}(\boldsymbol{I}-\boldsymbol{V}_R\boldsymbol{V}_R^T)||_F^2+||\boldsymbol{U}_C(\boldsymbol{Z}_1-\boldsymbol{Z}_2)\boldsymbol{V}_R^T||_F^2$$
$$=||(\boldsymbol{I}-\boldsymbol{U}_C\boldsymbol{U}_C^T)\boldsymbol{K}+\boldsymbol{U}_C\boldsymbol{U}_C^T\boldsymbol{K}(\boldsymbol{I}-\boldsymbol{V}_R\boldsymbol{V}_R^T)||_F^2+||\boldsymbol{U}_C(\boldsymbol{Z}_1-\boldsymbol{Z}_2)\boldsymbol{V}_R^T||_F^2$$
$$=||\boldsymbol{K}-\boldsymbol{U}_C\boldsymbol{U}_C^T\boldsymbol{K}\boldsymbol{V}_R\boldsymbol{V}_R^T||_F^2+||\boldsymbol{U}_C(\boldsymbol{Z}_1-\boldsymbol{Z}_2)\boldsymbol{V}_R^T||_F^2$$
$$=||(\boldsymbol{K}-\boldsymbol{U}_C\boldsymbol{Z}_1\boldsymbol{V}_R^T)||_F^2+||\boldsymbol{U}_C(\boldsymbol{Z}_1-\boldsymbol{Z}_2)\boldsymbol{V}_R^T||_F^2$$
$$=||(\boldsymbol{K}-\boldsymbol{C}\boldsymbol{U}^*\boldsymbol{R})||_F^2+||\boldsymbol{U}_C(\boldsymbol{Z}_1-\boldsymbol{Z}_2)\boldsymbol{V}_R^T||_F^2$$

而

$$||\boldsymbol{U}_C(\boldsymbol{Z}_1-\boldsymbol{Z}_2)\boldsymbol{V}_R^T||_F^2=||\boldsymbol{U}_C\boldsymbol{Z}_1\boldsymbol{V}_R^T-\boldsymbol{U}_C\boldsymbol{Z}_2\boldsymbol{V}_R^T||_F^2=||\boldsymbol{C}\boldsymbol{U}^*\boldsymbol{R}-\boldsymbol{C}\tilde{\boldsymbol{U}}\boldsymbol{R}||_F^2$$

则，

$$||\boldsymbol{K}-\boldsymbol{C}\tilde{\boldsymbol{U}}\boldsymbol{R}||_F^2\leqslant||(\boldsymbol{K}-\boldsymbol{C}\boldsymbol{U}^*\boldsymbol{R})||_F^2+||\boldsymbol{C}\boldsymbol{U}^*\boldsymbol{R}-\boldsymbol{C}\tilde{\boldsymbol{U}}\boldsymbol{R}||_F^2$$

根据引理 3.1，可得

$$||\boldsymbol{K}-\boldsymbol{C}\tilde{\boldsymbol{U}}\boldsymbol{R}||_F^2\leqslant||(\boldsymbol{K}-\boldsymbol{C}\boldsymbol{U}^*\boldsymbol{R})||_F^2+||\boldsymbol{C}\boldsymbol{U}^*\boldsymbol{R}-\boldsymbol{C}\tilde{\boldsymbol{U}}\boldsymbol{R}||_F^2$$
$$\leqslant(1+\epsilon)||\boldsymbol{K}-\boldsymbol{K}_k||_F^2+||\boldsymbol{C}\boldsymbol{U}^*\boldsymbol{R}-\boldsymbol{C}\tilde{\boldsymbol{U}}\boldsymbol{R}||_F^2$$

由$\boldsymbol{A}^+=(\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T$可得(Hoyle,2010)

$$\boldsymbol{C}\boldsymbol{U}^*\boldsymbol{R}-\boldsymbol{C}\tilde{\boldsymbol{U}}\boldsymbol{R}$$
$$=\boldsymbol{U}_C\boldsymbol{U}_C^T\boldsymbol{K}\boldsymbol{V}_R\boldsymbol{V}_R^T-\boldsymbol{U}_C(\boldsymbol{D}_C^T\boldsymbol{U}_C)^+(\boldsymbol{D}_C^T\boldsymbol{K}\boldsymbol{D}_R)(\boldsymbol{V}_R^T\boldsymbol{D}_R)^+\boldsymbol{V}_R^T$$
$$=\boldsymbol{U}_C\boldsymbol{U}_C^T\boldsymbol{K}\boldsymbol{V}_R\boldsymbol{V}_R^T-\boldsymbol{U}_C(\boldsymbol{D}_C^T\boldsymbol{U}_C)^+(\boldsymbol{D}_C^T\boldsymbol{K}\boldsymbol{D}_R)(\boldsymbol{V}_R^T\boldsymbol{D}_R)^+\boldsymbol{V}_R^T$$

$$
\begin{aligned}
&= \boldsymbol{U}_C\boldsymbol{U}_C^T\boldsymbol{K}\boldsymbol{V}_R\boldsymbol{V}_R^T - \boldsymbol{U}_C(\boldsymbol{V}_{\boldsymbol{F1}}\boldsymbol{\Lambda}_{F1}^T\boldsymbol{U}_{F1}^T\boldsymbol{Q}_1^T\boldsymbol{U}_C)^+ \\
&\quad (\boldsymbol{V}_{F1}\boldsymbol{\Lambda}_{F1}^T\boldsymbol{U}_{F1}^T\boldsymbol{Q}_1^T\boldsymbol{K}\boldsymbol{Q}_2\boldsymbol{U}_{F2}\boldsymbol{\Lambda}_{F2}\boldsymbol{V}_{F2}^T)(\boldsymbol{V}_R^T\boldsymbol{Q}_2\boldsymbol{U}_{F2}\boldsymbol{\Lambda}_{F2}\boldsymbol{V}_{F2}^T)^+\boldsymbol{V}_R^T) \\
&= \boldsymbol{U}_C\boldsymbol{U}_C^T\boldsymbol{K}\boldsymbol{V}_R\boldsymbol{V}_R^T - \boldsymbol{U}_C\boldsymbol{U}_C^+(\boldsymbol{Q}_1^T)^+(\boldsymbol{U}_{F1}^T)^+(\boldsymbol{\Lambda}_{F1}^T)^+\boldsymbol{V}_{F1}^+\boldsymbol{V}_{F1}\boldsymbol{\Lambda}_{F1}^T \\
&\quad \boldsymbol{U}_{F1}^T\boldsymbol{Q}_1^T\boldsymbol{K}\boldsymbol{Q}_2\boldsymbol{U}_{F2}\boldsymbol{\Lambda}_{F2}\boldsymbol{V}_{F2}^T(\boldsymbol{V}_{F2}^T)^+\boldsymbol{\Lambda}_{F2}^+\boldsymbol{U}_{F2}^+\boldsymbol{Q}_2^+(\boldsymbol{V}_R^T)^+\boldsymbol{V}_R^T \\
&= \boldsymbol{U}_C\boldsymbol{U}_C^T\boldsymbol{K}\boldsymbol{V}_R\boldsymbol{V}_R^T - \boldsymbol{U}_C[(\boldsymbol{U}_C^T\boldsymbol{U}_C)^{-1}\boldsymbol{U}_C^T](\boldsymbol{Q}_1^+)^T(\boldsymbol{U}_{F1}^+)^T(\boldsymbol{\Lambda}_{F1}^+)^T\boldsymbol{V}_{F1}\boldsymbol{\Lambda}_{F1}^T \\
&\quad \boldsymbol{U}_{F1}^T\boldsymbol{Q}_1^T\boldsymbol{K}\boldsymbol{Q}_2\boldsymbol{U}_{F2}\boldsymbol{\Lambda}_{F2}\boldsymbol{V}_{F2}^T(\boldsymbol{V}_{F2}^+)^T\boldsymbol{\Lambda}_{F2}^+\boldsymbol{U}_{F2}^+\boldsymbol{Q}_2^+(\boldsymbol{V}_R^+)^T\boldsymbol{V}_R^T \\
&= \boldsymbol{U}_C\boldsymbol{U}_C^T\boldsymbol{K}\boldsymbol{V}_R\boldsymbol{V}_R^T - [\boldsymbol{Q}_1^{-1}(\boldsymbol{Q}_1^T)^{-1}\boldsymbol{Q}_1^T]^T[\boldsymbol{U}_{F1}^{-1}(\boldsymbol{U}_{F1}^T)^{-1}\boldsymbol{U}_{F1}^T]^T \\
&[\boldsymbol{\Lambda}_{F1}^{-1}(\Lambda_{F1}^T)^{-1}(\Lambda_{F1})^T]^T\boldsymbol{V}_{F1}\boldsymbol{\Lambda}_{F1}^T\boldsymbol{U}_{F1}^T\boldsymbol{Q}_1^T\boldsymbol{K}\boldsymbol{Q}_2\boldsymbol{U}_{F2}\boldsymbol{\Lambda}_{F2}\boldsymbol{V}_{F2}^T[\ \boldsymbol{V}_{F1}^{-1}(\boldsymbol{V}_{F2}^T)^{-1}\boldsymbol{V}_{F2}^T]^T \\
&[\ \boldsymbol{\Lambda}_{F2}^{-1}(\boldsymbol{\Lambda}_{F2}^T)^{-1}\boldsymbol{\Lambda}_{F2}^T][\boldsymbol{U}_{F2}^{-1}(\boldsymbol{U}_{F2}^T)^{-1}\boldsymbol{U}_{F2}^T][\boldsymbol{Q}_2^{-1}(\boldsymbol{Q}_2^T)^{-1}\boldsymbol{Q}_2^T][\boldsymbol{V}_R^{-1}(\boldsymbol{V}_R^T)^{-1}\boldsymbol{V}_R^T]^T\boldsymbol{V}_R^T \\
&= \boldsymbol{U}_C\boldsymbol{U}_C^T\boldsymbol{K}\boldsymbol{V}_R\boldsymbol{V}_R^T - [(\boldsymbol{Q}_1\boldsymbol{U}_{F1}\boldsymbol{\Lambda}_{F1})^T]^{-1}\boldsymbol{V}_{F1}(\boldsymbol{Q}_1\boldsymbol{U}_{F1}\boldsymbol{\Lambda}_{F1})^T\boldsymbol{K} \\
&\leqslant \boldsymbol{K} - \boldsymbol{Q}\boldsymbol{Q}^T\boldsymbol{K}
\end{aligned}
$$

所以，$||\boldsymbol{C}\boldsymbol{U}^*\boldsymbol{R} - \boldsymbol{C}\tilde{\boldsymbol{U}}\boldsymbol{R}||_F^2 \leqslant ||\boldsymbol{K} - \boldsymbol{Q}\boldsymbol{Q}^T\boldsymbol{K}||_F^2$

定理 3.1 得证。

通过定理 3.1 可以看出，当c和r充分大时，对高维矩阵$\boldsymbol{K}$进行 CUR 分解重构后的低秩逼近误差接近于对$\boldsymbol{K}$直接进行随机 SVD 分解的低秩逼近误差；

定理 3.2：对于任意给定的矩阵$\boldsymbol{K} \in \mathbb{R}^{m\times n}$，从中抽取部分列和行构造矩阵$\boldsymbol{C} \in \mathbb{R}^{m\times c}$和$\boldsymbol{R} \in \mathbb{R}^{m\times r}$，利用随机 SVD 分解算法对两子矩阵进行降秩处理得到矩阵$\boldsymbol{D}_C$和$\boldsymbol{D}_R$，并以此计算矩阵$\tilde{\boldsymbol{U}} = (\boldsymbol{D}_C^T\boldsymbol{C})^+(\boldsymbol{D}_C^T\boldsymbol{K}\boldsymbol{D}_R)(\boldsymbol{R}\boldsymbol{D}_R)^+$，再根据$\tilde{\boldsymbol{K}} = \boldsymbol{C}\tilde{\boldsymbol{U}}\boldsymbol{R}$对矩阵$\boldsymbol{K}$进行分解重构，重构后的矩阵关于$F$范数的低秩逼近误差存在以下不等式：

$$||\boldsymbol{K} - \boldsymbol{C}\tilde{\boldsymbol{U}}\boldsymbol{R}||_F^2 \leqslant (1+\epsilon)||\boldsymbol{K} - \boldsymbol{K}_k||_F^2 - \sum_{i>k}\sigma_i^2(\boldsymbol{K}) \tag{3.8}$$

证明：根据引理 3.2 和定理 3.1，可得，

$$
\begin{aligned}
||\boldsymbol{K} - \boldsymbol{C}\tilde{\boldsymbol{U}}\boldsymbol{R}||_F^2 &\leqslant ||(\boldsymbol{K} - \boldsymbol{C}\boldsymbol{U}^*\boldsymbol{R})||_F^2 + ||\boldsymbol{C}\boldsymbol{U}^*\boldsymbol{R} - \boldsymbol{C}\tilde{\boldsymbol{U}}\boldsymbol{R}||_F^2 \\
&\leqslant (1+\epsilon)||\boldsymbol{K} - \boldsymbol{K}_K||_F^2 + ||\boldsymbol{Q}\boldsymbol{Q}^T\boldsymbol{K} - \boldsymbol{K}||_F^2 \\
&\leqslant (1+\epsilon)||\boldsymbol{K} - \boldsymbol{K}_K||_F^2 - \sum_{i>k}\sigma_i^2(\boldsymbol{K})
\end{aligned}
$$

证毕。

定理 3.2 显示，当$\boldsymbol{D}_C$的列数和$\boldsymbol{D}_R$的行数充分大时，对高维矩阵$\boldsymbol{K}$进行基于不等概自适应抽样和随机SVD分解的CUR矩阵分解重构精度几乎和$\boldsymbol{K}$的最佳低秩逼近精度一样好。

3.4 计算复杂度分析

本节中，m和n表示原始矩阵的行数和列数，c和r表示子矩阵的列数和行数，d_c和d_r表示矩阵$\boldsymbol{D}_C$和$\boldsymbol{D}_R$的列数，k表示所期望的低秩。在传统 CUR 矩阵分解算法中，计算伪逆矩阵$\boldsymbol{C}^+$和$\boldsymbol{R}^+$的时间复杂度为$O(mc^2+nr^2)$，矩阵相乘的时间复杂度为$O(mn\cdot min\{c,r\})$。因此，即使使用均匀抽样从矩阵$\boldsymbol{K}$中抽样构造矩阵$\boldsymbol{C}$和$\boldsymbol{R}$，不需要遍历原矩阵列和行时，计算$\hat{\boldsymbol{K}}=\boldsymbol{C}\boldsymbol{U}^*\boldsymbol{R}=\boldsymbol{C}(\boldsymbol{C}^+\boldsymbol{K}\boldsymbol{R}^+)\boldsymbol{R}$的时间复杂度也至少为$O(kmn+mn\cdot min\{c,r\})$，$O(kmn)$为均匀抽样时间复杂度。

不等概抽样的时间复杂度为$O(min\{cn,rm\}\cdot k^2)$，矩阵相乘的复杂度为$O(mn\cdot min\{c,r\})$，则基于不等概抽样计算矩阵$\hat{\boldsymbol{K}}$的时间复杂度为$O(min\{cn,rm\}k^2+mn\cdot min\{c,r\})$。

本节所应用的自适应抽样算法中，计算$\boldsymbol{R}^+$的时间复杂度$O(nr^2)$，矩阵相乘时间复杂度为$O(mn\cdot min\{c,r\})$，计算$\hat{\boldsymbol{K}}$的时间复杂度为$O(nr^2+mn\cdot min\{c,r\})$。

将三种抽样方法的时间复杂度进行对比，具体见表 3.1。

表 3.1 三种抽样方法时间复杂度对比

CUR 矩阵分解的抽样	计算复杂度
均匀抽样	$O(kmn+mn\cdot min\{c,r\})$
不等概抽样	$O(min\{cn,rm\}k^2+mn\cdot min\{c,r\})$
本文提出的方法	$O(nr^2+mn\cdot min\{c,r\})$

如表 3.1 所示，因为$c<n,r<m$，$k\leqslant min\{c,r\}$，所以$nr^2\leqslant kmn$，$nr^2\leqslant min\{cn,rm\}k^2$，相比其他两种抽样方法，本节提出的不等概自适应抽样的计算复杂度较低。

经过抽样得到子矩阵后，需要对子矩阵进行随机 SVD 分解，矩阵$\boldsymbol{C}$和矩阵$\boldsymbol{R}^T$进行随机 SVD 分解的时间复杂度分别为$O(m^2k+k^3)$、$O(n^2k+k^3)$，计算$\boldsymbol{D}_C^T\boldsymbol{C}$、$\boldsymbol{D}_C^T\boldsymbol{K}\boldsymbol{D}_R$、$\boldsymbol{R}\boldsymbol{S}_R$的时间复杂度分别为$O(mcd_c)$、$O(mn\cdot min\{d_c,d_r\})$和$O(nrd_r)$，$O(d_rr^2+d_cc^2+d_cd_r\cdot min\{c,r\})$表示计算伪逆矩阵$(\boldsymbol{D}_C^T\boldsymbol{C})^+$、$(\boldsymbol{R}\boldsymbol{S}_R)^+$的时间复杂度，

则计算矩阵$\tilde{\boldsymbol{U}}$的时间复杂度为$O(d_r r^2 + d_c c^2 + d_c d_r \cdot min\{c,r\})$；因此，本文提出的基于不等概自适应抽样和随机 SVD 分解的$\tilde{\boldsymbol{K}} = \boldsymbol{C}\tilde{\boldsymbol{U}}\boldsymbol{R}$矩阵分解逼近的时间复杂度为 $O(d_r r^2 + d_c c^2 + d_c d_r \cdot min\{c,r\})$，而基于不等概自适应抽样，利用 $\hat{\boldsymbol{K}} = \boldsymbol{C}\boldsymbol{U}^*\boldsymbol{R} = \boldsymbol{C}(\boldsymbol{C}^+\boldsymbol{K}\boldsymbol{R}^+)\boldsymbol{R}$ 进行 CUR 矩阵分解的复杂度为 $O(nr^2 + mn \cdot min\{c,r\})$，两种矩阵低秩逼近方法的时间复杂度对比如表 3.2 所示。

表 3.2 两种 CUR 矩阵分解方法复杂度对比

矩阵分解重构方法	时间复杂度
$\hat{\boldsymbol{K}} = \boldsymbol{C}\boldsymbol{U}^*\boldsymbol{R} = \boldsymbol{C}(\boldsymbol{C}^+\boldsymbol{K}\boldsymbol{R}^+)\boldsymbol{R}$	$O(nr^2 + mn \cdot min\{c,r\})$
本文提出的方法	$O(d_r r^2 + d_c c^2 + d_c d_r \cdot min\{c,r\})$

如表 3.2 所示，$d_c < c < n$，$d_r < r < m$，$d_r r^2 < nr^2$，但mn与$d_c c^2 + d_c d_r$没有明显的大小关系，只能说明本文提出的方法的时间复杂度不会明显高于前一种方法，随着参数的变换调整，两种方法的时间复杂度会出现相交点。

3.5 数值检验

3.5.1 模拟数据生成

本节首先通过四组随机模拟数据对不等概自适应抽样方法的有效性进行验证，数据大小都为 120000，分别记为 data1、data2、data3 和 data4，为了说明不同分布的数据对方法精度的影响，令 data1、data2 服从伽马分布，data3、data4 服从正态分布。

进一步为了说明服从同分布，但参数不同的数据对方法精度带来的影响，令 data1 服从形状参数$\alpha = 1$，位置参数$\beta = 1$的伽马分布；data2 服从形状参数$\alpha = 10$，位置参数$\beta = 1$的伽马分布；data3 服从均值$\mu = 0$，标准差$\sigma = 1$的正态分布；data4 服从均值$\mu = 0$，标准差$\sigma = 10$的正态分布。再将四组随机数转化为矩阵形式，分别表示为$\boldsymbol{K}_1 \in \mathbb{R}^{400\times300}$、$\boldsymbol{K}_2 \in \mathbb{R}^{400\times300}$、$\boldsymbol{K}_3 \in \mathbb{R}^{400\times300}$、$\boldsymbol{K}_4 \in \mathbb{R}^{400\times300}$，此时矩阵的秩都为 300。

3.5.2 抽样方法模拟

1.方法模拟与精度比较

本节通过矩阵重构误差对抽样方法的优良性进行模拟比较，为了比较结果的精确性，这里先不采用本文提出的关于矩阵$\boldsymbol{U}$的方法，而是令$\boldsymbol{U}=\boldsymbol{C}^{+}\boldsymbol{K}\boldsymbol{R}^{+}$，重构矩阵$\hat{\boldsymbol{K}}=\boldsymbol{C}\boldsymbol{U}\boldsymbol{R}=\boldsymbol{C}(\boldsymbol{C}^{+}\boldsymbol{K}\boldsymbol{R}^{+})\boldsymbol{R}$，分别比较基于不等概自适应抽样、均匀抽样和不等概抽样的 CUR 矩阵分解重构误差，具体见表 3.3，误差越小，抽样方法精度越高。

表 3.3 随机模拟数据低秩逼近相对误差

目标秩 k	数据集	抽样方法		
		不等概自适应抽样	不等概抽样	均匀抽样
10	data1	**0.5506**	0.5167	0.5085
	data2	**0.0926**	0.0927	0.0931
	data3	**0.9436**	0.9462	0.9503
	data4	**0.9434**	0.9519	0.9502
20	data1	**0.4224**	0.4326	0.4339
	data2	**0.0784**	0.0792	0.0792
	data3	**0.8265**	0.8341	0.8406
	data4	**0.8253**	0.8435	0.8407
30	data1	**0.3510**	0.3636	0.3645
	data2	**0.0656**	0.0665	0.0664
	data3	**0.6954**	0.7177	0.7139
	data4	**0.6948**	0.7128	0.7127
50	data1	**0.2374**	0.2486	0.2506
	data2	**0.0448**	0.0449	0.0457
	data3	**0.4758**	0.4912	0.4946
	data4	**0.4756**	0.5020	0.4943

通过 20 次抽样的误差均值来看，对于 data1，当$k=10$时，不等概自适应抽样的误差均值高于不等概抽样和均匀抽样；在其他秩时，不等概自适应抽样误差均值都低于其他两种抽样。此外可以看出，随着秩的增加，抽样的误差均值都呈下降趋势；当秩$k=50$时，不等概自适应抽样的误差均值仅为 23.74%。

data2 相比 data1，两组数据虽服从同一分布，但形状参数不同；可以看出，随着伽马分布形状参数的增加，三种抽样方法的误差均值都有较大的降低。随着

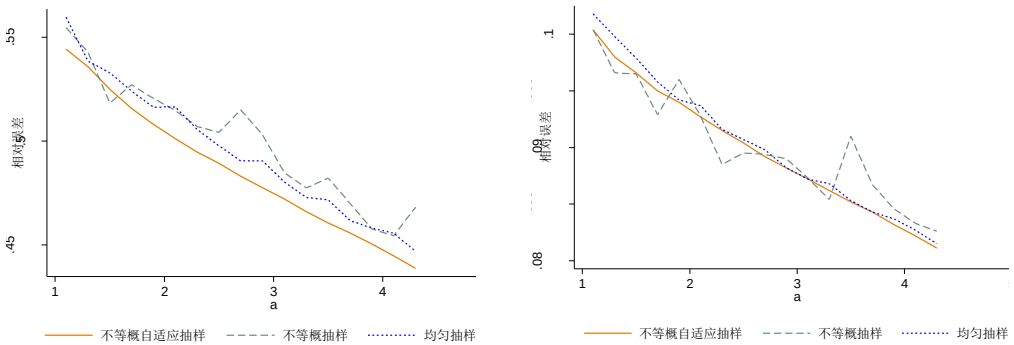
秩的增加，抽样的误差均值呈降低趋势，但在所有秩下，三种抽样的误差均值较接近，比如$k=50$时，不等概自适应抽样的误差均值为 4.48%，不等概抽样的误差均值为 4.49%，均匀抽样的误差均值为 4.57%。

对于 data3，在任何秩下，不等概自适应抽样的误差均值皆低于其他两种抽样，且随着秩的增加，抽样误差均值呈降低趋势，降低幅度明显。

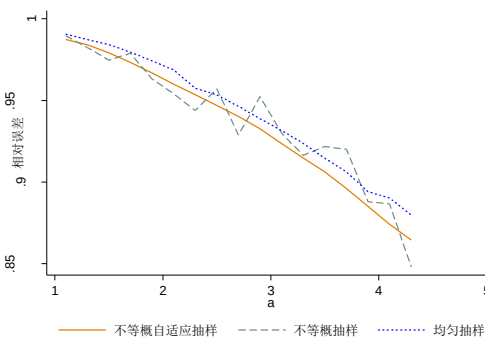
data4 与 data3 是同均值异方差，可以看出，随着模拟数据方差的增加，抽样的误差均值虽有所降低，但降低幅度不大，较接近于 data3 的误差均值。随着秩的增加，抽样误差均值仍呈降低趋势，且不等概自适应抽样的误差均值总是低于其他两种抽样。

整体来说，不等概自适应抽样的误差均值总是低于不等概抽样误差均值与均匀抽样误差均值，随着秩的增加，三种抽样方法的误差均值都呈下降趋势。此外，通过表 3.3 可以看出，秩$k=50$时，基于不等概自适应抽样的低秩重构矩阵与原数据矩阵的之间低秩逼近相对误差均值较低，表示此时的样本子矩阵具有较好代表性。

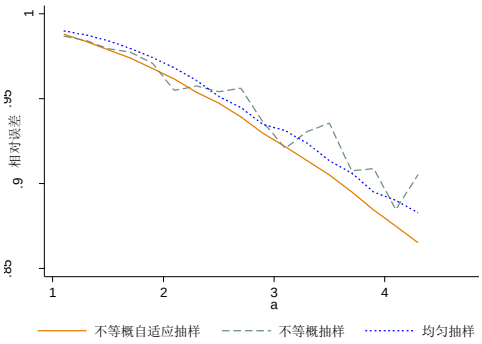
进一步，因为抽样的列数与行数不仅与秩k有关，还与迭代参数a相关。本节中，a的初始值为 0.5，步长为 0.2，循环迭代 20 次，将$k=10$和$k=50$时的三种抽样循环迭代的抽样误差通过图 3.1、图 3.2 可视化。可以看出，在整个抽样过程中，不等概自适应抽样不仅误差总是低于其他两种抽样，且该抽样稳定性一直较好；随着抽样列数$c=ak$的增加，不等概自适应抽样误差平稳降低，而其他两种抽样的误差都具有波动性。



(a)data1 (b)data2

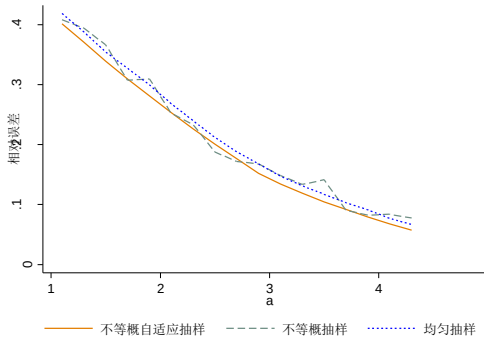


(c)data3

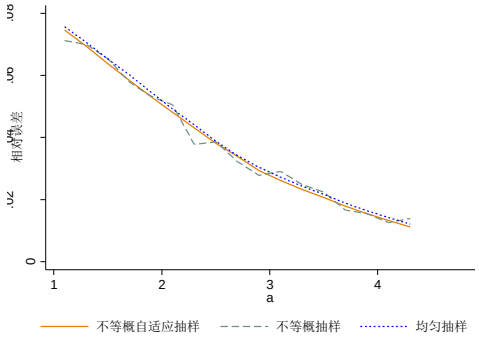


(d)data4

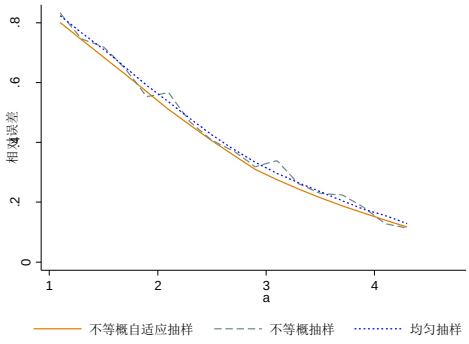
图 3.1 $k = 10$时各数据集不同抽样方法误差比较



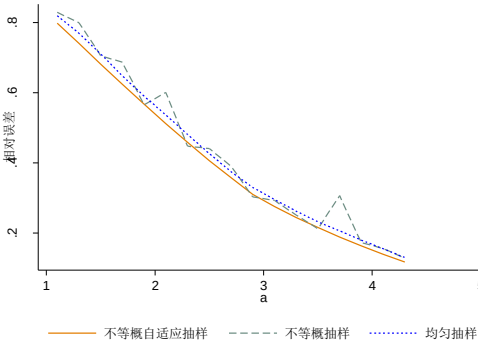
(a)data1



(b)data2



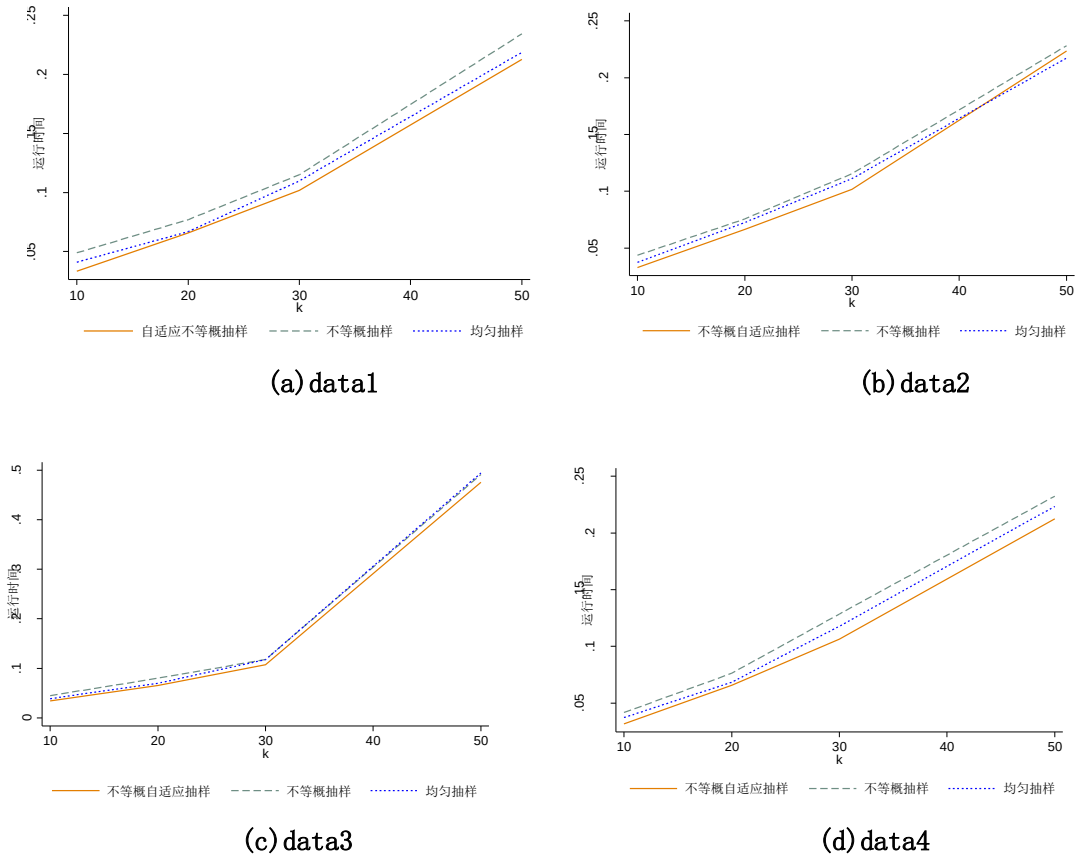
(c)data3



(d)data4

图 3.2 $k = 50$时各数据集不同抽样方法误差比较

2.抽样方法模拟运行时间比较



(a) data1 (b) data2

(c) data3 (d) data4

图 3.3 抽样方法运行时间比较

由图 3.3 可得，对于 data1，不等概自适应抽样的平均运行时间最少，其次是均匀抽样，再是不等概抽样；$k=20$时与$k=50$时，不等概自适应抽样的运行时间与均匀抽样运行时间较为接近，$20<k<50$时，三种抽样的运行时间相差较大。对于 data2，$k<45$时，不等概自适应抽样的运行最少，$k>45$时，该抽样方法运行时间上升，高于均匀抽样，而不等概抽样的运行时间总是高于其他两种抽样。对于 data3，$k<30$时，不等概自适应抽样运行时间少于均匀抽样，少于不等概抽样；$k>30$时，均匀抽样运行时间与不等概抽样重合，高于不等概自适应抽样。对于 data4，三种抽样的运行时间随着秩增加呈稳定上升趋势，不等概自适应抽样运行时间最少，其次是均匀抽样，再是不等概抽样。

总体来看，不等概自适应抽样的运行时间都是最少的，且随着秩的增加，抽样的运行时间呈增加趋势，这是因为k增加，抽样的列数或行数也增加，导致抽样的运行时间上升。

3.5.3 矩阵重构方法模拟

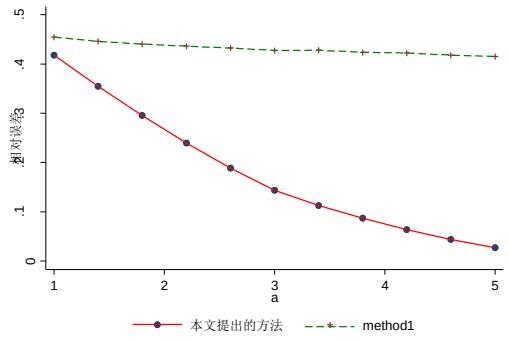
通过上一节对抽样方法的模拟比较结果，可以看出，不等概自适应抽样的精度，计算效率都优于不等概抽样和均匀抽样，因此，本节基于不等概自适应抽样研究矩阵重构方法的优良性，首先将四组模拟数据在不同秩下的重构误差均值进行展示，见表 3.4。

表 3.4 基于不等概自适应抽样和随机 SVD 分解的 CUR 矩阵分解重构误差

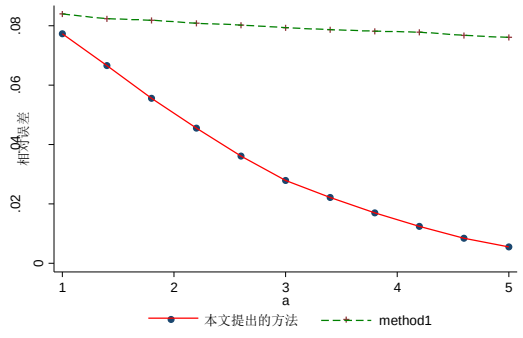
	data1	data2	data3	data4
k=10	0.4779	0.0883	0.9199	0.9194
k=20	0.3858	0.0719	0.7648	0.7638
k=30	0.3014	0.0566	0.6032	0.6026
k=50	0.1795	0.0341	0.3621	0.3622

如表 3.4 所示，同一组模拟数据随着秩的增加，基于不等概自适应抽样和随机 SVD 分解的 CUR 矩阵分解重构误差呈降低趋势，秩越大，误差越小。

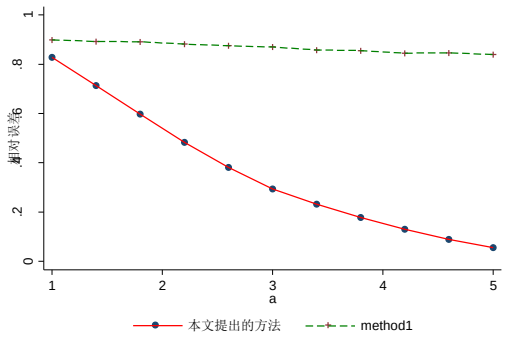
进一步，为精确比较矩阵重构方法改进前后的优良性差异，本节用矩阵 $\boldsymbol{U}=\boldsymbol{C}^{+}\boldsymbol{K}\boldsymbol{R}^{+}$（Wang，等 2013；Boutsidis 等，2017）的重构精度和时间作为参照，将该方法记为 method1；与本文新提出的基于不等概自适应抽样和随机 SVD 分解的矩阵重构方法（记为 method2）通过重构相对误差和运行时间进行对比。两种方法中，$k=50, c=ak, r_1=c, r_2=0.5r_1$，迭代参数选择$a\in[1,5]$，步长为 0.4，将循环 11 次的相对误差和运行时间可视化，如图 3.4、图 3.5 所示。



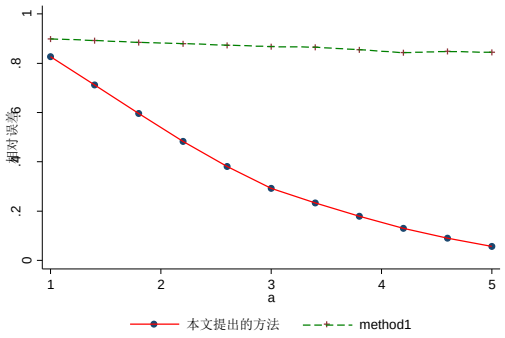
(a)data1



(b)data2



(c)data3



(d)data4

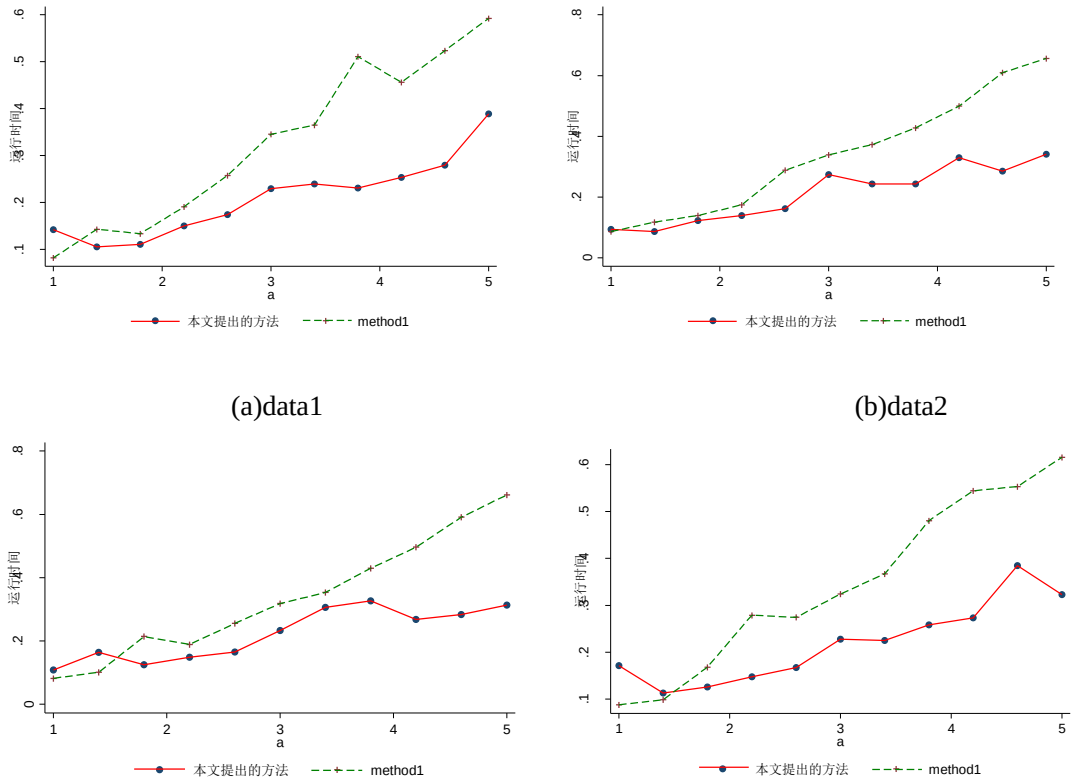
图 3.4 $k=50$不同数据集 CUR 矩阵分解精度比较

由图 3.4 可以看出，对于 data1，本节提出基于不等概自适应抽样和随机 SVD 分解的 CUR 矩阵分解方法的相对误差总是较低，随着a的增加，即样本行，样本列的增加，相对误差呈下降趋势。$k=50$时，method1 的重构误差随着a增加缓慢降低，而 method2 的重构误差明显降低，降低幅度远远大于 method1。

对于 data2，$k=50$时的 CUR 矩阵分解重构精确度折线图分布形状类似 data1 的图(a)，且 method2 的相对误差低于 method1，随着a增加，两方法的相对误差也越来越小。但 data2 与 data1 是同分布，形状参数不一样的两组模拟数据，形状参数增加，矩阵 CUR 分解重构误差明显降低，所以图 3.4 中(b)图的纵坐标值低于图(a)中的纵坐标值。

对于 data3，就矩阵重构相对误差来看，$k=50$时，method1 的相对误差变化较为平缓，折线图分布形状几乎呈“直线”式不变，而 method2 的相对误差呈“直线”式下降，降幅明显。

对于 data4，类似 data3，两数据的 CUR 矩阵分解重构精度折线图形状相近，且正态分布方差的增大对数据矩阵重构精度没有较大的影响，所以图(d)与图(c)较为相近；对于秩k，method2 的相对误差随着a增加呈降低趋势，远远低于 method1 的相对误差。



(a)data1 (b)data2

(c)data3 (d)data4

图 3.5 $k=50$不同数据集 CUR 矩阵分解效率比较

就运行时间来看，由图 3.5 可得，对于 data1，$k=50$时，当$1<a<1.4$时，两种方法的运行时间出现相交点，在相交点之前，method2 的运行时间较高，交叉点之后，method1 的运行时间高于 method2。对于 data2，$a=1$时，method2 的运行时间等于 method1，其他点时，随着a增加，method2 的运行时间总是低于 method1。对于 data3，$1.4<a<1.8$时，method2 的运行时间高于 method1，其他点处，method2 运行时间低于 method1。对于 data4，类似于 data3，两方法运行时间具有相交点，相交点之前，method2 运行时间较高，相交点后， method2 运行时间降低，一直低于 method1 运行时间。

整体来看，本节提出的基于不等概自适应抽样和随机 SVD 分解的 CUR 矩阵分解方法的精度优于 method1，而对于计算效率，在不同迭代参数 a 时，两种方法的表现具有差异性。因此，本节提出的矩阵分解重构方法的运行效率并不低于 method1 运行效率。

3.6 小结

CUR 矩阵分解是基于抽样对任意矩阵进行低秩分解逼近的一种有效手段，既能充分保留原始数据信息，也能降低高维矩阵维度。基于不等概自适应抽样和随机 SVD 分解，本节提出一种改进后的 CUR 矩阵分解重构方法，并通过模拟数据首先对本文提出的不等概自适应抽样就抽样误差和抽样运行效率与均匀抽样、不等概抽样进行了比较，研究结果得出，本节提出自适应抽样精度较高，运行效率也高于其他两种抽样方法；随着低秩k增加，抽样误差呈降低趋势。再通过模拟数据对基于自适应抽样和随机 SVD 分解的 CUR 矩阵分解重构方法和之前的 CUR 矩阵分解方法的精度和运行效率作了对比，研究结果发现，本节提出的方法的精度更高，运行效率也不低于之前的方法。

4 基于不等概抽样 Nyström 方法特征提取的上市公司谱聚类

上市公司是我国经济发展和体制改革的必然产物，已成为推动国民经济发展的重要力量，是现代经济发展微观主体融资的主要渠道(杜坤伦,2009)。能否准确把握上市公司的业绩表现，是投资者进行合理投资决策的关键。此外，在上市公司的定量分析中，聚类方法应用较为广泛。周焯华(2002)将聚类分析方法与证券投资进行了结合，基于行业因素、公司因素、收益性、成长性等基本方面对股票进行考察，并使用聚类分析方法来确定投资价值和投资范围。朱杰、缪瑞(2005)基于多元统计分析视角对北京上市公司的经营情况进行了分类研究，并使用了两种聚类方法进行了比较分析。韩海波和张仲杰(2006)就上市公司的分类方法进行了讨论，他们利用聚类方法对多只股票的众多财务数据进行处理，希望能够更全面客观的选出各板块的优质股。李德荣等(2011)利用聚类分析和因子分析对钢铁行业的上市公司进行了综合分析。刘宏杰(2014)基于 MATLAB 软件对中国创业板市场的上市公司利用类平均法进行了聚类分析，并基于财务指标将公司分为三类。楼润平、孙鹏(2017)利用聚类方法对在境外上市的中国互联网公司进行了归类，并对其发展过程进行了深入的探讨。杨莹、张学津等(2019)对上司公司金融板块进行了聚类研究，并基于推荐投资是视角下将其归为六类。芦苗、王冠华(2021)对农业上市公司进行了聚类分析，进而对农业上市公司的财务风险预警进行了深入的探讨。

综上分析，这些研究均基于上市公司某几项主要财务指标，利用主成分分析和因子分析得分，或直接利用多元数据进行聚类分析。从方法上来看，主成分分析和因子分析具有降低维度, 简化问题复杂度的功效。但该方法将本应合理分类的财务指标经过主成分分析和因子分析后降低了结果的可解释性；从数据信息上来看，上市公司财务指标包含维度很高，常见指标有 200 多个。仅从几个维度的财务指标进行分析，显然会损失大量信息。

因而本节在简化数据复杂度的同时，试图最大限度保留原始数据的有效性，提出了基于不等概抽样 Nyström 方法的特征提取方法。即通过提取影响上市公司业绩的主要特征指标，在降低数据维度和数据计算复杂度的同时最大可能保留原始数据信息，并在选取特征变量的基础上对上市公司进行谱聚类分析，旨在探索

分析上市公司特征，为投资者投资提供定量分析依据和投资警示。

4.1 谱聚类基本思想

给定一个由n个数据点组成的数据集，通过数据点之间的相似度可以构造一个$n\times n$的相似矩阵，再根据相似矩阵的特征向量进行聚类。

简而言之，谱聚类算法包括了三个步骤(Fiedler, 1973)：

第一步：图形构造。在n个待聚类点之间建立稀疏相似图；

第二步：计算谱嵌入。计算相似图的代表矩阵(如 Laplacian 矩阵)的前k个特征向量集$U_k=(u_1,u_2,\cdots,u_k)$，由U_k进一步计算谱嵌入$X=(x_1,x_2,\cdots,x_n)$，其中，第i个点的谱嵌入为$\boldsymbol{x}_i=\boldsymbol{U}_k(i,:)^{\mathrm{T}}/q(\boldsymbol{U}_k(i,:)_2)$，$q(\cdot)$为正则化函数；

第三步：聚类。对这些谱特征$X=(x_1,x_2,\cdots,x_n)$进行 k-means 聚类得到k个类中心$C=(c_1,c_2,\cdots,c_n)$，进一步将n个待聚类点根据其与各类中心远近程度聚类，得到k类。

4.2 基于不等概 Nyström 抽样方法的特征提取

特征提取即指对原始数据矩阵进行预处理，常见的方法包括主成分分析(PCA)、奇异值分解(SVD)、正交三角分解(QR)等。当矩阵规模较大时，直接对数据矩阵进行分解会导致内存溢出，增大算法的计算复杂度。因此，利用 Nyström 算法首先对大规模、稀疏矩阵进行低秩近似，降低维度，再做特征提取，可在提高算法的精确度同时降低计算复杂度。

4.2.1 相似矩阵的构建

设数据矩阵$\boldsymbol{K}\in\mathbb{R}^{n\times p}$，$n$为样本单元数，$p$为财务指标维度，则$x_i\in R^{n\times p}$为第$i$指标变量在$n$个样本单元上的取值，对数据标准化后，通过$k_{ij}=exp\left(-\frac{||x_i-x_j||^2}{2\sigma^2}\right)i,j=1,2,\cdots,p$，构建变量间的相似矩阵，也称为核矩阵，

$$\boldsymbol{K}=\begin{bmatrix}k_{11} & k_{12} & \cdots & k_{1p}\\ k_{21} & k_{22} & \cdots & k_{2p}\\ \vdots & \vdots & \vdots & \vdots\\ k_{p1} & k_{p2} & \cdots & k_{pp}\end{bmatrix}, \tag{4.1}$$

其中，矩阵$\boldsymbol{K}$为对称半正定矩阵。

4.2.2 特征提取

利用 Nyström 方法可以降低矩阵维度的优点，结合矩阵分解算法，如 SVD 分解、QR 分解、随机 SVD 分解等，构造 Nyström 特征提取算法，对指标变量进行特征提取。由(4.1)式相关矩阵构建与列抽样概率$P_i = \frac{||K^{(i)}||^2}{||K||_F^2}$可知，第$j$列表示第$j$个变量与所有变量$x_i$，$i = 1, 2, \cdots, p$的相关系数为列向量$K^{(i)}$，范数$||K^{(i)}||$越大，表示第$j$个变量与所有变量的相关性较高，被抽中入样的概率也较大。

利用 Nyström 方法得到低秩近似矩阵，减小矩阵规模，保留原数据矩阵的特性，降低后续矩阵分解的计算复杂度。而基于不等概抽样的 Nyström 方法，既可以保证所有变量被选中的随机性，又能体现变量间相关程度大小。因此，被抽中列对应第j个变量特征可视为被选中特征。具体的变量特征选择如算法 4.1 所示。

算法 4.1	变量特征提取
输入：	原数据矩阵$\boldsymbol{A}$，期望的低秩k。
输出：	选择出的变量特征矩阵。
step1：	数据预处理。对原数据矩阵$\boldsymbol{A}$进行中心标准化处理，处理后的矩阵记为$\boldsymbol{B}$；
step2：	构造变量相似度矩阵$\boldsymbol{K}$。每个元素k_{ij}可以通过高斯径向基核函数表示，即 $k_{ij} = exp\left(-\frac{||x_i - x_j||^2}{2\sigma^2}\right)$；
step3：	确定抽样比例。利用不等概 Nyström 算法得到低秩逼近误差处于可控正常水平时的抽样比例；；
step4：	变量特征提取。利用不等概 Nyström 抽样算法从变量与变量的相似矩阵$\boldsymbol{K}$中提取特征变量；
step5:	矩阵恢复。根据特征提取出的变量，从原数据矩阵$\boldsymbol{A}$提取相应变量指标，构造特征提取矩阵$\boldsymbol{Y}$。

4.3 上市公司谱聚类分析

4.3.1 数据来源与处理

本节的研究数据来源于锐思数据库，包括每股指标、盈利能力指标、偿还能力指标等 10 类股票财务比率指标，包含每股收益、每股净资产、每股营业收入等 232 个具体指标。通过公司年报，上市公司公开信息等资料，对数据中的异常值和缺失值进行了相关删除或插补。最终以 73 家上市公司，193 个股票财务指标作为研究对象。

4.3.2 抽样比确定与特征提取

通过不等概抽样 Nyström 方法对相似度矩阵$\boldsymbol{K}$进行低秩逼近，并对不同抽样比例下的相对误差进行对比。

表 4.1 上市公司数据低秩逼近结果对比

样本比 n/N	不等概抽样 Nyström	
	相对误差	绝对误差（d）
5%	0.3876	24.12
10%	0.2881	14.11
20%	**0.2007**	**11.51**
30%	0.1525	6.61
40%	0.1181	6.26
50%	0.1016	6.56

由表 4.1 可知，在不同的抽样比下，上市公司的股票收益数据应用本文的算法进行矩阵低秩逼近后的误差明显不同。随着抽样比例的增加，采样近似的相对误差和绝对误差都呈下降趋势，且当变量选择列抽样比为 20%时，误差大幅下降，重构矩阵与原数据矩阵的相对误差仅为 20.07%。

因此，本节选择列抽样比为 20%时进行特征提取，极大地降低原始数据变量的维度，又能最大程度保留原始数据信息，且在降低计算复杂度的同时，使算法的优良性得到充分的展示。根据算法 4.1，特征提取后的指标如表 4.2 所示。

表 4.2 特征选取的指标

变量	指标	变量	指标
Y_1	每股收益	Y_{16}	可持续增长率
Y_2	每股净资产	Y_{17}	应收账款周转天数
Y_3	每股息税前利润	Y_{18}	营运资金周转天数
Y_4	净资产收益率	Y_{19}	固定资产周转率
Y_5	资产净利率	Y_{20}	销售商品劳务收入现金/营业收入
Y_6	净利润/营业总收入	Y_{21}	每股现金及现金等价物余额
Y_7	息税折旧前利润/营业总收入	Y_{22}	股利保障倍数
Y_8	成本费用利润率	Y_{23}	流动资产/总资产
Y_9	速动比率	Y_{24}	非流动资产/总资产
Y_{10}	股东权益/负债合计	Y_{25}	股东权益/全部投入资本
Y_{11}	股东权益/带息债务	Y_{26}	流动负债权益比率
Y_{12}	营业利润增长率	Y_{27}	剔除预收账款后的资产负债率
Y_{13}	营业利润 3 年复合增长率	Y_{28}	股东权益比率
Y_{14}	利润总额 3 年复合增长率	Y_{29}	资本固定化比率
Y_{15}	每股经营活动产生的现金流量净额 3 年复合增长率	Y_{30}	净利润/营业总收入

4.3.3 基于选取特征的谱聚类

在上述特征提取后，数据包含为 73 个上市公司，30 个股票财务收益指标，均匀包含了原数据 10 大类指标，即经过特征选取后的指标有较好的代表性。本节将基于这 30 个指标对 73 上市公司进行谱聚类分析。谱聚类根据数据点的相似度矩阵进行分析，对于大规模数据具有优良的聚类效果，其过程中使用了降维，使计算复杂度也小于其他聚类算法。

1. 邻接矩阵构建

一般构建邻接矩阵的方法有三种，ϵ-临近法、K 近邻法和全连接法(贾洪杰,2017)。其中，K 近邻法会使重构后的邻接矩阵非对称。全连接法通过选择不同的核函数定义边的权重，常见的有多项式核函数、高斯径向基核函数、Sigmod 核函数。

本节采用高斯径向基核函数，此时，相似矩阵$\boldsymbol{S} \in \mathbb{R}^{n\times n}$和邻接矩阵$\boldsymbol{W}$相同。

$$\omega_{ij} = s_{ij} = exp\left(-\frac{||x_i - x_j||_2^2}{2\sigma^2}\right)$$

2. 度矩阵构建

根据谱聚类基本思想，利用特征提取后数据构造的一个无向加权图为$G(V,E)$，$V=\{1,2,\cdots,n\}$表示数据点的集合，$E=e_{ij}$表示连接两个顶点边集合，边的权重表示数据点之间的相似度。

有边连接的两个点之间的权重$\omega_{ij}>0$，否则，$\omega_{ij}=0$。度d_i表示和它相连所有边的权重之和，即$d_i=\sum_{j=1}^{n}\omega_{ij}$。度矩阵$\boldsymbol{D}$为对角阵。

$$\boldsymbol{D} = \begin{bmatrix} d_1 & \cdots & \cdots \\ \cdots & d_2 & \cdots \\ \vdots & \vdots & \ddots \\ \cdots & \cdots & d_n \end{bmatrix}_{n\times n}$$

3. 拉普拉斯矩阵构建

拉普拉斯矩阵通过前面定义的度矩阵和邻接矩阵构造，$\boldsymbol{L}=\boldsymbol{D}-\boldsymbol{W}$。矩阵$\boldsymbol{L}$

为对称阵，通过$\boldsymbol{L}_1 = \boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{L}\boldsymbol{D}^{-\frac{1}{2}}$对其进行标准化。

4. 上市公司的谱聚类

通过不等概抽样 Nystr öm 算法进行特征提取后，将提取后的指标对应原始数据，获得新的聚类数据集Y，再对Y进行谱聚类，具体的聚类算法流程如算法 4.2 所示。

算法 4.2 谱聚类算法

输入： 经过特征选取后的数据点集$Y = \{y_1, y_2, \cdots, y_n\}$，聚类数目$k$。
输出： 聚类产生的k个类。
Step1：构造相似矩阵：根据Y中数据点的相似性，构造相似度矩阵$\boldsymbol{S}$，每个元素s_{ij}可以通过高斯径向基核函数表示，即$s_{ij} = exp\left(-\frac{||x_i - x_j||^2}{2\sigma^2}\right)$。
Step2：构造度矩阵：根据数据点之间的权重构造度矩阵$\boldsymbol{D}_{nn}$。
Step3：计算拉普拉斯矩阵：根据度矩阵$\boldsymbol{D}_{nn}$和相似矩阵$\boldsymbol{S}$计算得到拉普拉斯矩阵$\boldsymbol{L}$。
Step4：对矩阵$\boldsymbol{L}$进行标准化，得到标准化后的矩阵$\boldsymbol{L}_1 = \boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{L}\boldsymbol{D}^{-\frac{1}{2}}$。
Step5：计算矩阵$\boldsymbol{L}_1$的前k个特征值$\lambda_1, \lambda_2 \cdots, \lambda_k$，及对应的特征向量$u_1, u_2 \cdots, u_k$。
Step6：将各自对应的特征向量组成矩阵，并按行标准化，最终形成$n \times k$的特征矩阵$\boldsymbol{F}$。
Step7：将$\boldsymbol{F}$中的每一行看作一个k维样本，利用 k-means 聚类进行聚类。

4.3.4 聚类个数的确定与效果评价

1. 聚类个数的确定

根据上文特征选择的结果，利用上述谱聚类算法进行聚类分析。首先，根据上市公司的数据特征、常见的分类方法以及可解释性，将 73 家上市公司的谱聚类个数选为$k = 4$。其次，利用采用交叉验证法确定高斯核函数的宽度参数σ，搜索范围为$[1\ , 20]$，搜索步长为 1。最终确定为$\sigma = 5.5$作为核函数的宽度参数。

2. 聚类效果评价准则

设总样品数为n，聚类时把所有样品合并为k个类$G_1, G_2, \cdots, G_k$，类G_i的样品数和重心分别为n_i和$\bar{x}_i$，$i = 1, 2, \cdots, k$，则$\sum_{i=1}^{k} n_i = n$，所有样品的总重心

$\bar{x}=\frac{1}{n}\sum_{i=1}^{k}n_i\bar{x}_i$，令

$$W=\sum_{j=1}^{n}(x_j-\bar{x})'(x_j-\bar{x}) \tag{4.2}$$

为所有样品的总离差平方和。

$$W_i=\sum_{j\in G_i}(x_j-\bar{x}_i)'(x_j-\bar{x}_i) \tag{4.3}$$

为类G_i中样品的类内离差平方和。

$$P_k=\sum_{i=1}^{k}W_i$$

为k个类的类内离差平方和之和。则

$$W=P_k+\sum_{i=1}^{k}n_i(\bar{x}_i-\bar{x})'(\bar{x}_i-\bar{x})$$

再令

$$R^2=1-\frac{P_k}{W} \tag{4.4}$$

其中，P_k/W值越小(即R^2越大)，表明类内离差平方和在总离差平方和中所占的比例越小，也就是说各类分得越开。因此，R^2统计量可用于评价合并成k个类时的聚类效果，R^2值越大，聚类效果越好(王学民,2017)。

按照谱聚类的聚类过程，根据特征选择后的 30 个股票收益指标，对 73 个上市公司进行聚类分析，数据较多，选择对数据应用范围较广的谱聚类分析方法。为了使数据更平稳、均匀地分布，首先对数据做了标准化处理，再按照谱聚类算法聚类，最终将样本聚为 4 类。

对谱聚类的 4 类聚类结果进行评价，可得$R^2=0.71$。而对相同指标及数据利用 k-means 聚类方法聚类结果的$R^2=0.4$。显然，谱聚类的聚类效果优于 k-means 聚类效果。

4.3.5 聚类结果分析

按照谱聚类的聚类过程，根据特征选择后的 30 个股票收益指标，对 73 家上市公司进行谱聚类，最终将样本聚为 4 类，具体结果如表 4.3 所示。

表 4.3 上市公司聚类结果

类别	类大小	公司名称
Cluster1	20	深圳能源、招商局积余产业运营、天马微电子、华锦化学工业、中金岭南有色金属、南玻集团、招商局港口集团、富奥汽车零部件、赛格股份、冰山冷热科技、江铃汽车、安道麦、长安汽车、鲁泰纺织、本钢板材、新华制药、鞍钢股份、中远海运能源运输、石油化工、复星医药
Cluster2	23	宝安集团、长城开发、华联控股、中洲投资、京基智农时代、德赛电池科技、华强实业、北方国际合作、长城科技、华侨城、天健(集团)、许继电气、物业发展、国药集团、大名城企业、普天通信、灿坤实业、万科企业、国际海运集装箱、中兴通讯、中联重科、丽珠医药、海信家电
Cluster3	17	盐田港、广聚能源、海洋直升机、华数传媒、新疆国际实业、深粮控股、精密科技、深圳房地产、深圳纺织、佛山电器、威孚高科技、张裕葡萄酿酒、黄山旅游、海南航空、杭州汽轮、中鲁远洋渔业、皖通高速
Cluster4	13	农产品集团、TCL 科技集团、东方盛虹、电力发展、东方科技、锦州港、建设汽车系统、山东航空、瓦房店轴承、华能国际电力、南方航空、东方航空、兖州煤业

第一类包括 20 家公司，这类的公司的每股净资产、净资产收益率、资产净利率、净利润/营业总收入、息税折旧前利润/营业总收入、营业利润增长率、等 10 项指标均值为正值，但涉及到如每股收益、每股息税前利润、成本费用利润率等 20 个指标均值为负值，且绝对值较大。该类公司可能会重组，或者会退市，属于高风险投资类公司，投资者需要谨慎考虑。

第二类包括 23 家公司，这类公司包括每股收益、每股净资产、每股息税前利润、净资产收益率等总资产周转率等 22 项指标值为正值，速动比率、股东权益/负债合计等 8 项指标均值为负值，且绝对值较小，属于低风险投资类公司，是投资者关注的主要对象，该类公司股票可长期持有。

第三类包括 17 家公司，该类公司有 9 项指标值均为正值，包括，成本费用利润率、速动比率、股东权益/负债合计、股东权益/带息债务等指标，其余指标均值皆为负值。该类公司总资产增长率的绝对值大，公司股价持续性强，总体呈

下降趋势，不建议投资者长期持有。

第四类包括 13 家公司，该类公司包括 4 项指标均值均为正，26 项指标值均为负值，其中包括每股收益、每股净资产、每股息税前利润、营业利润 3 年复合增长率等正向指标的指标值基本皆为负值。考虑到此类公司指标值为负的指标较多，并且盈利能力和投资报酬能力较低，投资者应该规避此类公司。

4.4 小结

本节结合使用特征提取与聚类分析两种实证分析研究方法，对上市公司的股票收益绩效指标数据进行了聚类分析。首先是利用不等概 Nyström 方法对评价指标较多的数据集进行了特征提取。相比主成分分析、因子分析等降维分析方法，不等概 Nyström 方法可以在充分保留原数据信息的同时，保证选取特征的代表性。而面对大规模数据集，本节提出的不等概 Nyström 方法更具有实用性。其次，利用谱聚类对特征提取后的数据进行聚类分析。谱聚类相比 kmeans 聚类，对数据的不同分布具有更强的适应性，计算复杂度更低，聚类效果更优，更能体现上市公司的真实绩效情况，有利于投资者做出更为准确的投资选择。

本节通过不等概 Nyström 方法对 193 个股票收益绩效指标数据，按照 20% 的抽样比例进行特征提取，提取出 30 个指标，均匀包含了原数据的 10 大分类指标数据，表示特征提取的结果具有较好的代表性。利用谱聚类对特征提取的指标数据进行聚类分析，最终将本文选取的 73 家上市公司分为 4 类，并通过聚类效果评价表明此次聚类具有良好的效果。

5 基于不等概自适应抽样与随机 SVD 分解的 CUR 矩阵重构的偏好特征提取

随着网络的快速发展，信息资源高速膨胀，更多的用户喜欢在网上购物、读书、看电影。对于像美团、淘宝、大众点评等依靠网络获取利润的企业，通过用户行为特征分析用户偏好，根据用户偏好可以进行相关产品推荐，减少用户搜索时间，以最低时间成本获取利润。但随着网络用户和产品的迅速增加，偏好特征数据越来越多，从海量数据中分析用户的偏好特征需要更多时间和存储空间。因此，对于该类网络企业，选择能够快速、准确分析用户行为偏好特征和产品偏好特征的方法显得尤为重要。

本节通过第 3 节提出的基于不等概抽样与随机 SVD 分解的行列联合抽样，CUR 矩阵分解方法，分解用户-产品评分矩阵。该方法基于原始数据抽样，因此数据可解释性较高，意义明确。相比传统特征提取方法，本章的方法将高维数据特征提取问题转为低维矩阵分解，计算更加简单。通过电影评分数据集进行实证检验，在相同实验条件下，就准确率与压缩率对方法性能进行评价，并进一步和 SVD 矩阵分解方法进行比较。

5.1 偏好特征提取算法

算法 5.1 基于 CUR 偏好特征提取算法

输入： 初始数据集，目标低秩k，迭代参数a。

输出： 用户社区特征$\boldsymbol{CU}$，产品聚类特征$\boldsymbol{UR}$，最有影响力的产品数量r，最有影响力的用户数量c

step1： 对数据集进行预处理成用户—产品评分数据集；

step2： 根据算法 3.1 得到矩阵$\boldsymbol{C}$、$\boldsymbol{R}$，$\boldsymbol{C}$为用户对产品聚类的偏好特征，$\boldsymbol{R}$为用户社区对产品的偏好特征；

step3： 根据算法 3.2 对矩阵$\boldsymbol{C}$、$\boldsymbol{R}^T$进行随机 SVD 分解得到$\boldsymbol{D}_C$和$\boldsymbol{D}_R$；

step4： 根据式 3.4 构造矩阵$\boldsymbol{U}=(\boldsymbol{D}_C^T\boldsymbol{C})^+(\boldsymbol{D}_C^T\boldsymbol{K}\boldsymbol{D}_R)(\boldsymbol{R}\boldsymbol{D}_R)^+$，$\boldsymbol{U}$为用户社区对产品聚类的偏好特征。

step5： 利用 step2 得到的$\boldsymbol{C}$和$\boldsymbol{R}$，step3 得到的$\boldsymbol{U}$，构造$\boldsymbol{CU}$和$\boldsymbol{UR}$,$\boldsymbol{CU}$为用户社区的特征，$\boldsymbol{UR}$为产品聚类的特征；

step6： 求矩阵$\boldsymbol{CU}$的列数c，$\boldsymbol{UR}$的行数r，r为最有影响力的产品数量，c为最有影响力的用户数量；

step6： 返回$\boldsymbol{C}$、$\boldsymbol{U}$、$\boldsymbol{R}$、$\boldsymbol{CU}$、$\boldsymbol{UR}$、r、c。

算法 5.1 中，根据算法 3.1 计算构造矩阵$\boldsymbol{C}$所需的列数和矩阵$\boldsymbol{R}$所需的行数时，需要计算原始矩阵每列每行被选择的概率，这个概率代表矩阵每列每行的潜在特征显著水平。当某个产品被用户打分较高时，这该产品所在列的范数越高，被选择的概率也较高。此外，当矩阵的某列代表的产品获得的评分较高时，说明该列所包含的特征也较高，因此每一列的被选择概率可以作为该列的特征值。矩阵每行的被选择概率也可以作为该行的特征值，概率越大，特征水平越高。

在选择矩阵列和行时，按照入样概率最大化原则，将入样概率较大的列和行从大到小提取，此目的就是提取特征水平较高的列和行。

矩阵$\boldsymbol{CU}$和$\boldsymbol{UR}$相当于权重矩阵，$\boldsymbol{CU}$表示用户特征偏好所占权重，$\boldsymbol{UR}$表示产品偏好特征所占权重，因为矩阵$\boldsymbol{C}$和$\boldsymbol{R}$分别包含产品特征和用户特征，所以 CUR 分解可同时提取产品和用户特征，这是其他矩阵低秩逼近方法所不具有的优势。

5.2 准确率与压缩率

5.2.1 准确率

将原始评分数据表示为矩阵$\boldsymbol{K} \in \mathbb{R}^{m\times n}$,使用 CUR 分解压缩高维评分矩阵，从$\boldsymbol{K}$中抽取列和行构造矩阵$\boldsymbol{C}$和$\boldsymbol{R}$，通过学习获得矩阵$\boldsymbol{U}$，使得 CUR 尽可能逼近矩阵$\boldsymbol{K}$，即 CUR 表示一个更紧凑的评分数据矩阵。为了评价 CUR 矩阵分解的优良性，误差率

$$error = \frac{||\boldsymbol{K} - \boldsymbol{CUR}||_F^2}{||\boldsymbol{K}||_F^2} \times 100$$

准确率为$r = 1 - error$。

5.2.2 压缩率

压缩率是原始矩阵中数据元素数与低秩近似矩阵元素总数的比率，在 CUR 矩阵分解中，通过 CUR 近似逼近矩阵$\boldsymbol{K}$，压缩率可以表示为：

$$R = \frac{mn}{mc + cr + rn}$$

因为$\boldsymbol{C}$和$\boldsymbol{R}$中有cr个元素是共有的，进一步可以对压缩率进行优化，减少冗余，提高压缩率，可以表示为：

$$R=\frac{mn}{mc+cr+(rn-cr)+(c+r)}.$$

5.3 实验及结果分析

5.3.1 数据来源

本节利用真实数据集，Movielen 数据集对基于不等概抽样与随机 SVD 分解的 CUR 矩阵分解偏好特征提取算法进行实证分析。Movielen 数据集是由美国 Minnesota 大学 GroupLens 项目组提供，包含 6040 名用户对 3952 部电影的评分数据。

5.3.2 数据处理

Movielen 数据集中包含 6040 名用户和 3952 部电影，电影数目较多，且每位用户不可能看完全部电影，只能对一部分看过的电影打分，因此该电影评分数据集为稀疏矩阵。为了方便说明算法有效性，本文选取 200 名用户，对 50 部电影的评分数据评估基于不等概抽样与随机 SVD 分解的 CUR 矩阵分解偏好特征提取算法性能，并与已知特征偏好提取方法性能进行比较。

为了使 CUR 矩阵分解更加准确，提取有意义的偏好特征，首先需要对稀疏矩阵进行填充，考虑矩阵的空值产生的主要原因包括：①用户没有观看该部电影；②用户看了电影但没有打分。因此，本文对稀疏矩阵进行“0”值填充。

5.3.3 实验讨论

首先对大小为200×50的原始电影评分数据集中的电影特征、用户信息进行统计，如图 5.1 和图 5.2 所示。原数据中共包含戏剧、音乐、动作等 8 种类型的电影，即 8 种特征，动画类和喜剧类电影数目较多；而电影评分的用户有 3 个年龄阶段特征，评分人数较多的是 10-25 岁。

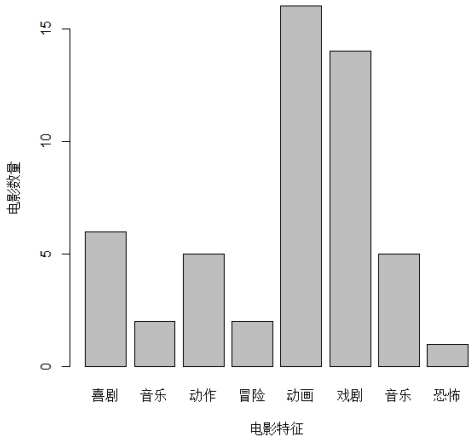


图 5.1 电影主要特征

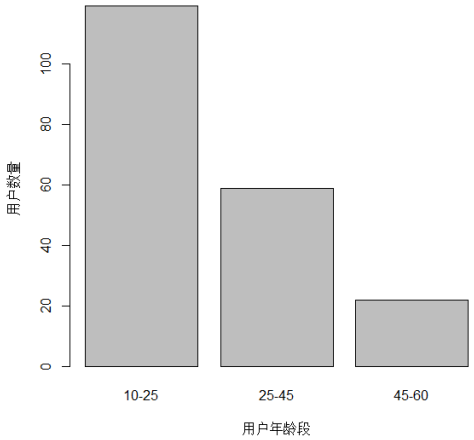


图 5.2 用户主要特征

计算矩阵每一列和每一行的l_2范数，将其与矩阵 F 范数之比分别作为每一部电影和每一个用户被选择的概率，如图 5.3 和图 5.4 所示。

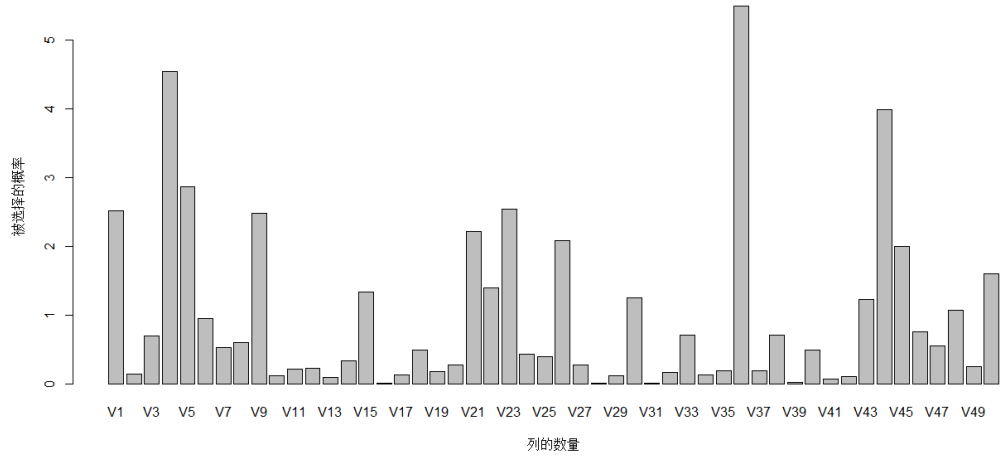


图 5.3 每部电影被选择的概率

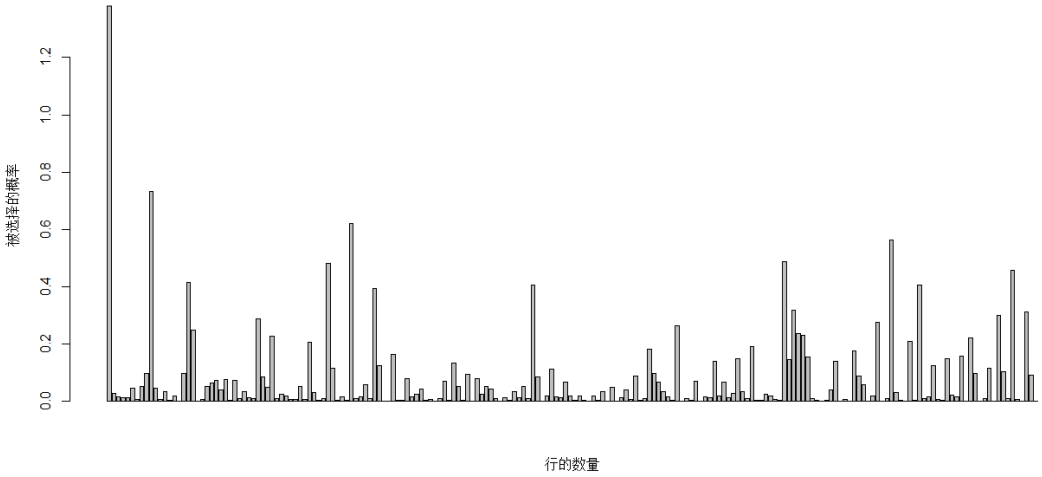


图 5.4 每位用户被选择的概率

从图 5.3 中可以看出，一些列的被选择概率较高，表示这些列代表的电影特征较明显，更容易被抽样算法选择。相反，一些列的被选择概率特别低，说明这些列的电影特征非常弱，可以忽略，也不会被抽样算法选择，从而提高 CUR 矩阵分解偏好特征提取算法提取电影特征的精确度。同样地，图 5.4 反映了用户特征，特征明显的一些用户更容易被选择。

将图 5.3 中特征明显、被选择概率较高的 16 部电影抽选出来，即从原矩阵中抽取 16 列构成矩阵$\boldsymbol{C}$，该矩阵几乎包含了原矩阵所有表示电影列的主要特征。将矩阵$\boldsymbol{C}$表示的电影特征通过图 5.5 展示。可以看出，原数据中的电影特征为 8 类，矩阵$\boldsymbol{C}$中包含的电影特征为 6，提取的电影偏好特征较准确。且比较受欢迎的电影特征为戏剧、动作，其次为动画和戏剧，再是冒险和恐怖。

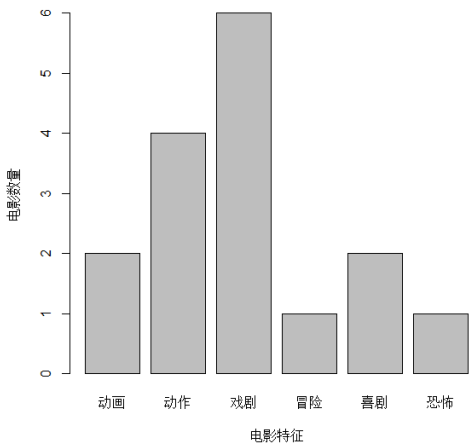


图 5.5 抽取的电影特征

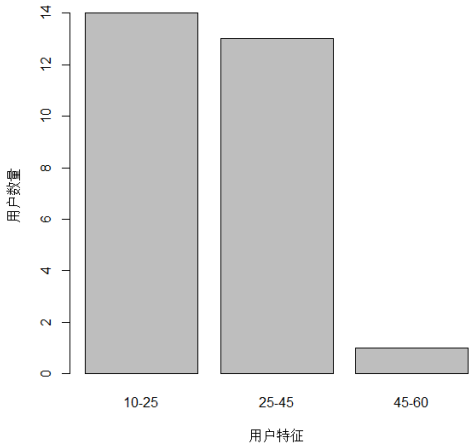


图 5.6 抽取的用户特征

将图 5.4 中用户特征明显、被选择概率较高的 20 名用户抽取，构造矩阵用户偏好特征矩阵$\boldsymbol{R}$，将抽取出的用户特征表示为图 5.6。可以看出，特征提取后的用户特征仍然为 3，喜欢看电影的用户主要集中在 10-25 岁，25-40 岁两个年龄段，用户偏好特征提取效果较好。

5.3.4 结果分析

得到用户对产品聚类的偏好特征$\boldsymbol{C}$，用户社区对产品的偏好特征$\boldsymbol{R}$后，需要构造用户社区对产品聚类的偏好特征$\boldsymbol{U}$，进一步得到为用户社区的特征$\boldsymbol{CU}$，产品聚类的特征$\boldsymbol{UR}$。通过CUR可以得到用户电影评分矩阵的近似低维表示矩阵，当最有影响力的产品数量为 16，最有影响力的用户数量为 20 时，CUR 矩阵分解低秩重构的准确率为 48%，压缩率为 2.36.

5.3.5 结果对比

本节使抽取的电影特征数和用户特征数发生动态变化，即增加抽取的列数和行数，比较此时基于不等概自适应抽样和随机 SVD 分解下 CUR 矩阵分解特征偏好提取的准确率和压缩率，并将本文的方法与一般的 SVD 分解特征偏好提取方法就准确率进行比较。

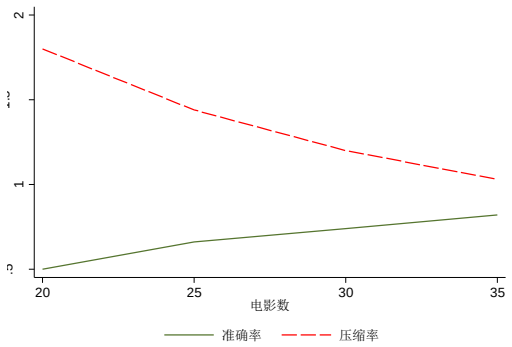


图 5.7 选取的不同列数对准确率和压缩率的影响

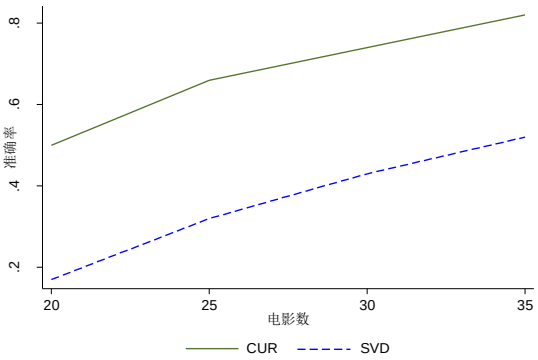


图 5.8 不同方法准确率对比

在图 5.7 中，利用基于不等概自适应抽样和随机 SVD 分解的 CUR 矩阵分解进行偏好特征提取，总共进行两轮抽样，抽取的行数$r = 1.5c$，可以看出，随着抽取的列数和行数，即电影数和用户数的增加，该方法的准确率呈上升趋势，压缩率呈下降趋势。由图 5.8 可以看出，利用 CUR 矩阵分解进行特征偏好提取的

精确度远远高于 SVD 分解下特征偏好提取的精确度。

5.4 小结

本节利用基于不等概自适应抽样和随机 SVD 分解的 CUR 矩阵分解进行偏好特征提取，通过真实数据集，Movielen 数据集进行实证检验。研究结论得出，利用 CUR 矩阵进行偏好特征提取算法性能较好，提取的用户或产品的特征能较好地反映原始数据特征；且随着抽样提取的列数和行数的增加，偏好特征提取的准确率呈上升趋势，压缩率呈下降趋势；近一步，将基于 CUR 矩阵分解的偏好特征提取方法随基于 SVD 分解的偏好特征提取方法相比，得到本节提出的方法的准确度远远高于后者。

6 总结与展望

6.1 总结

由于目前海量数据大多以高维矩阵形式存在，本文针对高维矩阵降维，从抽样的角度出发，研究高维矩阵低秩逼近方法以及误差测度。从方法精度和计算复杂度着手，希望提高方法精度的同时可以降低计算复杂度。主要工作包括两部分：

一是从理论分析角度出发，首先基于不等概抽样，构造了不等概 Nyström 方法，使其在抽样构造样本子空间时可以充分考虑到矩阵每列信息的不同，得到的样本更具代表性、随机性，且能充分保留原数据的有效信息；进一步利用 Nyström 方法进行矩阵重构时，结合随机 SVD 分解方法，加速对交叉矩阵奇异向量的求解，提高矩阵重构精度。数据模拟显示，基于不等概抽样和随机 SVD 分解的 Nyström 方法不仅保证了抽出样本的有效性，而且在提高矩阵低秩逼近精度的同时有效降低了计算复杂度。

其次，针对 Nyström 方法只适用于 SPSD 矩阵低秩重构的不足，进一步推广到一般矩阵的低秩逼近，基于不等概自适应抽样和随机 SVD 分解，提出一种改进后的 CUR 矩阵分解重构方法，并给出了新的误差界。对不等概自适应抽样就抽样误差和抽样运行效率与均匀抽样、不等概抽样通过模拟数据进行了比较，研究发现，不等概自适应抽样精度更高，运行效率也更高。基于行列抽样构成的样本矩阵，利用随机 SVD 分解进行了二次低秩逼近，并近似得到最优联合矩阵，进一步得到原矩阵的 CUR 分解重构矩阵。再通过模拟数据对基于自适应抽样和随机 SVD 分解的 CUR 矩阵分解重构方法和之前已有的 CUR 矩阵分解方法的精度和运行效率作了对比，研究结果发现，本文提出的方法的精度更高，运行效率也不低于之前的方法。

二是从实际应用的角度出发，首先将基于不等概抽样的 Nyström 低秩逼近方法运用于谱聚类。利用不等概 Nyström 方法对评价指标较多的数据集进行了特征提取。充分保留原数据信息的同时，保证选取特征的代表性；再利用谱聚类对特征提取后的数据进行了聚类分析，相比一般的 k-means 聚类，本文的方法整体提

高了聚类精度。

其次，将基于不等概自适应抽样和随机 SVD 分解的 CUR 矩阵分解应用于偏好特征提取，相比基于 SVD 分解的体征提取方法，本文提出的方法只对某些列和某些行进行运算，可以较大地降低计算复杂度。而且通过实证发现，利用本文提出的 CUR 矩阵分解进行偏好特征提取，准确度也优于基于 SVD 分解的偏好特征提取。

6.2 展望

关于高维矩阵低秩逼近，本文从抽样的角度出发，对 Nyström 低秩逼近方法和 CUR 矩阵分解方法做了一些改进工作，但关于高维矩阵低秩逼近的许多算法仍处于理论分析阶段，应用于实际问题中，仍存在较多困难。主要表现在以下几方面。

（1）Nyström 低秩逼近主要针对对称半正定矩阵，对数据要求较高。本文在应用中首先对原始数据矩阵进行了相似矩阵求解，基于相似矩阵进行低秩逼近，满足该方法对数据的特定性要求。在下一步研究中，需要探索更好地满足该类低秩逼近方法要求的数据处理技术。

（2）高维矩阵低秩逼近中，关于低秩的确定对方法精度具有重要影响。本文第 2 节中通过抽样比例确定目标秩，第 3 节中，主观确定目标秩，在下一步研究中，需要探索确定低秩的具体算法。

（3）关于误差测度，本文基于矩阵 F 范数进行测度，该方法易受异常值影响。因此，需要进一步研究关于高维矩阵低秩逼近误差测度方法。

参考文献

[1]Arslan O, Theodorou E A, Tsiotras P. Information-theoretic stochastic optimal control via incremental sampling-based algorithms [C]//2014 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL). IEEE, 2014: 1-8.

[2]Anupama N, Jena S, Sankar V R. A Novel Approach Using Incremental Fusion Sampling for Data Stream Mining [J].International Journal of Computer Sciences and Engineering, 2019,7(5):407-415.

[3]Boutsidis C, Woodruff D P. Optimal CUR matrix decompositions [J].SIAM Journal on Computing, 2017, 46(2): 543-589.

[4]Chen J J. Determinants of capital structure of Chinese-listed companies [J]. Journal of Business research, 2004, 57(12): 1341-1351..

[5]Chen G, Firth M, Xu L. Does the type of ownership control matter? Evidence from China's listed companies [J]. Journal of Banking & Finance, 2009, 33(1):171-181.

[6]Depoers F. A cost benefit study of voluntary disclosure: some empirical evidence from French listed companies [J].European Accounting Review, 2000, 9(2):245-263.

[7]Drineas P, Mahoney M W, Cristianini N. On the Nyström Method for Approximating a Gram Matrix for Improved Kernel-Based Learning [J].Journal of machine learning research, 2005, 6(12):156-158.

[8]Drineas P, Kannan R, Mahoney M W. Fast Monte Carlo algorithms for matrices I: Approximating matrix multiplication [J].SIAM Journal on Computing, 2006, 36(1): 132-157.

[9]Drineas P, Kannan R, Mahoney M W. Fast Monte Carlo algorithms for matrices II: Computing a low-rank approximation to a matrix[J].SIAM Journal on computing, 2006, 36(1): 158-183.

[10]Deshpande A, Rademacher L, Vempala S, et al. Matrix approximation and projective clustering via volume sampling [J].Theory of Computing, 2006, 2(1): 225-247.

[11]Drineas P, Mahoney M W, Muthukrishnan S.Relative-error CUR matrix decompositions [J].SIAM Journal on Matrix Analysis and Applications, 2008, 30(2): 844-881.

[12]Elhamifar E, Vidal R.Sparse subspace clustering: Algorithm, theory, and applications [J].IEEE transactions on pattern analysis and machine intelligence, 2013, 35(11): 2765-2781.

[13]Fine S, Scheinberg K. Efficient SVM training using low-rank kernel representations[J].Journal of Machine Learning Research, 2001,2(Dec): 243-264.

[14]Fiedler M. Algebraic connectivity of graphs[J].Czechoslovak mathematical

journal, 1973,23(2):298-305.

[15]Frieze A,Kannan R,Vempala S.Fast Monte-Carlo algorithms for finding low-rank approximations[J].Journal of the ACM (JACM), 2004, 51(6): 1025-1041.

[16]Fine S, Scheinberg K.Efficient SVM training using low-rank kernel representations[J].Journal of Machine Learning Research, 2001, 2(Dec): 243-264.

[17]Golub G H, Hoffman A, Stewart G W. A generalization of the Eckart-Young-Mirsky matrix approximation theorem [J]. Linear Algebra and its applications, 1987, 88: 317-327.

[18]Stewart G.Four algorithms for the the efficient computation of truncated pivoted QR approximations to a sparse matrix [J].Numerische Mathematik,1999,83(2):324-332.

[19] Halko N, Martinsson P G, Tropp J A. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions [J]. SIAM review, 2011, 53(2): 217-288.

[20]Wang H, Liu D J and Pu G L. Nuclear reconstructive feature extraction [J]. Neural Computing and Applications, 2019, 31(7) : 2649-2659.

[21]Hlindzich D. Medical image analysis methods for anatomical surface reconstruction using tracked 3D ultrasound [D]. Diss, 2014.

[22]Holodnak J T. A Probabilistic Subspace Bound with Application to Active Subspaces[J]. Siam Journal on Matrix Analysis & Applications, 2018, 39(3):1208-1220.

[23]Hoyle D C. Accuracy of pseudo-inverse covariance learning—A random matrix theory analysis[J].IEEE transactions on pattern analysis and machine intelligence, 2010, 33(7): 1470-1481.

[24]Joachims T.Making Large-Scale SVM Learning Practical[J].Technical Reports,1998, 8(3):499-526.

[25]Ji S, Zhang Z, Ying S, et al. Kullback-Leibler Divergence Metric Learning [J]. IEEE Transactions on Cybernetics, 2020, PP(99):1-12.

[26]Wang J, Yu Z.Quadratic curve and surface fitting via squared distance minimization[J]. Computers & Graphics, 2011, 35(6): 1035-1050.

[27]Wang J, Wu L, et al. Maximum weight and minimum redundancy: A novel framework for feature subset selection [J].Pattern Recognition, 2013, 46(6):1616-1627.

[28]Jin J. Multivariate Statistical Analysis on Competitiveness of Environmental Listed Companies in China[J].International Business & Management, 2011, 2(2):1-5.

[29]Jia H, Ding S, Du M. Self-Tuning p-Spectral Clustering Based on Shared Nearest Neighbors[J]. Cognitive Computation, 2015, 7(5):622-632.

[30]Jia H, Ding S, Du M. A Nyström spectral clustering algorithm based on probability incremental sampling[J]. Soft Computing, 2017, 21(19): 5815-5827.

[31]Kumar S, Mohri M, Talwalkar A. Sampling methods for the Nyström method[J].

Journal of Machine Learning Research, 2013, 13(1):981-1006.

[32]Kumar S, Mohri M, Talwalkar A.On sampling-based approximate spectral decomposition[C]//Proceedings of the 26th annual international conference on machine learning. 2009: 553-560.

[33]Li M, Bi W, Kwok J T, et al. Large-Scale Nyström Kernel Matrix Approximation Using Randomized SVD[J]. IEEE Transactions on Neural Networks & Learning Systems, 2014, 26(1):152-164.

[34]Li M, Kwok Y, Lü B.Making large-scale Nyström approximation possible[C]//ICML 2010-Proceedings, 27th International Conference on Machine Learning. 2010: 631.

[35] Li C Y, Zhu C G. A multilevel univariate cubic spline quasi-interpolation and application to numerical integration [J]. Mathematical Methods in the Applied Sciences, 2010, 33(13):1578-1586.

[36]Liu H, Jing L, Qian Y, et al. Adaptive Local Low-rank Matrix Approximation for Recommendation[J].ACM Transactions on Information Systems, 2019, 37(4):1-34.

[37]Mahoney M W, Drineas P. CUR matrix decompositions for improved data analysis [J]. Proceedings of the National Academy of Sciences, 2009, 106(3):697-702.

[38]Rahmani M, Atia G K. Randomized robust subspace recovery for big data [J]. IEEE, 2015:1-6.

[39]Rahmani M, Atia G. A subspace learning approach for high dimensional matrix decomposition with efficient column/row sampling[C]//International Conference on Machine Learning. PMLR, 2016: 1206-1214.

[40]Loyola R, Pedergnana M, García S G. Smart sampling and incremental function learning for very large high dimensional data [J]. Neural Networks, 2016, 78(D3):75-87.

[41]Shang R, Xu K, Jiao L. Subspace learning for unsupervised feature selection via adaptive structure learning and rank approximation[J]. Neurocomputing, 2020, 413: 72-84.

[42]Wold S, Esbensen K, Geladi P. Principal component analysis [J].Chemometrics and Intelligent Laboratory Systems,1987, 2(1):37-52.

[43]Wang S, Pedrycz W, Zhu Q, et al. Subspace learning for unsupervised feature selection via matrix factorization[J]. Pattern Recognition, 2015, 48(1): 10-19.

[44]Shi W, Guo Y F, Xue X. Matrix-based kernel principal component analysis for large-scale data set[C]//2009 International Joint Conference on Neural Networks. IEEE, 2009: 2908-2913.

[45]Shusen W. et al. Improving CUR Matrix Decomposition and the Nyström Approximation via Adaptive Sampling[J].Journal of Machine Learning Research, 2013, 14: 2729-2769.

[46]Wang S, Pedrycz W, Zhu Q, et al. Unsupervised feature selection via maximum projection and minimum redundancy[J].Knowledge-Based Systems,2015,75: 19-29.

[47]Taheri E, Ferdowsi M H, Danesh M. Closed-loop randomized kinodynamic path

planning for an autonomous underwater vehicle[J].Applied Ocean Research, 2019, 83:48-64.
[48]Yang W, Daid. Two-dimensional maximum margin feature extraction for face recognition [J].Systems Man and Cybernetics, 2009, 39(4):1002-1012.
[49]Williams C, Seeger M. The Effect of the Input Density Distribution on Kernel-based Classifiers[C]// Seventeenth International Conference on Machine Learning. Morgan Kaufmann Publishers Inc. 2000.
[50]Williams C, Seeger M. Using the Nyström method to speed up kernel machines[C]//Proceedings of the 14th annual conference on neural information processing systems. 2001 (CONF): 682-688.
[51]Wang S, Zhang Z, Zhang T. Towards more efficient SPSD matrix approximation and CUR matrix decomposition[J].The Journal of Machine Learning Research,2016, 17(1): 7329-7377.
[52]Wang S, Zhang Z. Improving CUR matrix decomposition and the Nyström approximation via adaptive sampling[J]. The Journal of Machine Learning Research, 2013, 14(1): 2729-2769.
[53]Yunhe W, Yuan G, Chao X. Manifold Learning Method for Large Scale Dataset Based on Gradient Descent[J]. Atlantis Press ,2013,12: 1180-1187.
[54]Saeys Y, Inza I, Larranaga P. A review of feature selection techniques in bioinformatics[J]. bioinformatics, 2007, 23(19): 2507-2517.
[55]蔡雨虹,吴小俊.用于人脸识别的类内低秩子空间学习[J/OL].计算机科学与探索 :1-10[2022-05-20].http://kns.cnki.net/kcms/detail/11.5602.TP.20210608.1059.013.html
[56]段菲,王慧敏,张超.面向数据表示的 Cauchy 非负矩阵分解[J].计算机科学,2021,48(06):96-102.
[57]杜坤伦.上市公司对我国经济发展的影响[J].经济学家, 2009,8(008):101-102.
[58]耿妍,窦霁虹,王俊荣,孙梦娇,张安申.基于子空间特点的高维数据分类方法分析[J].延安大学学报(自然科学版),2016,35(03):29-32.
[59]顾天奇,罗祖德,胡晨捷,林述温.测量数据的曲线曲面拟合算法[J].东北大学学报(自然科学版),2021,42(03):408-413.
[60]贺文琪,刘保龙,孙兆川,王磊,李丹萍.基于核保持嵌入的子空间学习[J].计算机科学,2021,48(06):79-85.
[61]韩海波,张忠杰.上市公司分类方法新探[J].商业时代, 2006(25):65-67.
[62]贺玲,吴玲达,蔡益朝.数据挖掘中的聚类算法综述[J].计算机应用研究,2007(01):10-13.
[63]贾洪杰.大规模复杂数据的谱聚类研究[D].中国矿业大学,2017.
[64]李毅,米子川.大数据挖掘的均匀抽样设计及数值分析[J].统计与信息坛,2015,30(04):3-6.
[65]雷恒鑫,刘惊雷.基于行列联合选择矩阵分解的偏好特征提取[J].模式识别与人工智能,2017,30(03):279-288.
[66]刘松华,张军英,丁彩英.核矩阵列相关低秩近似分解算法[J].模式识别与人工智能,2011,24(06):776-782.
[67]李小艳,陈绍平.改进差分进化算法求解 B 样条曲线曲面拟合问题[J].计算机应

用与软件,2018,35(03):275-281+298.
[68]李骜,刘鑫等.基于低秩表示的鲁棒判别特征子空间学习模型[J].电子与信息学报,2020,42(05):1223-1230.
[69]李德荣,何莉敏,李玉.聚类分析和因子分析在股票投资中的应用[J].内蒙古统计, 2011(01):29-31.
[70]李海萍.上市公司被 ST 后盈余管理增强还是减弱[J].财会通讯, 2019,(01):21-24.
[71]刘宏杰.中国创业板市场上市公司的聚类分析—基于 MATLAB 的不同财务指标实证研究[J].东北大学学报(社会科学版), 2014, 16(04):360-365.
[72]楼润平,孙鹏,毛彧.中国互联网境外上市公司的聚类、回归与演化分析[J].统计与信息论坛, 2017, 32(12):64-71.
[73]芦笛,王冠华.农业上市公司的财务风险预警研究—基于因子分析法和聚类分析法[J].会计之友, 2019(24):79-83.
[74]李平.ST 上市公司"摘帽"成功概率与盈余管理相关性研究[J].财会通讯, 2020(14):88-91.
[75]唐文俊,左亚尧,张波,张祖传.一种基于密度聚类 Nystrom 抽样算法[J].计算机工程与科学,2012,34(11):148-152.
[76]王学民.应用多元统计分析[M]（第 5 版）.上海财经大学出版社,2017.
[77]王思奇.基于渐进迭代逼近的自适应曲线曲面拟合研究[D].大连理工大学, 2018.
[78]武继刚,陈招红,孟敏.基于低秩稀疏表示的子空间学习研究综述[J].华中科技大学学报(自然科学版), 2021,49(02):1-19.
[79]许楠.基于子空间学习的多视角聚类方法研究[D].大连理工大学, 2019.
[80]晏振,戴晓文,田茂再.基于杠杆值大数据集抽样的异常点诊断[J].数理统计与管理, 2016, 35(5):794-802.
[81]杨莹,张学津,苏小琪,葛志远.上市公司金融板块的聚类分析探讨[J].中国市场, 2019(34):4-7+15.
[82]杨美姣,刘惊雷.基于 Nyström 方法的偏好特征提取[J].计算机应用,2018,38(09):2515-2522.
[83]曾琦,李国盛,郭云鹏等. 高维数据降维中SVD与CUR分解对比分析[J].中原工学院学报, 2014, 25(06): 80-84.
[84]赵知劲,刘中健,赵治栋.基于 KL 散度的增量非负矩阵分解盲源分离算法[J].杭州电子科技大学学报,2014,34(05):7-11.
[85]张婷.一元二元三次样条空间上的曲线曲面拟合[D].大连理工大学, 2019.
[86]曾清红,卢德唐.基于移动最小二乘法的曲线曲面拟合[J].工程图学学报, 2004(01):84-89.
[87]邹珊.面向高维数据的共享子空间识别方法研究[D].北京交通大学,2015.
[88]詹永杰,龙飞,卜轶坤.基于独立子空间分析特征学习的表情识别[J].系统仿真学报,2015,27(10):2316-2319+2327.
[89]周焯华, 陈文南, 张宗益.聚类分析在证券投资中的应用[J].重庆大学学报(自

然科学版),2002(07):122-126.
[90]朱杰,缪瑞.上市公司聚类判别分析研究[J].统计与决策,2005(18):41-43.

致谢

毕业论文写至此处，即将完结，内心首先比较激动，成就感满满。其次，想到即将结束自己的硕士学习阶段，有许多不舍的回忆，也有许多要感谢的人。

首先要感谢的就是恩师牛成英教授，老师对待科研的认真、严谨态度深深影响着我，在我每一篇论文写作过程中，老师不仅给予耐心的指导，而且多次细心的帮我修改，包括调整论文框架、修改文字表述等，从整体到细微，每一次的修改建议都使我受益良多，逐步提升了我的写作水平，才能有毕业论文的顺利完成。除了学业上的辛勤指导，老师也教给我许多为人处事的态度，持之以恒，终有所成。

其次，在兰州财经大学七年的学习，使我对统计学这门学科有了较深的认识与理解，从本科到硕士毕业，不仅完善了理论学习，更提升了实际应用能力，感谢母校的栽培，感谢兰财各位老师的谆谆教诲与倾囊相授。

最后，要感谢家人的支持、朋友的鼓励与倾听、室友的爱心与包容，同门师弟师妹的诸多帮助，还有自己的努力与坚持。

虽然前路未知，但学到的知识与良师益友的鼓励将永远铭记心中，陪伴终生，前行路上继续未来可期，亦往事可回首。