

南 开 大 学

本 科 生 毕 业 论 文（设 计）

中文题目：基于 L21 范数正则化的多非负矩阵分解算法的实验分析

外文题目：Research on Multiple Non-negative Matrix Factorization with L21-norm Regularization

学 号：1511509

姓 名：赵欣璇

年 级：2015 级

专 业：软件工程

系 别：软件工程

学 院：软件学院

指导教师：谢茂强 副教授

完成日期：2018 年 5 月 1 日

关于南开大学本科生毕业论文（设计）的声明

本人郑重声明：所呈交的学位论文，是本人在指导教师指导下，进行研究工作所取得的成果。除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人创作的、已公开发表或没有公开发表的作品内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。本学位论文原创性声明的法律责任由本人承担。

学位论文作者签名：

年 月 日

本人声明：该学位论文是本人指导学生完成的研究成果，已经审阅过论文的全部内容，并能够保证题目、关键词、摘要部分中英文内容的一致性和准确性。

学位论文指导教师签名：

年 月 日

摘 要

推荐系统 (Recommender Systems, RSs) 是一种信息过滤系统, 可以根据历史信息得到用户兴趣或项目特征, 从而为用户推荐他感兴趣的产品, 例如电影、书籍、音乐等。尤其是随着信息技术的高速增长, 以及互联网日新月异的发展速度, 信息超载是每个互联网用户都在面临的问题, 推荐系统在人工智能领域当中的作用就尤为重要。越来越多的研究人员投入到推荐系统领域, 而且提出了很多有效的算法来解决问题, 本文提出了一种基于 L21 正则化的非负矩阵分解算法 (Multi-View Non-negative Matrix Factorization with Hierarchical Side-Information, HMNMF), 有效地利用数据的层次结构信息解决 Top-N 推荐系统问题。通过将算法应用在三个经典数据集上, 并且同其他推荐算法进行对比, 发现 HMNMF 算法在准确性指标上具有明显的性能优势。

根据实验结果, 本文又提出了两个需要进一步探索的问题, 一、如何用科学的方式验证 HMNMF 算法相比于其他算法具有优异的性能表现; 二、如何验证层次结构在 HMNMF 算法中的作用。对此, 我们设计了一系列相应的补充实验进行验证, 旨在通过详细的实验验证不同因素对于算法性能的提升作用, 为今后的工作夯实基础。

关键词: 推荐系统; 非负矩阵分解; 层次结构; 实验分析

Abstract

Recommender System is an information filtering system that can derive user interests or project characteristics based on historical information and provide users with advice. Especially with the rapid growth of information technology and the rapid development of Internet, information overload is a problem that faced by every Internet user. The role of recommender system in the field of artificial intelligence is particularly important. More and more researchers are investing in the field of recommender system, and many effective algorithms are proposed to solve the problem. This paper proposes a Multi-view Non-negative Matrix Factorization algorithm with Hierarchical side-informtion (HMNMF) based on L21 regularization, which effectively solves the Top-N recommender system problem by using hierarchical information of data. By applying the algorithm to three classical data sets and comparing with other recommended algorithms, it is found that the HMNMF algorithm has obvious performance advantages in accuracy index. According to the experimental results, this paper puts forward a series of questions to explore the influence of different factors on the results, and designs the corresponding supplementary experiments to verify. It aims to verify the improvement of the performance of different algorithms by detailed experiments, and lay a solid foundation for future work.

Keywords: recommender system; non-negative matrix factorization; hierarchical structure; experimental analysis

目 录

第一章 绪论.....	1
第一节 研究背景和意义.....	1
第二节 研究现状.....	1
第三节 论文的主要工作.....	2
第四节 论文的内容组织.....	2
第二章 背景知识介绍.....	4
第一节 推荐系统.....	4
第二节 协同过滤.....	7
第三节 非负矩阵分解.....	8
第四节 层次结构信息.....	9
第三章 相关算法介绍.....	13
第一节 基于协同过滤的推荐算法.....	13
第二节 基于非负矩阵分解的推荐算法.....	14
第四章 实验准备和 HMNMF 算法.....	17
第一节 实验准备.....	17
第二节 HMNMF 算法.....	19
第三节 实验结果.....	22
第五章 实验验证与分析.....	25
第一节 L21 范数健壮性的验证.....	25
第二节 层次结构的作用.....	29
第六章 总结与展望.....	32

参考文献.....	33
致谢.....	35

第一章 绪论

第一节 研究背景和意义

随着电子商务网站的发展，单独一个网站就会有上亿条的可用信息甚至更多，这远远超出了用户的处理能力，网络信息的爆炸性增长和多样化使得用户面临着选择的难题，如何在大量的可用信息中进行有效过滤得到使用户满意的项目（产品或服务）变得极为迫切。这就需要推荐系统的帮助，在大量的可用信息中为用户提供建议，帮助用户做出抉择。推荐系统的运用场景是极其广泛而丰富的，无论是书籍、音乐、电影或各种商品、服务，需要用户选择项目的场景就需要用到推荐系统，甚至一些其他问题如响应预测、基因-表型关联等问题都可以抽象为推荐系统问题，因此推荐算法有着极其广泛的应用场景，对于推荐算法的研究有着非常重要的意义。

具体的实验对于机器学习领域中算法的学习是极其重要的，一个好的算法不仅要有理论支撑，更重要的是要能在实际应用中表现出优异的性能，科学严谨的实验能够验证算法在实际数据集中应用结果的好坏，并且完善的验证实验可以使得算法更加具有说服力，同时可以推动算法的完善和改进。

综合以上两点内容，本文在提出基于 L21 正则化的非负矩阵分解(Multi-View Non-negative Matrix Factorization with Hierarchical Side-Information, HMNMF) 算法的基础上，进一步探索分析了不同因素对于 HMNMF 算法性能的影响，验证了该推荐算法的优异性能表现。

第二节 研究现状

推荐算法作为一种应用场景极其广泛的机器学习方法，近几年来受到了越来越多的关注，国内外已经有许多专业人士提出了用于推荐系统的相关算法。推荐算法大致可以分为三类：基于邻近的算法、基于模型的算法、两者结合的算法。其中一种比较经典的基于邻近的算法，是 Koren 提出的 NNCosNgbr 算法^[5]，旨

在通过分析商品之间的相似性，将相似商品推荐给目标用户。基于模型的方法主要是利用低秩矩阵来预测初始评分矩阵中未知的数据，其中基于非负矩阵分解（Non-negative Matrix Factorization, NMF）的算法具有突出的性能表现，不仅能够探索评分矩阵的低秩性质，还能得到用户和项目对应的隐式特征向量。

鉴于 NMF 良好的性质，之后提出了很多基于 NMF 变种的算法，例如 Singh 和 Gordon 提出的集合非负矩阵分解算法（ColNMF）^[7]、Ou 提出的多视图的非负矩阵分解算法（MultiNMF）^[8]，都是利用多视图学习，通过多个特征集合同时做矩阵分解，得到一致的用户和项目的隐式特征矩阵，从而提高算法的性能；再例如 Mashhoori 和 Hashemi 提出的层次矩阵分解（HMF）^[9]、Menon 提出的 HSCF 算法^[3]，都是在算法中加入了数据的层次结构约束，利用层次结构使得用户或项目更具有自己的特性，从而加强算法的性能。

第三节 论文的主要工作

本文主要提出了一种基于 L21 范式且带有层次结构辅助信息的多非负矩阵分解的推荐算法。算法充分利用了数据的层次结构信息，显示的将评分矩阵分为上下两层，同时利用 L21 范数对于噪声的抗干扰性，显著提高了算法的性能。通过与其他推荐算法做对比实验，进一步分析算法中不同因素对实验结果的影响，并设计实验对猜想进行验证。本文的主要工作如下：

- （1）根据数据的层次约束构建损失函数，提出了基于 L21 范式的多非负矩阵分解算法；
- （2）分析实验结果，通过 Friedman 检验、配对 T 检验、噪声实验验证 HMNMF 相比于其他算法的优异性能；
- （3）设计实验验证层次结构对算法性能的影响，探索层次约束其作用的具体方式。

第四节 论文的内容组织

本文共分为六章，各章的内容如下：

第一章，绪论。介绍了本文的研究背景和意义，分析了有关推荐系统的研究现状，并简述了本文的主要工作。

第二章，背景知识介绍。介绍了推荐系统中几个关键的概念，包括推荐系统的概念、发展及应用，协同过滤，非负矩阵分解，以及数据中的层次结构关系。

第三章，相关算法介绍。系统阐述了推荐系统中较为经典的推荐算法，介绍了基于协同过滤的推荐算法如 NNCosNgbr，基于 NMF 的推荐算法如 GNMF、MultiNMF、HMF 等。

第四章，实验准备和 HMNMF 算法介绍。介绍了实验中所用到的数据集来源以及预处理；对 HMNMF 算法的损失函数及优化公式作了介绍；并得到了不同算法在各个数据集上运行的实验结果。

第五章，实验验证与分析。对于实验结果提出一系列问题，包括 HMNMF 相比于其他算法的性能优势、层次结构对算法性能的影响等；并设计相关实验进行了验证。

第六章，总结与展望。最后一章对本文的工作做出了总结，并提出了实验过程中的优点和不足，以及未来推荐系统的发展。

第二章 背景知识介绍

第一节 推荐系统

2.1.1 推荐系统概述

推荐系统是一种可以为用户（User）提供建议的软件工具和技术，这些建议涉及到不同领域的决策过程，例如买什么东西、听哪类型的歌曲、阅读什么在线新闻等等。可以认为，推荐系统就是按一定算法规则，为用户推荐项目（Item）的过程，而这个项目的范围是很广泛的，只要涉及到需要用户做出选择的领域就会和推荐系统有所关联。但通常来讲，在一个推荐系统中只关注一种特定类型的产品，例如 CD 唱片、在线新闻等，并且根据推荐系统本身的设计、图形用户界面、用于推荐的核心推荐技术进行定制，以便为特定类型的项目提供有用并且有效的建议。

推荐系统主要针对的是网站上会提供大量的项目选择，而没有太多历史经验可供参考的用户，推荐系统就是通过分析具体问题为这些用户提出合理的建议。一个很经典的例子就是图书推荐系统，显而易见它的功能是帮助用户选择要读的书。再例如亚马逊网站就有一个推荐系统用来给每一个用户生成一个个性化的在线商店，这样用户就可以更加方便快捷的找到自己想要的商品，提高用户体验的同时可能会大额促进销售利益。上面举的这两个例子，都是用于个性化的推荐，所有不同的用户或用户组会得到不同的建议。另一方面，推荐系统还有用于非个性化的推荐，这种推荐系统的实现要简单的多，通常的应用场景是杂志或者报纸。比较典型的例子有排名前十的书籍、CD 等，本文关注的是个性化的推荐系统。

在个性化推荐系统中，最简单的形式就是给每个用户返回一个项目的排名列表。产生一个这样的项目列表，就是指推荐系统利用用户的隐式特征或一些其他的约束，来预测什么是最合适的产品或是服务。例如表 2.1 所示，系统已知部分用户对产品的评分，表格中符号“？”表示对应用户对产品的评分是未知的，并且在实际应用中，未知评分占据的比例是较大的，也就是说评分矩阵是一个极度稀疏的矩阵，推荐系统的任务就是预测矩阵中未知的评分，为每一个用户生成一

个项目推荐列表以供参考。

表 2.1 部分用户对书籍的评分矩阵

	Jane Eyre	The Goldfather	Pride and Prejudice	The Wretched
Alice	5	?	2	?
Bob	?	3	?	2
Cindy	1	?	4	?
David	?	?	?	1

2.1.2 推荐系统的发展

推荐系统的发展源于生活中一个很常见的现象：人们经常依赖他人提供的建议进行日常选择。例如在选择一本书阅读的时候，我们通常会依赖于身边人或事同行业朋友的推荐；雇主在职业招聘中依赖推荐信；我们要看电影的时候，通常会先浏览一些影评人写的电影评价。推荐系统最初就是模仿这种行为，在一个社区中的用户范围内，提供具有相似特征的用户。这样，当一个用户想要得到建议时，推荐系统就可以为他提供相似用户所选择的产品。这种方法被称作是基于用户的协同过滤，基本原理是相似用户所感兴趣的产品也应该是相似的。例如图 2.1 所示，将相似用户喜欢的项目推荐给目标用户。

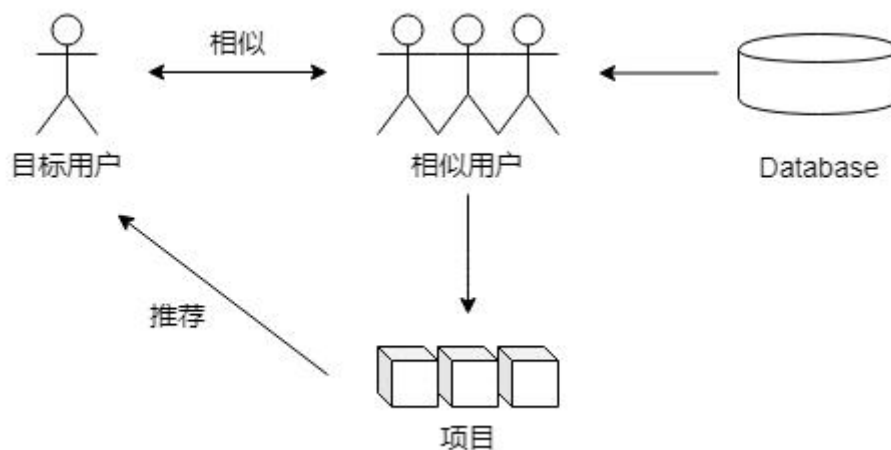


图 2.1 基于用户的协同过滤算法思想

随着电子商务网站的发展，在大量的产品中过滤信息并提供建议变得极为迫

切，因为让用户从这些网站提供的大量项目（产品或服务）中找到合适的选择是极其困难的。网络上可用信息的爆炸增长和多样性，以及新型电子商务服务（例如购买产品、产品比较、拍卖等）的快速引进，远远超过了我们自己能处理的信息容量范围。这通常会使用户不知所措，最终做出错误的决策。选择的多样性，非但没有给用户带来好处，反而降低了用户的幸福感。事实上，过多的选择会使自由、自我管理、自我决策变得过度，从而让用户感到痛苦。

近些年来，推荐系统已被证明是一种解决上文提到的信息超载问题的有效的解决方法。根据用户的请求、具体的推荐方法、用户的特征和需要等，推荐系统使用存储在数据库中的关于用户、项目、以及历史数据等各种数据生成推荐，并返回给用户。用户可能接受，也可能不接受系统的推荐，并且可能在下一个阶段提供一个显式或隐式的反馈。推荐数据库会及时存储用户的各种反馈，在下一次用户使用该系统的时候，推荐系统就可以利用先前存储的历史数据做新的推荐，从而不断推进推荐系统的更新和完善。

2.1.3 推荐系统的应用

推荐系统得到迅速发展的主要原因是电子商务网站的发展，其主要应用场景也是在电子购物网站，帮助用户在大量可用信息中筛选合适的商品，促进网站商品的营销。随着推荐系统的进一步发展，在推荐算法也应用于电影推荐、图书推荐、音乐推荐、新闻推荐等，除此之外，很多其他领域的问题也可以抽象为推荐系统的问题，例如基因-表型关联预测、响应预测等。总之，推荐系统的应用场景是极其广泛的，并且与我们的生活息息相关。

第二节 协同过滤

推荐系统如今已经发展的较为成熟，并且可以大致分为四类：基于内容的推荐系统、基于知识的推荐系统、协同过滤推荐系统以及混合推荐系统。在这四种方法中，协同过滤算法是应用最为广泛并且效果也较突出的，因此接下来我们将介绍一下协同过滤。协同过滤算法最初是在 20 世纪 90 年代被提出的，并在许多

推荐系统中得到了有效的应用。它是根据相似用户的评分爱好为每一个用户生成一个排名列表，大多数协同过滤算法都是通过寻找相似的用户，然后根据这些用户的偏好和之前的评分来预测商品的评分。

根据数据的处理方式，协同过滤技术可以进一步分为两种类型：基于模型的和基于内存的算法。基于模型的协同过滤技术旨在建立一个模型来表示用户评价数据，并使用该模型预测用户对特定项目的偏好。通常是指利用系统中用户和项目的潜在特征向量进行建模计算，例如矩阵分解算法，通过将项目和用户转换为相同的潜在因素空间，构成了一种替代方法，我们将在下一部分具体解释矩阵分解算法的过程。

另一方面，基于内存的算法使用所有的用户评级数据来预测一个项目对于一个指定用户（之前没有过该用户对这个项目评分的历史纪录）的评分，也可以被称作是基于邻近的方法。顾名思义，基于内存的技术可以分为基于用户和基于项目的协同过滤。基于用户的协同过滤是第一种自动协同过滤方法，它的工作原理是找到其他具有与目标用户相似的评级偏好的用户，并使用他们的评级来预测目标用户对相关项目的评级。与基于用户的协同过滤相比，基于项目的协同过滤是从项目的角度发展起来的，例如通过搜集项目的一些特征从而找到相似项目，这样就可以给一个用户推荐相似项目。

第三节 非负矩阵分解

提到协同过滤，就不可避免的会提到矩阵分解算法，因为基于矩阵分解算法的协同过滤被证明具有较高的准确性和可扩展性，因此矩阵分解算法几乎成为协同过滤的首选算法。并且在矩阵分解过程中，确保模型的非负性是非常重要的，例如在推荐系统的评分矩阵中，评分必须满足非负性才有实际意义。

非负矩阵分解是一种很好的用于数据降维的方法，对于高维的输入数据，将其分解成两个较低维度的矩阵乘积的形式，并且使两个分解矩阵的乘积尽可能的等于原始矩阵的数据。非负矩阵分解的原理可以用下边这个公式来表示：

$$V_{m \times n} \approx W_{m \times k} \times H_{k \times n} \quad (2.1)$$

其中， V 表示 $m \times n$ 的原始评分矩阵， m 是用户的数量， n 是项目的数量，矩

阵中的每个元素 $V_{i,j}$ 表示第 i 个用户对于第 j 个项目的评分，这个数据可能存在，也可能是空缺（就是实验中要预测的数值）； W 和 H 矩阵是两个分解得到的低维矩阵，分别表示用户和项目的隐式特征向量， k 表示特征向量的维度，通常 k 会选择比 m 或 n 都要小的数值，这样才能达到压缩数据的效果。那具体是怎么通过 W 和 H 表示评分矩阵中各个元素的评分的呢？我们通过一个具体实例进行说明：

$$\begin{bmatrix} 4 & 2 & 4 & 2 & 4 \\ 4 & 5 & 4 & 3 & 0 \\ 4 & 5 & 0 & 1 & 5 \\ 3 & 1 & 2 & 3 & 4 \\ 4 & 3 & 4 & 1 & 4 \\ 4 & 4 & 2 & 3 & 2 \end{bmatrix} \approx \begin{bmatrix} 5.1235 & 2.0760 & 2.3614 \\ 0.0448 & 5.7084 & 4.8532 \\ 6.2315 & 5.2303 & 0 \\ 5.2355 & 1.1929 & 0.5959 \\ 4.9321 & 2.9483 & 2.0466 \\ 2.7536 & 4.6727 & 1.7317 \end{bmatrix} \times \begin{bmatrix} 0.4368 & 0.0373 & 0.3870 & 0.1898 & 0.7887 \\ 0.3453 & 0.9262 & 0 & 0.1477 & 0.0344 \\ 0.3166 & 0 & 0.9043 & 0.2865 & 0 \end{bmatrix}$$

图 2.2 非负矩阵分解实例

图 2.2 中的初始矩阵是一个 6×5 大小的评分矩阵 V ，且评分范围是 1-5，我们记 V_{ij} 表示第 i 个用户对第 j 个项目的评分，即用户数量为 6，项目数量为 5。通过非负矩阵分解可以得到两个低秩矩阵 $W_{(6 \times 3)}$ 和 $H_{(3 \times 5)}$ ，分别是用户和项目的特征矩阵， W 的每一行表示每一个用户的特征向量，同理 H 的每一列表示每个项目的特征向量。通过 W 和 H 对应行、列的乘积可以计算出 V 中的每一个元素，通过矩阵乘积可以验证 $V_{ij} \approx \sum_k W_{ik} \times H_{kj}$ ，可以看出非负矩阵分解利用低秩矩阵不仅保证了数据的完整有效性，还满足了矩阵元素非负的条件，使其符合它本身的实际意义。

第四节 层次结构信息

2.4.1 数据的层次结构信息

在现实世界中，大量的数据信息是非常复杂并且是具有层次结构的^[1]，尤其是推荐系统中的项目，例如电影、书籍、音乐、商品都是具有层次分类的，例如，图 2.3(a) 和 2.3(b) 是来自 Netflix DVD 租赁页面的两个快照。

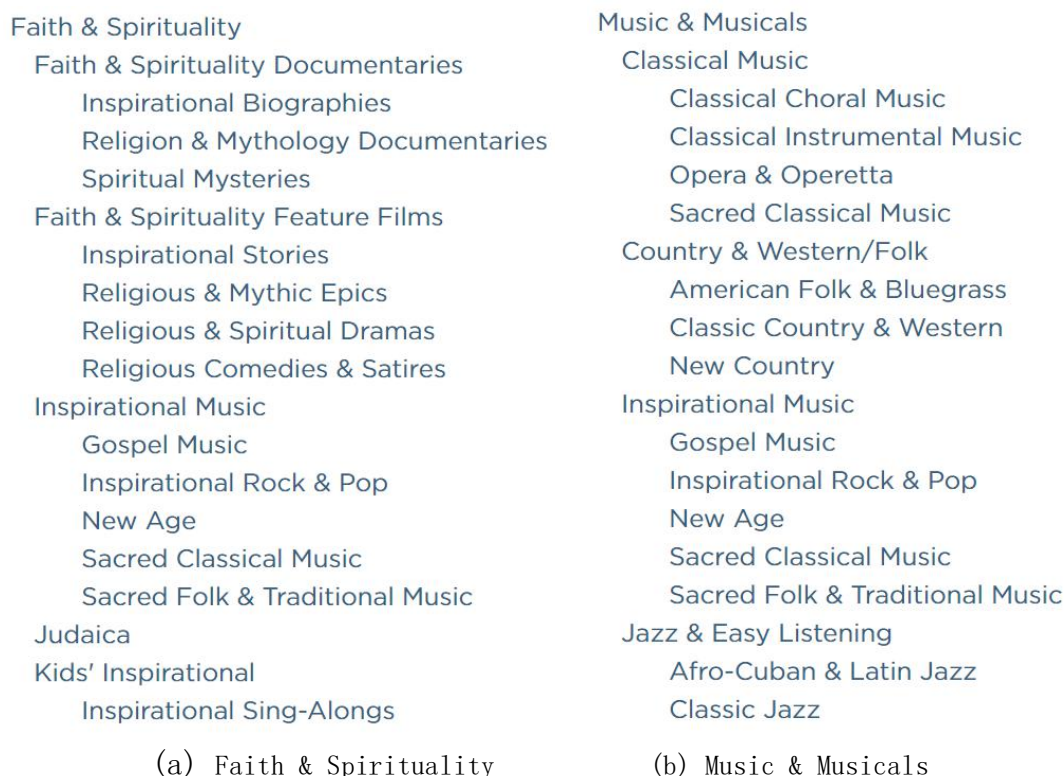


图 2.3 Netflix 网站电影层次结构实例

从图中可以看出，电影被分类为类型→子类型→详细类别的分层结构。例如，电影 *Schindler's List* 首先属于 **Faith&Spirituality** 类型，它属于子类型 **Faith & Spirituality Feature Films**，并且被进一步归类为 **Inspirational Stories**。我们还可以观察分层结构，即类别→子类别。例如，“制作自己的工作纸钟”一书属于“时钟与手表”，它是古董和收藏品的子类别。除了项目的层次结构之外，用户的偏好还呈现层次结构，例如，用户通常可能更喜欢 **Faith Spirituality** 中的电影，更具体地说，他/她在 **Inspirational Stories** 的子类别下观看电影。同样，古董钟收藏家可能对古董和收藏类别下的钟表子类别感兴趣。同一层次结构层中的项目可能共享相似的属性，因此它们可能会收到类似的评分。同样，同一层次结构层中的用户可能会分享相似的偏好，因此他们可能会对某些项目进行类似评分。

通过以上分析，可以得出结论：属于同一类别的项目理论上具有相似的属性，因此它们获得的评分也会类似；同样，属于同一层次结构层中的用户可能会具有相似的兴趣，因此他们对某些项目的评分可能会类似。因此，推荐系统可以利用项目或用户的显示层次结构来提高推荐算法的可靠性^[2]。

2.4.2 HMNMF 算法中的层次结构

在推荐系统问题中，评分矩阵的稀疏性一直是解决问题的一大难题，而数据的层次结构信息对于解决稀疏性提供了很好的帮助。因此也有很多算法利用数据的层次结构建模达到优化的目的，在此我们简单介绍两种不同的模型，来深入理解层次结构是如何被利用及优化的。

HMNMF 算法中，输入数据是一个 $m \times n$ 大小的非负评分矩阵 R ， m 和 n 分别表示用户和项目的数量。其中根据项目之间的层次结构用 $n_1 \times n_2$ 大小的 M 矩阵表示， n_1 和 n_2 分别表示 parent-level 和 child-level 层项目的数量， M_{ij} 表示第 i 个项目和第 j 个项目之间是否有直接层次关系。为了利用上述层次结构信息，原始评分矩阵被分为两层： $R_1(m \times n_1)$ 和 $R_2(m \times n_2)$ ， R_1 只包含 parent-level 对应的 n_1 个项目的评分，同理， R_2 只包含 child-level 对应的 n_2 个项目的评分，见下图。

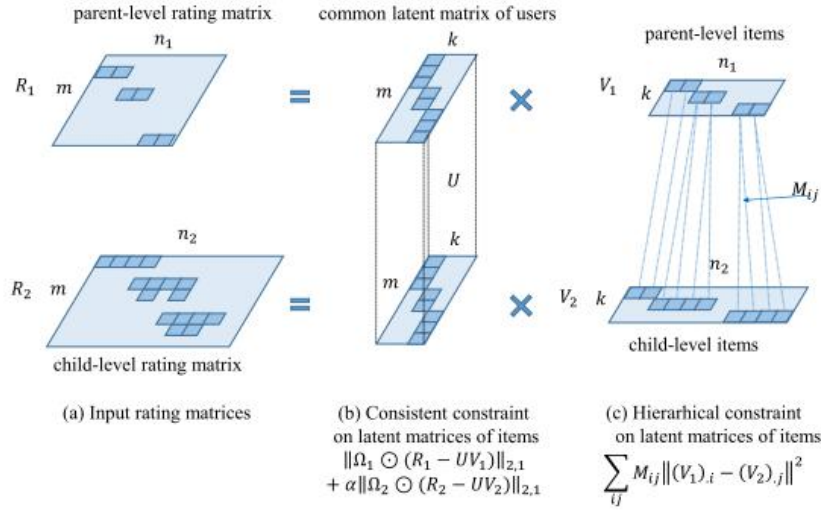


图 2.4 HMNMF 算法中的层次结构约束

之后， R_1 和 R_2 两个评分矩阵将会同时做矩阵分解。在这个算法中，我们不考虑用户之间的层次关系，因此用户的隐式特征向量使用同一个矩阵 U 表示。对于 M 矩阵中非零的元素位置，即存在父子关系的项目应该具有相似的特征向量，在优化目标公式中将会体现这一点约束。HMNMF 算法的公式如下所示：

$$\min_{U, V_1, V_2} \|\Omega_1 \odot (R_1 - UV_1)\|_{2,1} + \alpha \|\Omega_2 \odot (R_2 - UV_2)\|_{2,1}$$

$$+ \beta \sum_{ij} M_{ij} \left\| (V_1)_{(:,i)} - (V_2)_{(:,j)} \right\|^2 + \gamma \left(\|U\|_F^2 + \|V_1\|_F^2 + \|V_2\|_F^2 \right)$$

$$s.t. U \geq 0, V_1 \geq 0, V_2 \geq 0 \quad (2.2)$$

公式中第一项和第二项是对两层分解矩阵做约束，第三项是对具有层次关系的项目特征向量进行约束，使具有层次关系的项目尽可能具有相似的特征向量。

2.4.3 层次结构在响应预测问题中的应用

响应预测是指对于显示在网页中的广告，预测其被点击的概率^[3]。为什么要预测这个概率呢？是因为其中涉及到一个商业问题，对于两个主体：网页的发布者和广告商，广告商想要在某个网页上登自己的广告，就必须给网页所有者支付一定的费用，网页发布者的收益等于广告商的出价乘以点击次数。这个时候对网页发布者而言，广告点击次数的预测就显得极为重要，因为他要选择一个使自己收益最大的广告。在响应预测问题中，我们用预测点击率（Click-Through Rate, CTR）来评价预测的效率。

响应预测问题同样可以抽象成推荐系统中评分矩阵的补全问题——将网页类比用户，广告类比项目，矩阵中的预测点击率类比为评分。在这个问题中，所面临的一个难题同样是原始数据的极端稀疏性，利用网页和广告的层次结构信息，可以改善每个网页和广告的隐式特征向量，同时克服数据的极端稀疏性。

以广告为例，下图展示了响应预测问题中数据的层次结构：

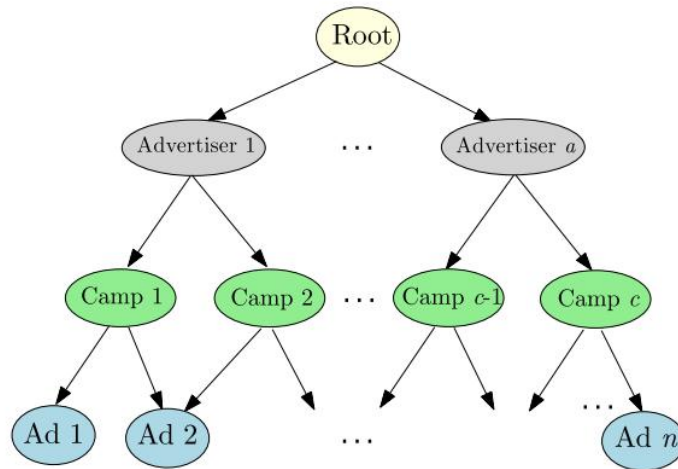


图 2.5 响应预测问题中广告的层次结构

如图所示是一个树形的层次结构，只有一个根节点 **root**，对于一个广告商 **advertiser** 会发布不同领域（**campaign**）的广告，不同领域同样会有不同产品的广告。属于同一领域的广告应该具有相似的特征向量。在这个算法中，具体分三个步骤对层次关系进行了优化，通过正则化、向上聚合、残差拟合，使层次结构得到了充分利用，达到了算法优化的目标。

第三章 相关算法介绍

根据以上分析可以知道，推荐算法主要分为基于邻近的算法、基于模型的算法和两者混合的算法。本文我们主要介绍前两种推荐算法。

第一节 基于协同过滤的推荐算法

基于邻近的算法又可以分为基于相似用户和基于相似项目的邻近算法，其中由于项目的独特性、多样性，它的特征是比较容易收集的，而用户特征大多数为隐式的，很难收集并处理，另外，基于项目相似的推荐算法在处理数据上时间复杂度相对较小，因此基于项目相似的推荐算法具有更广泛的应用。其中较为经典的是 CorNgbr^[4] 和 NNCosNgbr，其主要思想是将相似项目推荐给目标用户。

3.1.1 非标准化余弦邻域算法

NNCosNgbr 是一种基于项目的邻域算法，它采用余弦相似性作为距离测量。用户 u 对项目 i 的评分 $\hat{r}_{u,i}$ 的计算公式如下：

$$\hat{r}_{u,i} = b_{u,i} + \sum_{j \in J_i} s_{i,j} (r_{u,j} - b_{u,j}) \quad (3.1)$$

其中， $s_{i,j}$ 表示项目 i 和项目 j 之间的余弦相似度； J_i 表示项目 i 的邻域； $r_{u,i}$ 表示用户 u 对项目 j 的评分； $b_{u,i}$ 表示用户 u 和项目 i 之间的基线估计，其计算公式如下：

$$b_{x,i} = \mu + b_x + b_i \quad (3.2)$$

其中， μ 表示评分矩阵中所有评分的平均值； b_x 表示用户 x 的评分偏差，指用户 x 的所有评分均值与 μ 之间的偏差； b_i 表示项目 i 的评分偏差，指项目 i 的所有评分均值与 μ 之间的偏差。这一项旨在考虑用户 x 和项目 i 各自的评分特性，提高算法的效率。

3.1.2 基于协同过滤算法的不足

基于用户的和基于项目的推荐算法都是使用较为广泛的协同过滤推荐算法，并且具有较好的推荐准确性。但这种算法还存在着很大的不足。

首先，基于用户的协同过滤算法往往会优先选择将比较热门的产品推荐给用户，然而这不是我们想看到的，对于推荐系统，我们要着重解决的是 80% 的 “the long tail”（长尾商品），这就需要全面而充分的挖掘用户的兴趣特征，将长尾商品准确的推荐给用户；而且更重要的一点，基于用户的协同过滤对于用户数量很大的推荐系统，计算任务是极其繁重的，代价很大。

另外，基于项目的协同过滤算法计算量没有那么大，但还是存在着缺点，基于项目的协同过滤中，只会把用户喜欢的项目的相似项目推荐给用户，这使得推荐系统有很大的局限性，对于用户没有了解过的项目就会“石沉大海”，假设用户在购买了一件商品后，大概率是不会重复购买同一类商品的，因此基于项目的协同过滤算法在很多场景中是不适用的。

第二节 基于非负矩阵分解的推荐算法

基于模型的推荐算法中，NMF 是一种应用最广泛，并且实验效果和性能优于其他很多算法的推荐算法。NMF 最初是由 Lee 和 Seung 在 1999 年提出的^[6]，旨在通过低秩矩阵探索用户和项目的特征向量，从而更好的预测评分。NMF 良好的性能使得之后提出了很多基于 NMF 变种的推荐算法，这些算法利用了数据集的不同特征结构，使得算法具有更优越的性能。

首先一类是利用数据的多视图特征，通过多个特征集合同时做矩阵分解，得到一致的用户和项目的隐式特征矩阵，从而提高算法的性能，例如 Singh 和 Gordon 提出的集合非负矩阵分解算法（ColNMF）、Ou 提出的多视图的非负矩阵分解算法（MultiNMF）；再例如 Mashhoori 和 Hashemi 提出的层次矩阵分解（HMF）、Menon 提出的 HSCF 算法，都是在算法中加入了数据的层次结构约束，利用层次结构使得用户或项目更具有自己的特性，从而加强算法的性能；还有一类是针对数据内部包含几何结构的，例如有些数据具有图的网状结构，这是算法中可以利用的很重要的信息，由蔡登教授提出的 GNMF 算法^[10]，就是利用

数据内部的图约束来提高算法的性能。

接下来我们对两个经典的基于 NMF 变种的推荐算法做详细介绍。

3.2.1 多视图的非负矩阵分解

在实际应用中，图像、音乐或商品都可以由多视图特征表示，例如对于网页而言，可以通过网页中图像的视觉特征来描述网页，也可以用网页中的文本特征来描述网页，不同特征视图提供的信息是具有一致性的，通过探索不同视图的一致性和互补性，多视图算法具有更好的泛化能力。多视图的非负矩阵分解是对于数据具有多个视图特征的系统，对不同视图的评分矩阵同时做矩阵分解，并利用一定的约束条件将从多个视图学习到的潜在特征向量达到一致。其优化目标如下：

$$\begin{aligned} \min \sum_{i=1}^n \|\Omega_i \circ (R_i - U_i V_i)\|_F^2 + \sum_{i=1}^n \alpha_i \|U_i - U^*\|_F^2 \\ \text{s.t. } U^* \geq 0; \forall k, \|(U_i)_{(k,:)}\|_1 = 1; \forall i, U_i \geq 0, V_i \geq 0 \end{aligned} \quad (3.3)$$

其中， $(R_i)_{m \times n_i}$ 表示第 i 层视图中包含 m 个用户、 n_i 个项目的评分矩阵； U_i 和 V_i 分别是第 i 层视图中分解的用户和项目的潜在特征矩阵； U^* 表示多个视图中用户的共识潜在特征矩阵；公式中 $\|(U_i)_{(k,:)}\|_1 = 1$ 这一项归一化约束旨在消除结果的方差。Multi-View NMF 通过利用隐藏在多个视图中的潜在用户结构，一定程度上提高了算法的性能。

3.2.2 图正则化的非负矩阵分解

在很多情况下，数据空间具有内部几何和区分结构，例如有些数据具有图的网状结构，这在实际应用中是极为重要的信息，GNMF 算法是由蔡登教授等人提出的，旨在通过利用数据内部的图约束，在矩阵分解的同时保持数据内部的结构关系，提高算法性能和推荐系统的准确率。GNMF 使用拉普拉斯约束使样本中距离较近的点在新的空间坐标下同样还是临近点，即保持数据在空间上的结构关系。

GNMF 算法的优化目标公式如下：

$$O_F = \|X - UV^T\|_F^2 + \lambda \text{Tr}(V^T L V)$$

$$s.t. U \geq 0, V \geq 0 \quad (3.4)$$

其中第一项使矩阵分解中 F 范数约束，第二项是拉普拉斯约束，L 是拉普拉斯矩阵，它的定义公式如下：

$$L = D - R \quad (3.5)$$

其中，D 是对角矩阵，且 D 的计算公式如下：

$$D_{ii} = \sum_j R_{ij} \quad (3.6)$$

R 表示权重矩阵，利用原始数据的图结构中点间距离进行计算，记 $N_p(x_i)$ 为 x_i 的 p 个临近点，则 R 的定义公式如下：

$$R_{ij} = \begin{cases} e^{-\frac{\|x_i - x_j\|^2}{\sigma}}, & x_i \in N_p(x_j) \text{ 或 } x_j \in N_p(x_i) \\ 0, & \text{其它} \end{cases} \quad (3.7)$$

3.2.3 基于 NMF 的推荐算法的优势

相比于协同过滤算法，NMF 算法直接从分解出的矩阵中学习得到预测评分的结果，从而能够在全局的角度充分挖掘用户和项目潜在的特征向量，从而显著提高了算法的性能。而且，NMF 算法实现过程中计算消耗的代价很小，一般只需要迭代 50~100 次就可以使矩阵趋向于稳定，NMF 算法的灵活性又使得在 NMF 算法基础上，可以有效利用数据的结构特征改善算法，从而提高算法性能。

第四章 实验准备和 HMNMF 算法

第一节 实验准备

4.1.1 数据集介绍

本次实验中用到的数据集主要有两类，第一类是 Douban Book 和 Yahoo Music 数据集，这两个数据集中的元素都是用户对项目的评分，并且评分范围都是 1~5 的整数；第二类是从 KEGG 网站下载的 Pathway 数据集，Pathway 用 0 或 1 来表示基因和表型之间有无连接。

Yahoo Music 数据集中的元素是用户对音乐的评分，评分范围都是 1~5 的整数，其中用户数量大约有 20 万个，音乐数量大约有 13 万首。数据集中的音乐具有一个树状的层次结构，包括根节点共有三层，剩下两层中，上层表示的是较为抽象的类型，下层表示的是在父节点类型范围中更加具体的类型。由于 Yahoo Music 数据集的数据量太大，我们在其中选取了一部分作为实验数据，包含 2652 个用户对 3831 首音乐的评分，其中有 2313 首音乐是来自第二层，剩下 1518 首来自第三层。利用 MATLAB 对数据进行处理后，得到了两个层次对应的初始评分矩阵 $(A_1)_{2652 \times 2313}$ 和 $(A_2)_{2652 \times 1518}$ ，分别表示选中的用户对上层音乐和下层音乐的评分，同时还有 logical 类型的矩阵 $(\Omega_1)_{2652 \times 2313}$ 和 $(\Omega_2)_{2652 \times 1518}$ 表示对应评分是否存在；另外根据实验数据的层次关系构建了上下两层音乐的关联矩阵 $M_{2313 \times 1518}$ 。

Douban Book 数据集中的元素是指用户对书籍的评分，类似于 Yahoo Music 数据集，书籍具有层次结构，评分范围都是 1~5 的整数。实验数据中用户数量是 1040，书籍的总数量是 4608，其中有 917 个书籍来自上层，剩余 3691 个书籍来自第二层。利用 MATLAB 对数据进行预处理，为了便于实验，我们用相同的变量存储数据，包括两个层次中的初始评分矩阵 $(A_1)_{1040 \times 917}$ 和 $(A_2)_{1040 \times 3691}$ ，以及对应的 0-1 矩阵 $(\Omega_1)_{1040 \times 917}$ 和 $(\Omega_2)_{1040 \times 3691}$ 用于表示对应用户和书籍之间是否具有评分数值；同样保存了矩阵 $M_{917 \times 3691}$ 用于表示实验数据中上下层书籍之间的层次关联关

系。

KEGG Pathway 数据集中的元素是基因与表型之间的关联关系，把他抽象成一个推荐系统，则实验的任务就是为了预测每一个基因和每一个通路之间的关联关系。其中基因数量为 3280，表型大概有 20000 个，这些表型之间存在一个层次结构，共包含 16 层，但不完全是树状的层次结构，例如某些节点具有多个父节点，每个父节点也会有多个孩子节点。在本次实验中，根据层次关系，将表型项最多、信息较为丰富的两层抽取出来作为实验数据，也就是第七层和第八层的基因-表型的关联信息。其中，基因数量是 3280，上下两层的表型数量分别是 2241 和 3707，我们把基因抽象成评分矩阵中的用户，表型抽象成项目，算法的作用是要预测基因和表型之间的关联。经过 MATLAB 对数据的处理，类似于其他两个数据集，得到了上下两层的基因-表型关联矩阵 $(A_1)_{3280 \times 2241}$ 和 $(A_2)_{3280 \times 3707}$ ，同样有表示上下两层之间表型的关联矩阵 $M_{2241 \times 3707}$ 。

以下是对三个数据集中各个元素数量的统计结果。

表 4.1 各个数据集的具体描述

	Yahoo Music	Douban Book	Pathway
用户数量	2652	1040	3280
项目数量 (parent-level)	2313	917	2241
项目数量 (child-level)	1518	3691	3707
已知评分数量	486167	5351	48537
层次关联数量	989301	9394	2707
稀疏度 (parent-level)	0.0481	0.0011	0.0025
稀疏度 (child-level)	0.2818	0.0028	0.0003

4.1.2 对比算法的选定

为了使实验更具有说服力，实验中选取了一些比较经典的推荐算法作参考，以此来对比分析算法性能的优劣。实验中一共选择了以下 6 种算法：ColNMF, MultiNMF, HMF, GNMf, TMC, NNCosNgbr。

其中，前四种基于非负矩阵分解的算法，集合非负矩阵分解（ColNMF）和多视图的非负矩阵分解（MultiNMF）都是针对多视图学习，通过多个特征集合达到数据融合或数据集成的效果，对不同视图的评分矩阵同时做矩阵分解，并利用一定的约束条件将从多个视图学习到的潜在特征向量达到一致，从多个视图探索评分矩阵的性能从而提高推荐系统的性能。层次矩阵分解（HMF）将数据的层次结构信息加入到矩阵分解中，例如 child-level 层的商品的特征向量是由它自己和它双亲节点的特征向量来表示的；图正则化的非负矩阵分解（GNMF）则利用了数据中的图约束信息来提升算法性能。

除此之外，还选取了两个推荐系统中十分经典的算法作为基本的度量标准：低秩矩阵分解（TMC）和非标准化余弦邻近算法（NNCosNgrbr）。

最后选择基于 Frobenius 范式的 HMNMF-F 算法，和本文中我们提出的 HMNMF 做对比，探究 L21 范式对噪声的消除作用以及对算法性能的提升作用。

第二节 HMNMF 算法

4.2.1 损失函数

HMNMF 算法的主要思想是根据 item 之间的层次关系，将评分矩阵显式地分解为两层： R_1 和 R_2 ，同时对两个评分矩阵做矩阵分解，并且根据层次约束使具有层次关系的 item 具有相似的特征向量，其优化目标公式为：

$$\begin{aligned} \min_{U, V_1, V_2} & \left\| \Omega_1 \circ (R_1 - UV_1) \right\|_{2,1} + \alpha \left\| \Omega_2 \circ (R_2 - UV_2) \right\|_{2,1} \\ & + \beta \sum_{ij} M_{ij} \left\| (V_1)_{(:,i)} - (V_2)_{(:,j)} \right\|^2 + \gamma \left(\|U\|_F^2 + \|V_1\|_F^2 + \|V_2\|_F^2 \right) \\ s.t. & U \geq 0, V_1 \geq 0, V_2 \geq 0 \end{aligned} \quad (4.1)$$

其中， R_1 、 R_2 分别表示 parent-level 和 child-level 的初始评分矩阵，大小分别为 1046×917 、 1046×3691 ； U 矩阵表示分解得到的用户特征矩阵； V_1 、 V_2 分别表示两层商品的特征向量； Ω_1 、 Ω_2 是两个 0-1 矩阵，标识初试评分矩阵中是否有历史评分数据； M_{ij} 是表示层次关系的矩阵，其大小是第一层商品数量 $n_1 \times$

第二层商品数量 n_2 ； $\|X\|_{2,1} = \sum_i \sqrt{\sum_j X_{ij}^2}$ 表示矩阵 X 的 L21 范式。

4.2.2 公式优化

在公式 4.1 中所展示的损失函数中，L21 范式是不可导的，无法使用常规的求导梯度下降的方式优化，我们采用增广拉格朗日乘子法（Lagrangian Multiplier Method, ALM）进行优化，ALM 方法是求解最优化问题常用的一种方法。

首先定义两个变量 E_1 、 E_2 ，令 $E_1 = \Omega_1 \circ (R_1 - UV_1)$ 、 $E_2 = \Omega_2 \circ (R_2 - UV_2)$ ，同时将公式 4.1 中的第三项转换为矩阵的形式，则上述优化目标函数可以写成：

$$\begin{aligned}
 & \min_{U, V_1, V_2, E_1, E_2} \|E_1\|_{2,1} + \alpha \|E_2\|_{2,1} \\
 & + \beta (\text{tr}(V_1 D_1 V_1^T) + \text{tr}(V_2 D_2 V_2^T) - 2\text{tr}(V_1 M V_2^T)) \\
 & + \gamma (\|U\|_F^2 + \|V_1\|_F^2 + \|V_2\|_F^2) \\
 & s.t. E_1 = \Omega_1 \circ (R_1 - UV_1), E_2 = \Omega_2 \circ (R_2 - UV_2) \\
 & U \geq 0, V_1 \geq 0, V_2 \geq 0
 \end{aligned} \tag{4.2}$$

之后利用 ALM 方法，增加几个拉格朗日乘子，并定义

$B = \Omega_1 \circ (R_1 - UV_1) + \frac{1}{\mu} \lambda_1$ ，可得到 E_1 、 E_2 的更新公式如下：

$$(E_1)_{(i,:)} = \begin{cases} \left(1 - \frac{1}{\mu \|B_{(i,:)}\|}\right) B_{(i,:)} & \text{if } B_{(i,:)} \geq \frac{1}{\mu} \\ 0 & \text{else} \end{cases} \tag{4.3}$$

$$(E_2)_{(i,:)} = \begin{cases} \left(1 - \frac{\alpha}{\mu \|B_{(i,:)}\|}\right) B_{(i,:)} & \text{if } B_{(i,:)} \geq \frac{\alpha}{\mu} \\ 0 & \text{else} \end{cases} \tag{4.4}$$

同样可以得到 U 、 V_1 、 V_2 的更新公式如下：

$$\begin{aligned}
 \nabla(U) &= 2 \left(\Omega_1 \circ \left(\Omega_1 \circ (UV_1 - R_1) + E_1 - \frac{1}{\mu} \lambda_1 \right) \right) V_1^T \\
 &+ 2\alpha \left(\Omega_2 \circ \left(\Omega_2 \circ (UV_2 - R_2) + E_2 - \frac{1}{\mu} \lambda_2 \right) \right) V_2^T + 2\gamma U
 \end{aligned} \tag{4.5}$$

$$\begin{aligned}\nabla(V_1) = & \mu U^T \left(\Omega_1 \circ \left(\Omega_1 \circ (UV_1 - R_1) + E_1 - \frac{1}{\mu} \lambda_1 \right) \right) \\ & + 2\beta(V_1 D_1^T - V_2 M^T) + 2\gamma V_1\end{aligned}\quad (4.6)$$

$$\begin{aligned}\nabla(V_2) = & \mu U^T \left(\Omega_2 \circ \left(\Omega_2 \circ (UV_2 - R_2) + E_2 - \frac{1}{\mu} \lambda_2 \right) \right) \\ & + 2\beta(V_2 D_2^T - V_1 M^T) + 2\gamma V_2\end{aligned}\quad (4.7)$$

HMNMF 算法实现的主要代码如下：

```
for i=1:MaxIter
    disp([' Iter= ' num2str(i)]);
    %update E1
    B=Y1.*(A1-G*P1)+(1/miu)*Lambdal;
    for row=1:size(E1,1)
        if norm(B(row,:))>=1/miu
            E1(row,:)=(1-1/(miu*norm(B(row,:))))*B(row,:);
        else
            E1(row,:)=0;
        end
    end
    %update E2 /*...*/
    %update G
    B=Y1.*(E1-(1/miu)*Lambdal)*P1'+alpha*Y2.*(E2-(1/miu)*Lambdal2)*P2';
    B_plus=(abs(B)+B)/2;
    B_minus=(abs(B)-B)/2;
    Delta_plus=Y1.*(G*P1)*P1'+alpha*Y2.*(G*P2)*P2'+gamma*G+B_plus;
    Delta_minus=Y1.*A1*P1'+alpha*Y2.*A2*P2'+B_minus;

    G=G.*(Delta_minus./Delta_plus);
    G(isnan(G))=0;
    G(isinf(G))=0;
    %update P1
    B=miu*G'*(Y1.*(E1-(1/miu)*Lambdal));
    B_plus=(abs(B)+B)/2;
    B_minus=(abs(B)-B)/2;
    Delta_plus=miu*G'*(Y1.*(G*P1))+2*beta*P1*D1+2*gamma*P1+B_plus;
    Delta_minus=miu*G'*(Y1.*A1)+2*beta*P2*M'+B_minus;

    P1=P1.*(Delta_minus./Delta_plus);
    P1(isnan(P1))=0;
    P1(isinf(P1))=0;
```

```

%update P2  /*...*/
%update ALM parameters
Lambda1=Lambda1+miu*(Y1.*(A1-G*P1)-E1);
Lambda2=Lambda2+miu*(Y2.*(A2-G*P2)-E2);
miu=min(1, miu*rou);
t = t+1;

[L(t, 1), L(t, 2)]=objective(alpha, beta, gamma, miu, A1, A2, G, P1, P2, M, Y1, Y2, E1, E2, Lambda1, Lambda2);
disp(['L=' num2str(L(t, 1))]);
end

```

第三节 实验结果

如上所示，本次实验分别在 Douban Book、Yahoo Music、Pathway 数据集上进行，并且每组实验除 HMNMF 算法外选取了 9 个对比实验。实验过程包括具体的调参实验以及模型学习等，最终得到测试集中对每个 user-item 的预测评分，根据一定的评价指标比较各个算法的性能。

4.3.1 评价指标的选择

本次实验选择了 RMSE、HR、ARHR 作为评价指标。以下是对这三个指标的简要介绍。

众所周知，均方根误差（标准误差）RMSE 是最经典的用于评估预测值与真实值之间误差大小的全局评价指标，但 RMSE 将系统中的所有项目等量齐观，忽略了 Top-N 推荐系统中前 N 项的重要性，因此除 RMSE 外，又选择了 HR 和 ARHR 指标衡量 Top-N 的性能^[11]。

HR 和 ARHR 是根据预测得到的每个用户的前 N 个项目推荐列表进行计算的，因此可以更加直接的展现 Top-N 推荐系统的性能。HR（命中率）的计算公式如下：

$$HR = \frac{1}{n} \sum_i^n \frac{(\#hit)_i}{(\#users)_i} \quad (4.8)$$

其中， n 表示系统中项目的数量； $(\#hit)_i$ 表示测试集中有多少个用户的 Top-N

列表中包含第 i 个项目，即项目 i 被命中的次数； $(\#users)_i$ 表示第 i 个项目在测试集中一共出现在多少个用户的 Top-N 列表中，即出现的总次数。从公式中可以看出， HR 指标衡量的是召回率，并且值越大越好，但 HR 的缺点是没有考虑项目在 Top-N 列表中的具体排名，因此我们又选用 $ARHR$ 来作进一步评估， $ARHR$ 对于 Top-N 列表中排名更靠前的项目占比更大，其计算公式如下：

$$ARHR = \frac{1}{n} \sum_i^n \frac{1}{(\#users)_i} \sum_{j=1}^{(\#hits)_i} \frac{1}{p_{ij}} \quad (4.9)$$

其中， p_{ij} 表示项目 i 在用户 j 的 Top-N 列表中的排名序列。

4.3.2 实验结果展示

以下各表展示了不同算法在三个数据集上的结果。

表 4.2 YahooMusic 数据集结果

评价指标	RMSE	HR	ARHR
NNCosNgbr	2.8189	0.3584	0.1085
TMC	2.6573	0.3264	0.1122
MultiNMF	1.2546	0.4040	0.1436
ColNMF	1.2478	0.4282	0.1360
GNMF	1.1502	0.4500	0.1323
HMF	1.2375	0.4281	0.1328
HMNMF-F	1.0811	0.4298	0.1361
HMNMF	1.0699	0.5088	0.1499

表 4.2 DoubanBook 数据集结果

评价指标	RMSE	HR	ARHR
NNCosNgbr	4.2606	0.1009	0.0785
TMC	4.0471	0.0342	0.0315
MultiNMF	3.3001	0.0141	0.0129
CoINMF	3.3333	0.0430	0.0380
GNMF	1.1689	0.1202	0.0893
HMF	3.3356	0.0430	0.0380
HMNMF-F	1.1511	0.1202	0.0894
HMNMF	1.1506	0.1202	0.0894

表 4.3 Pathway 数据集结果

评价指标	RMSE	HR	ARHR
NNCosNgbr	1.2109	0.0251	0.0246
TMC	0.9942	0.2000	0.1417
MultiNMF	0.1571	0.2246	0.1394
CoINMF	0.1571	0.2246	0.1394
GNMF	0.1225	0.2323	0.1425
HMF	0.1558	0.2249	0.1420
HMNMF-F	0.1225	0.2322	0.1444
HMNMF	0.1213	0.2299	0.1472

从实验结果可以看出，HMNMF 算法在各个数据集上的表现都比较突出，在三项指标上都优于其他算法，HMNMF 相比于最经典的基于邻近的算法 NNCosNgbr 性能提升了 60%以上，相对于 NMF 变种算法性能提升也都在 20%左右。但某些指标上，例如 HMNMF 在和 HMNMF-F 比较时，例如 Douban Book 数据集中的结果“1.1506-0.1202-0.0894”与“1.1511-0.1202-0.0894”相比没有突出的性能表现，对此我们将在下一章节具体讨论分析和实验验证。

第五章 实验验证与分析

第一节 L21 范数健壮性的验证

首先对于上一章节所得到的实验结果，我们提出以下几个问题：

问题 1：HMNMF 算法相比于其他算法到底有没有突出的优异性能表现？

问题 2：HMNMF 算法相比于 HMNMF-F 算法的评价指标没有很好的提升，这是不是说明 L21 范式的 HMNMF 算法和 F 范式的 HMNMF-F 算法没有明显的差异？

问题 3：若问题 2 是肯定的结果，那么 L21 范式还有作用吗？

5.1.1 Friedman 检验

对于问题一，我们结合上一章的实验结果分析，HMNMF 算法在多个数据集、多个性能度量指标下的都具有明显的优势，但这只是我们直观的感受，根据评价指标数值的大小比较，就可以得出某个学习器性能更优的结论吗？显然这种方式是不够科学严谨的。首先判断一个算法的好坏是根据它的泛化性能，而不仅仅是在实验测试数据集下得到的实验结果，实验中得到的结果只是在某一个特例下的结果，不能够完全代表该算法的泛化性能。另外，很多机器学习算法具有明显的随机性，例如本文中的 HMNMF 算法，用户和项目的特征矩阵是随机初始化得到的，之后的梯度下降过程也是具有随机性的，更何况还有实验参数、测试集的影响，会使算法每次运行得到的结构都不一样。那么有没有科学的检验方法判断各个算法性能的好坏呢？

这里我们介绍一下统计假设检验，统计假设检验从统计学的角度提供了一种科学的对算法性能作比较的方法。统计假设检验解决的问题是：假设在特定数据集下的实验测试结果显示算法 A 的性能优于算法 B，则在统计意义上算法 A 的泛化性能是否也优于算法 B？以及得到这个结论的概率有多大？

在本次实验中，我们采用 Friedman 检验来验证 HMNMF 算法是否具有突出的性能表现，Friedman 检验是基于各个算法的性能排序的，例如对本次实验采用

的三个数据集、九个算法，根据每个算法在每个数据集下得到的结果计算 RMSE 评估标准，根据 RMSE 进行排序得到表 5.1。其中，排序 1 表示该算法性能最好，2 其次，以此类推；对于 RMSE 相同的算法，对排名取平均值；最后对每个算法的序值求其平均值，得到每个算法的平均序值。

表 5.1 各个算法对于 RMSE 结果的排序

	HMNMF	HMNMF-F	HMF	GNMF	ColNMF	MultiNMF	TMC	NNCosNgbr
Yahoo Music	1	2	3	4	5	6	7	8
Douban Book	1	2	6	3	5	4	7	8
Pathway	1	2.5	4	2.5	5.5	5.5	7	8
平均序值	1	2.167	4.333	3.167	5.167	5.167	7	8

Friedman 检验是为了验证这些算法的性能是否相同，若各个算法的性能相差不大，则他们的平均序值 r_i 应该一致。假设数据集的数量为 N ，算法数量为 k ，则 r_i 在理想状态下应该服从正态分布，公式 5.1 中定义的变量 τ_{χ^2} 应服从自由度为 $k-1$ 的 χ^2 分布。

$$\tau_{\chi^2} = \frac{k-1}{k} \cdot \frac{12N}{k^2-1} \sum_{i=1}^k \left(r_i - \frac{k+1}{2} \right)^2 = \frac{12N}{k(k+1)} \left(\sum_{i=1}^k r_i^2 - \frac{k(k+1)^2}{4} \right) \quad (5.1)$$

Friedman 检验中通常利用公式 5.2 中定义的变量 τ_F 做计算， τ_F 服从自由度为 $k-1$ 和 $(k-1)(N-1)$ 的 F 分布，将计算结果与标准临界值比较判断假设是否正确。实验中我们输入各个算法在不同数据集下的实验结果，利用 MATLAB 中的 `friedman()` 函数直接对结果做 F 检验，表 5.3 分别是对三个评价指标下做 F 检验的结果。

表 5.2 F 检验结果

	RMSE	HR	ARHR
Prob>Chi-sq	0.0053	0.0328	0.0609

我们做零假设 H_0 ：所有算法性能相同。则 F 检验的结果表示实验统计数

据符合零假设的概率， p 值越小对零假设的有效性产生怀疑。结果显示，“所有算法性能相同”这个假设是不成立的，也就是说，HMNMF 算法相较于其他算法有显著的差异。

至此，我们已经解决了本节开头提出的问题一，用科学严谨的统计假设检验方法验证了 HMNMF 相对于对比算法在泛化性能上的优异表现。

5.1.2 配对 T 检验

Friedman 检验很好的解决了问题一，但 F 检验只是利用了各个算法的结果排名，而不是具体数值，再具体观察实验结果可以注意到，对于 F 范式的 HMNMF-F 和 L21 范式的 HMNMF 结果并没有相差太多，针对这一现象我们提出了问题二，期望能对特定的两个算法做比较实验，并且要验证两个算法的结果是否有明显的差异。对此设计如下实验进行验证：对这两个算法的实验结果做配对 T 检验，验证 HMNMF 与 HMNMF-F 是否具有明显的差异。

首先简要解释一下为什么选择配对 T 检验。T 检验在整体上是为了检验两个样本均值是否相同、是否来自同一分布，或者说是否具有明显的差异。T 检验可以大致分为三类：独立样本 T 检验、双样本 T 检验、配对 T 检验。独立样本 T 检验只有一个样本，假设样本的均值为 \bar{X} ，已知总体均值为 μ ，我们想要通过 T 检验判断这个样本是否来自这个总体，也就是说这个样本来自的总体和均值为 μ 的总体是否有显著的差别，首先我们要做出零假设和对立假设，一般零假设是假设样本与总体之间没有显著差异，对立假设可以是 $\bar{X} \neq \mu$ 、 $\bar{X} \geq \mu$ 或 $\bar{X} < \mu$ 。确定置信水平后，根据计算的 t 值与 t 临界值相比较，判断 t 值所在区域从而判断是否接受零假设。双样本 T 检验是用于判断两个独立的样本来自的总体是否有显著的差异；而配对 T 检验是针对两个互相关联的样本之间的比较，例如不同算法在相同数据集上运行的结果、同一测试集在前后不同时间点得到的数据等，显而易见本次实验中应选用配对 T 检验进行验证。

在本次实验中，两个配对样本指的是两个不同的算法在同一测试集下得到的多个指标值，配对 T 检验需要先计算出这些样本 (X_i, Y_i) 之间的差异 $D_i = X_i - Y_i$ ，之后的工作相当于对 D_i 做独立样本 T 检验。我们对 HMNMF 和 HMNMF-F 在不

同数据集上的结果做配对 T 检验，检验这两个算法的结果是否具有明显的差异，做零假设 H_0 ：两个算法得到的结果没有显著差异。下表是不同置信水平下的实验结果：

表 5.3 配对 T 检验结果

	概率	$\alpha=0.05$	$\alpha=0.1$	$\alpha=0.4$
YahooMusic	0.4182	0.0000	0.0000	0.0000
DoubanBook	0.4226	0.0000	0.0000	0.0000
Pathway	0.8941	0.0000	0.0000	0.0000

表中概率一列表示否定零假设的概率，从实验结果可以看出，在 0.05 甚至 0.4 的置信度下，HMNMF 与 HMNMF-F 算法都没有明显的差异，这说明我们的假设是正确的，问题二的答案是肯定的。由此引出了前面提过的问题三——这是否说明 L21 范式相比于 F 范式没有对算法做出较大的性能提升？对此我们又根据理论依据提出假设并设计相应的实验进行验证。

5.1.3 高斯噪声检验

根据以上分析，我们需要设计实验验证 L21 范数对于噪声干扰的健壮性。在推荐系统的初始评分矩阵 A_1 、 A_2 中随机添加一定量的高斯噪声，借助 MATLAB 中的 `normrnd()` 函数产生同等大小数组的高斯噪声，并将其添加到评分矩阵中。实验中选择均值为 0，标准差为 [0.1, 1, 5, 10] 的不同的噪声，并且实验只包含了 HMNMF 算法和 HMNMF-F 算法作为对比，以验证 L21 范数对噪声的抗干扰性。

在不同噪声下，HMNMF 和 HMNMF-F 的性能表现如图 5.1 所示：

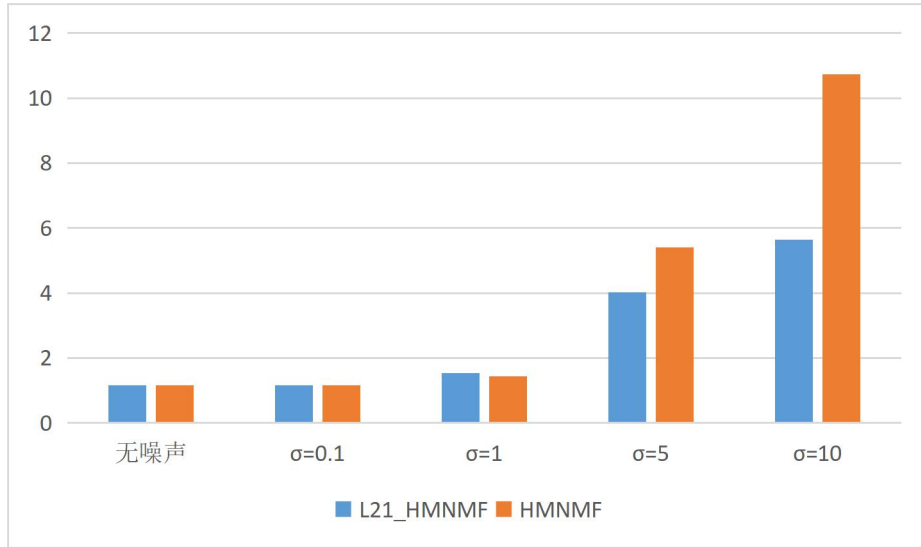


图 5.1 DoubanBook 数据集中不同噪声下的 RMSE 结果

由图可见,随着噪声标准差的增加,HMNMf 算法的性能明显优于 HMNMf-F 算法,这说明 L21 范数确实能在一定程度上消除噪声的干扰,对于实际应用中噪声较大的数据,HMNMf 算法具有更稳定的性能。

第二节 层次结构的作用

在 HMNMf 算法中,有效的利用了数据的层次结构约束,但如何验证层次约束在实验中的作用呢?在分析了各个数据集的 M 矩阵后,我们发现一个 parent 会对应多个 child,且数量从 0~2000 不等;同样一个 child 对应多个 parent 节点,从 0~300 不等,可以看出数据的层次关系较为分散,不是很有规律。有关层次约束,我们提出了以下两个问题,并针对问题设计了实验验证。

问题 1: 数据的层次结构关系对算法有没有作用?

问题 2: 具体是通过什么方式起作用的?

5.2.1 删除一定比例的层次关系

对于第一个问题,很简单,通过随机删除一定比例的层次关系,观察算法在同一数据集下的结果进行对比。以下是实验结果:

表 5.4 YahooMusic 数据集下的结果

	0%	20%	40%	60%	80%	100%
RMSE	1.0699	1.0733	1.0756	1.0781	1.0848	1.0808
HR	0.5088	0.5053	0.4891	0.4790	0.4710	0.4522
ARHR	0.1499	0.1503	0.1448	0.1419	0.1391	0.1332

表 5.5 DoubanBook 数据集下的结果

	0%	20%	40%	60%	80%	100%
RMSE	1.1506	1.8978	2.3761	2.9267	3.3730	3.4079
HR	0.1202	0.1079	0.0953	0.0779	0.0627	0.0430
ARHR	0.0894	0.0823	0.0740	0.0625	0.0526	0.0380

表 5.6 Pathway 数据集下的结果

	0%	20%	40%	60%	80%	100%
RMSE	0.1220	0.1274	0.1375	0.1469	0.1504	0.1569
HR	0.2299	0.2289	0.2279	0.2250	0.2242	0.2237
ARHR	0.1472	0.1464	0.1456	0.1443	0.1435	0.1433

通过实验结果，可以明显观察到随着删除层次关系的比例增加，各项指标趋于更差的结果，说明层次结构对于算法性能的提升，问题一得到解决。

5.2.2 对比相似度

对于问题二，我们可以在 HMNMF 算法的损失函数中观察到：算法是通过使具有层次关系的项目之间具有相似的特征向量，这也正符合现实中的理解——对于具有层次关系的歌曲或是书籍，是属于同一类的，理应具有相似的特征向量。因此我们设计如下实验进行验证：统计有层次结构的商品的特征向量之间的相似度，以及没有层次结构的商品之间的相似度，从而验证在 HMNMF 算法的作用下，属于同一类别的商品是否具有更相似的特征向量。

表 5.7 项目特征向量的相似度比较

		Yahoo Music	Douban Book	Pathway
初始评分矩阵	有层次结构	2.8748	2.8526	2.8750
	没有层次结构	2.8746	2.8729	2.8766
补全后的 评分矩阵	有层次结构	0.7831	4.6089e-04	0.0030
	没有层次结构	1.2188	1.5753	2.5704

从表中可以观察到，通过 HMNMF 算法的作用，具有层次关系的项目的特征向量之间的相似度明显得到了提升，从而说明层次结构约束在算法中起到了明显的作用，对算法性能的提升有重要的作用。

第六章 总结与展望

本文的主要工作分为三个阶段，首先是进行大量的文献阅读，通过阅读论文学习相关知识，打造基础；接下来提出基于多非负矩阵分解的推荐算法，并在数据集上进行了实验，并与其他算法进行对比；在得到实验结果之后，对实验结果进行分析、提出假设，并设计了一系列实验对其进行验证。

通过实验验证，HMNMF 算法相比于其他算法具有以下优势：第一，相比于其他多个经典算法，HMNMF 在各项指标中都表现出了优异的性能；第二，HMNMF 算法有效利用了数据的层次结构信息，利用层次约束极大的提高了算法性能；第三，L21 范式对于噪声有很好的抗干扰性，使得算法更加健壮。

但目前实验中还存在一些不足。例如算法运行过程计算量大，验证实验不够充分，希望日后能补充、完善实验，对算法进行改进。

参考文献

- [1] Wang S, Tang J, Wang Y, et al. Exploring Implicit Hierarchical Structures for Recommender Systems[C]//IJCAI. 2015: 1813-1819.
- [2] Maleszka M, Mianowska B, Nguyen N T. A method for collaborative recommendation using knowledge integration tools and hierarchical structure of user profiles[J]. Knowledge-Based Systems, 2013, 47: 1-13.
- [3] Menon A K, Chitrapura K P, Garg S, et al. Response prediction using collaborative filtering with hierarchies and side-information[C]//Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2011: 141-149.
- [4] Koren Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model[C]//Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2008: 426-434.
- [5] Cremonesi P, Koren Y, Turrin R. Performance of recommender algorithms on top-n recommendation tasks[C]//Proceedings of the fourth ACM conference on Recommender systems. ACM, 2010: 39-46.
- [6] Lee D D, Seung H S. Learning the parts of objects by non-negative matrix factorization[J]. Nature, 1999, 401(6755): 788.
- [7] Singh A P, Gordon G J. Relational learning via collective matrix factorization[C]//Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2008: 650-658.
- [8] Ou W, Yu S, Li G, et al. Multi-view non-negative matrix factorization by patch alignment framework with view consistency[J]. Neurocomputing, 2016, 204: 116-124.
- [9] Mashhoori A, Hashemi S. Incorporating hierarchical information into the matrix factorization model for collaborative filtering[C]//Asian Conference on Intelligent Information and Database Systems. Springer, Berlin, Heidelberg, 2012: 504-513.
- [10] Cai D, He X, Han J, et al. Graph regularized nonnegative matrix factorization for data representation[J]. IEEE transactions on pattern analysis and machine intelligence, 2011, 33(8): 1548-1560.
- [11] Ning X, Karypis G. Slim: Sparse linear methods for top-n recommender systems[C]//2011

- IEEE 11th International Conference on Data Mining. IEEE, 2011: 497-506.
- [12] 张旭, 陈志奎, 高静. 基于图正则化和 $l_1(1/2)$ 稀疏约束的非负矩阵分解算法[J]. 小型微型计算机系统, 2018,39(11):2480-2484.
- [13] 黄卫春, 赵杨, 熊李艳. 用于独立特征学习的稀疏非负矩阵分解算法[J/OL]. 计算机应用研究:1-5[2019-04-29].<https://doi.org/10.19734/j.issn.1001-3695.2018.10.0724>.
- [14] 孙静, 蔡希彪, 姜小燕, 孙福明. 基于图正则化和稀疏约束的增量型非负矩阵分解[J]. 计算机科学, 2017,44(06):298-305.
- [15] 王娟, 熊巍. 基于矩阵分解的最近邻推荐系统及其应用[J]. 统计与决策, 2019,35(06):17-20.
- [16] Lee D D, Seung H S. Algorithms for non-negative matrix factorization[C]//Advances in neural information processing systems. 2001: 556-562.
- [17] Lee D D, Seung H S. Learning the parts of objects by non-negative matrix factorization[J]. Nature, 1999, 401(6755): 788.
- [18] Valcarce D, Parapar J, Barreiro Á. Language models for collaborative filtering neighbourhoods[C]//European Conference on Information Retrieval. Springer, Cham, 2016: 614-625.

致 谢

值此论文完成之际，我首先要向我的导师谢茂强老师表达深深的敬意和感激之情，本论文是在谢茂强老师的悉心指导下完成的。在课题的研究过程中，老师时常与我交流、讨论，对于我的疑惑和问题，老师都能清楚的给出解答和建议；在进行实验和论文撰写过程中，每当我遇到困难，老师都会耐心与我交流，给予了我很大的帮助。

同时，感谢指导我的刘嘉晖师兄和金旭师兄，在进行实验和论文撰写过程中，师兄们时常关心我的进度，对于我遇到的问题也都悉心解答；另外，在实验室完成毕业设计的这段时间，和师兄、同学们相处的很愉快，在研究学习之余，常常一起打球运动，相信这一定是一段难忘的回忆。

时光荏苒，感谢大学四年帮助过我的所有老师和同学们，希望在未来的岁月里，我能够不忘初心，砥砺前行。