## 摘要

在互联网时代，互联网技术以及电子商务的高速发展生成了海量信息，对于信息的消费者也就是用户来说，从海量的信息中找到自己感兴趣的信息变得越来越难；对于信息的生产者来说，如何让自己生产的信息在信息的汪洋大海中脱颖而出也是亟待解决的问题。推荐系统应运而生，它将用户和信息联系了起来，一方面，推荐系统为用户找到感兴趣的项目/服务，且可能为用户找到喜欢的新事物；另一方面，推荐系统为用户提供了个性化的服务，也提高了用户对商家的信任度和粘性，这对商家来说也是十分有益的。然而，传统的协同过滤算法依赖于用户或项目的相似度计算，随着系统中的数据量的极速增加，传统的线性的相似度计算方法表现出了局限性。在此提出一种两阶段联合哈希的协同过滤算法，通过在保留了用户对项目的偏好的情况下，将用户和项目映射到低维空间，使得推荐的过程转化为在低维空间搜索与目标用户具有较小海明距离的项目，省去了用户或项目之间的相似度计算，实现了高效的推荐性能。

本文的主要工作有：

首先，对于系统中的评分数据，我们提出了对其中一个视角（用户或项目）来对该视角进行哈希的方法，从用户或项目视角应用主成分分析技术，得到该视角的低维特征表示，对其应用迭代量化技术进而生成该视角的二值码，从而实现了对评分数据全局特征的提取，也为进一步生成另一视角的二值码奠定了基础。

然后，基于上一过程中得到的其中一个视角的二值码，用已有的评分信息约束用户与项目在海明空间的距离，生成另一视角的二值码，进而实现了对评分数据局部特征的提取，也为接下来利用训练生成的用户和项目的二值码进行高效的推荐做好了准备。

最后，在提出的两阶段联合哈希的基础上，利用二值码的特性和相似性检索的方法，提出了一种快速高效的推荐算法，通过以上两个过程的哈希编码工作，有效的减少了推荐过程中的计算消耗和存储消耗，我们最终提出了一种基于两极端联合哈希的推荐算法。通过在数据集上的仿真实验结果表明，提出的算法能显著提高推荐质量和推荐的效率。

**关键词**：两阶段联合哈希算法；协同过滤；主成分分析；迭代量化方法；海明距离

## Abstract

In the Internet age, the rapid development of Internet technologies and e-commerce has generated massive amounts of information. It is becoming increasingly difficult for consumers of information, that is, users, to find information that they are interested in from vast amounts of information; for information production. For those concerned, how to make the information they produce stand out from the sea of information is an issue that needs to be solved. The recommendation system came into being. It links the user with the information. On the one hand, the recommendation system finds the items/services that are of interest to the user and may find new things that the user likes. On the other hand, the recommendation system provides the user with Personalized services also increase the trust and stickiness of users for businesses, which is also very beneficial to businesses. However, the traditional collaborative filtering algorithm relies on the similarity calculation of users or items. With the rapid increase of the amount of data in the system, the traditional linear similarity calculation method shows limitations. A two-stage joint hashing collaborative filtering algorithm is proposed in this paper. By retaining the user's preference for the item, the user and the item are mapped to the low-dimensional space, so that the recommended process is converted to searching in the low-dimensional space. Target users have smaller Hamming distance items, eliminating the need for similarity calculations between users or items, and achieving efficient recommendation performance.

The main tasks of this article are:

First of all, for ratings data in the system, we proposed a method for hashing this perspective from one perspective (user or project), applying principal component analysis techniques from a user or item perspective to obtain a low-dimensional feature representation of the perspective. It applies iterative quantification techniques to generate binary codes for this perspective, thereby realizing the extraction of the global features of scoring data and laying the foundation for the further generation of binary codes of another perspective.

Then, based on the binary code of one of the perspectives obtained in the previous process, the existing rating information is used to restrict the distance between the user and the item in the Hamming space, and a binary code of another perspective is generated, thereby realizing the localization of the score data. The extraction of features also prepares for efficient recommendation using the binary codes of users and items generated by the training.

Finally, on the basis of the proposed two-stage joint hash, using the characteristics of binary codes and similarity retrieval methods, a fast and efficient recommendation algorithm is proposed. Through the above two processes of hash coding, effective To reduce the computational and storage consumption in the recommendation process, we finally proposed a recommendation algorithm based on the union of two extreme hash. The simulation results on the data set show that the proposed algorithm can significantly improve the recommendation quality and the efficiency of the recommendation.

**Key words**: Two-stage Joint Hashing; collaborative filtering; principal component analysis; iterative quantization; Haiming distance

目录

[摘要 1](#_Toc509262601)

[Abstract 2](#_Toc509262602)

[第一章 引言 6](#_Toc509262603)

[1.1 研究背景及意义 6](#_Toc509262604)

[1.2 本文的主要工作 7](#_Toc509262605)

[1.3 本文的组织结构 7](#_Toc509262606)

[1.4 本章小结 8](#_Toc509262607)

[第二章 相关研究现状 9](#_Toc509262608)

[2.1 个性化推荐系统 9](#_Toc509262609)

[2.2 哈希学习技术 19](#_Toc509262610)

[2.3 本章小结 23](#_Toc509262611)

[第三章 两阶段联合的哈希算法 24](#_Toc509262612)

[3.1问题描述 24](#_Toc509262613)

[3.2 基于用户视角的两阶段联合哈希算法 24](#_Toc509262614)

[3.3 基于项目视角的两阶段联合哈希算法(ITSH) 26](#_Toc509262615)

[3.4 本章小结 27](#_Toc509262616)

[第四章 基于两阶段联合哈希的推荐算法 28](#_Toc509262617)

[4.1 推荐结果生成 28](#_Toc509262618)

[4.2 算法整体结构 29](#_Toc509262619)

[4.3 本章小结 29](#_Toc509262620)

[第五章 实验设计与分析 31](#_Toc509262621)

[5.1 实验数据集 31](#_Toc509262622)

[5.2 评价指标 31](#_Toc509262623)

[5.3 实验结果 32](#_Toc509262624)

[本章小结 34](#_Toc509262625)

[第六章 总结与展望 35](#_Toc509262626)

[4.1 本文总结 35](#_Toc509262627)

[4.2 未来研究方向 35](#_Toc509262628)

[参考文献 37](#_Toc509262629)

[在校研究成果 40](#_Toc509262630)

[致谢 41](#_Toc509262631)

## 第一章 引言

### 1.1 研究背景及意义

当今的信息时代，信息技术的迅速发展导致网上信息量呈爆炸式增长，信息量的爆炸式增长也使得信息的质量良莠不齐，低质量甚至无价值的东西使得用户在面对大量信息时难以获得真正需要的信息，从而降低了对信息的使用效率，导致了信息过载的问题。对于信息的消费者也就是用户来说，从海量的信息中找到自己感兴趣的信息变得越来越难；对于信息的生产者来说，如何让自己生产的信息在信息的汪洋大海中脱颖而出也是亟待解决的问题。个性化推荐系统（Personalized Recommendation System）应运而生，作为解决信息过载问题强有力的方法之一，一方面，推荐系统根据用户的信息需求、兴趣等产生个性化的推荐，为用户提供更高质量的信息服务，与此同时，也为用户节省了大量的时间和精力；另一方面，推荐系统为用户提供了个性化的服务，也提高了用户对商家的信任度和粘性，这对商家来说也是十分有益的。

个性化推荐系统能够利用系统中用户的资料、行为和历史记录等信息以及项目的标签，内容等信息，通过计算相似度的方法对用户的偏好进行预测，来达到将特定的商品或信息推送给目标用户的目的。个性化推荐系统可以主动的为用户提供感兴趣的商品或信息，而不需要用户的主动操作，系统在后台就能够对用户的偏好进行计算预测，这进一步提高了用户或许感兴趣信息的效率。个性化推荐技术中的一种主流方法是协同过滤技术，它主要利用系统中的其他用户的信息来预测当前用户对系统中项目的可能评分，进而选择出可能符合当前用户偏好的项目。传统的协同过滤推荐技术通常将用户对项目的评分矩阵作为用户或项目的表示，然后通过对评分矩阵进行处理，预测用户对项目的评分，进而将预测的结果推荐给用户。

此外，随着大数据时代的来临，由于数据量的体积在日益极速膨胀，系统中的用户和项目的数据呈现高维特征，使得传统的依赖于计算用户或项目的相似度来获取近邻的算法计算和存储开销太高，也大大降低了推荐结果的计算速度，这给传统的协同过滤算法带来了巨大的挑战。研究者们在面对大数据的规模制约着传统算法的有效性问题时，提出了两种思路：一种是对现有算法进行改造，设计可以扩展从而能够应对当前大规模数据环境的算法，算法伸缩的技术有分布式学习和随机优化等方法；另一种是对数据进行降维或消减，使其达到传统算法能处理的规模；降维或消减的方法主要包括随机投影和哈希学习等技术。随机投影通过随机的方式产生投影方向，达到降低数据维度的目地。由于其简单快捷，适用于大数据环境，也在解决分类、最小二乘等问题中得到应用与发展；哈希学习通过机器学习机制将数据映射成二值码的形式[[[1]](#endnote-1)]，从而得到特征的二进制哈希表示，在学习的过程中，要使得哈希码尽可能地保留原数据空间中的近邻关系。哈希学习[[[2]](#endnote-2)]有利于减小存储空间，提高学习速度，降低数据维度，因为其能显著减少数据的存储和计算开销，已经成为机器学习领域和大数据学习领域的研究热点，在信息检索、模式识别、计算机视觉等领域得到广泛的研究与应用。

因此，本文提出了基于两阶段联合哈希的推荐算法，将传统的协同过滤算法与哈希学习相结合，实现了高质高效的个性化推荐。在大数据时代背景下，本文提出的方法在应对大规模数据协同过滤的扩展性问题和推荐效率问题时更有价值。

### 1.2 本文的主要工作

本文针对大规模数据背景下的个性化推荐系统，提出了一种两阶段联合哈希的协同过滤推荐算法，利用了高效的基于哈希编码的近似最近邻搜索，实现了高效、高质量的个性化推荐。

本文的主要工作有：

首先，对于系统中的评分数据，我们提出了对其中一个视角（用户或项目）来对该视角进行哈希的方法，从用户或项目视角应用主成分分析技术，得到该视角的低维特征表示，对其应用迭代量化技术进而生成该视角的二值码，从而实现了对评分数据全局特征的提取，也为进一步生成另一视角的二值码奠定了基础。

然后，基于上一过程中得到的其中一个视角的二值码，用已有的评分信息约束用户与项目在海明空间的距离，生成另一视角的二值码，进而实现了对评分数据局部特征的提取，也为接下来利用训练生成的用户和项目的二值码进行高效的推荐做好了准备。

最后，在提出的两阶段联合哈希的基础上，利用二值码的特性和相似性检索的方法，提出了一种快速高效的推荐算法，通过以上两个过程的哈希编码工作，有效的减少了推荐过程中的计算消耗和存储消耗，我们最终提出了一种基于两极端联合哈希的推荐算法。通过在数据集上的仿真实验结果表明，提出的算法能显著提高推荐质量和推荐的效率。

### 1.3 本文的组织结构

本文的余下的部分按照下面的结构来安排。

本文的第二章对目前的推荐系统和哈希技术相关领域的的国内外研究现状和成果做了介绍和总结。首先，介绍了个性化推荐系统，包括基于内容的推荐，基于协同过滤的推荐和基于混合技术的推荐，其中重点介绍了协同过滤相关技术，基于协同过滤的推荐又主要包括基于内存的推荐和基于模型的推荐，我们对基于内存的推荐的两种主要的方法做了比较详尽的介绍，也对这两种方法的优缺点做了分析。接下来，我们又介绍了基于模型的推荐中的代表方法，给出了几种基于模型的推荐方法的主要形式。最后，我们介绍了哈希学习的发展和现状，并且给出了局部敏感哈希和迭代量化哈希这两种经典的哈希方法的介绍。

本文的第三章主要研究了两阶段联合的哈希算法，首先在评分上对其中一个视角应用主成分分析和迭代量化技术，生成该视角的二值码，也实现了对评分数据的全局特征的提取，接下来，利用评分和已经得到的二值码，建立损失方程，求解最优化问题，来迭代生成另一视角的二值码，进而实现对评分数据局部特征的捕捉。通过以上的过程，我们就生成了用户和项目的二值码，提出了两阶段联合的哈希算法。

本文的第四章主要讨论了基于两阶段联合哈希的推荐算法。首先，我们分析了用二值码来进行相似度检索的优势，然后，我们进一步对生成的二值码做0,1化处理，使得它能节省更多的存储空间和加快相似度检索的速度，最后，我们将生成二值码的工作和用二值码来检索、推荐的工作进行了整合，提出了基于两阶段联合哈希的推荐算法。

本文的第五章介绍了本文所提出的基于两阶段联合哈希的推荐算法在MovieLens数据集上与对比算法的实验结果及分析。首先，我们介绍了实验选择的对比方法，然后介绍了实验使用的数据集，以及对数据集所做的处理，接下来，我们介绍了选取的评价指标，最后，我们介绍和分析了实验的结果。

本文的第六章简要概括了本文的主要工作，对全文做了总结然后对未来的研究做了展望。

### 1.4 本章小结

本章我们首先介绍了相关的研究现状，阐述了本文所研究的基于两阶段联合哈希的推荐算法对于提高传统协同过滤算法推荐效率和推荐质量上的作用和意义，然后对本文所做的主要工作做了简单的介绍，最后对本文的组织结构做了介绍。

## 第二章 相关研究现状

### 2.1 个性化推荐系统

当今的信息时代，信息技术的迅速发展导致网上信息量呈爆炸式增长，信息量的爆炸式增长也使得信息的质量良莠不齐，低质量甚至无价值的东西使得用户在面对大量信息时难以获得真正需要的信息，从而降低了对信息的使用效率，导致了信息过载的问题。个性化推荐系统（Personalized Recommendation System）应运而生，一般而言，个性化推荐系统就是根据用户的历史行为、社交关系、兴趣点等信息来判断用户感兴趣的项目的工具和技术。这些项目的形式可能包括音乐、电影和书籍等，具体来说，推荐系统需要预测目标用户对项目集中的项目的评分或偏好，为进一步的系统的推荐提供决策，实现对用户的个性化需求的技术支撑。作为解决信息过载问题强有力的方法之一，一方面，推荐系统根据用户的信息需求、兴趣等产生个性化的推荐，为用户提供更高质量的信息服务，与此同时，也为用户节省了大量的时间和精力；另一方面，推荐系统系统也可以提高与用户的信任度和粘性，增加创收。推荐系统的价值可以通过一组数据来说明，Netflix上有2/3被观看的电影来自推荐，Google新闻中有38%的点击量来自推荐，而知名电商网站Amazon中有35%的销量来自推荐。

目前，个性化推荐系统已经在互联网领域的各个方面得到了广泛的应用，较为突出的有电子商务、社交网络和在线影音等。以国内著名电商网站淘宝网[[[3]](#endnote-3)]为例，图2-1和图2-2为两个用户所看到得淘宝网首页，是非常明确的 ‘个性化推荐’的体现。又如当当网[[[4]](#endnote-4)]，可以看出网站在各个层面上实现了个性化推荐，如根据当前商品的推荐（图2-3）和根据历史浏览记录的推荐（图2-4）。个性化推荐技术一般可分为三种类型，分别是：基于内容的推荐技术、基于协同过滤的推荐技术和基于混合技术的推荐技术。

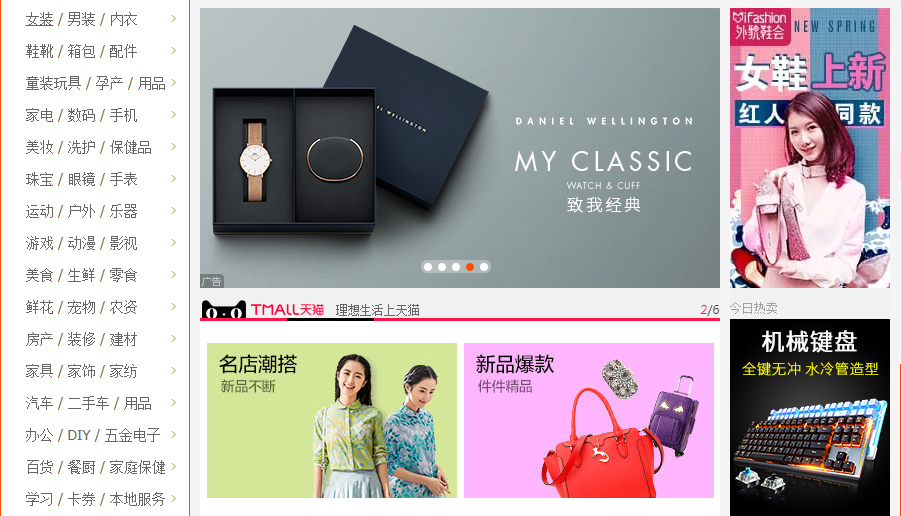


图2-1



图2-2



图2-3



图2-4

#### 2.1.1基于内容的推荐技术

基于内容的推荐技术是根据项目的内容或者通过项目的内容提取出的项目的特征，来为目标用户推荐跟其过去偏好的项目相似的结果[[[5]](#endnote-5)]。基于内容的推荐算法一般会根据用户的历史信息建立该用户的用户概貌（user profile），得到用户的偏好的描述，然后根据项目集中项目的特征来计算候选项目与目标用户偏好之间的相似度，进而对得到的相似度计算结果进行排序，为用户推荐相似度最高的项目。所以，这种方法的核心在于为用户和项目提取合适的特征和基于提取的特征上的相似度计算。

基于内容的推荐技术有着简单，有效的优点，在广告过滤，垃圾邮件过滤，色情网页过滤等场景下有着不错的发挥；基于内容的推荐技术也有着极其明显的缺点，如难以区分项目内容的品质以及风格，而且无法为用户提供新的感兴趣的推荐服务，它只能发现和用户已有的兴趣相似的项目。

#### 2.1.2基于协同过滤的推荐技术

协同过滤(Collaborative Filtering)最早由Goldberg等提出并应用于一个电子文档过滤系统Tapestry[[[6]](#endnote-6)],之后的几十年来，不断的有专家和学者加入到研究这个课题的队伍中来。其中，美国明尼苏达州立大学的GroupLens所设计实现的Grouplens系统[[[7]](#endnote-7)]，基于用户对新闻的偏好在过去与将来都是一致的，从而利用用户之间的协作，从大量的Usenet新闻中发现用户们感兴趣的新闻，给用户提供个性化的推荐服务。同样来自明尼苏达州立大学的Movielens系统[3]也对协同过滤系统的兴起起到了推动作用，Movielens系统所使用的数据集被公开后，为后来的众多研究者提供了便利，也使得协同过滤算法获得更为广泛的关注与研究。

2006年，美国Netflix公司举行的一场竞赛，将协同过滤算法的热度推到了高峰，Netflix公开了该网站用户对电影评分的部分数据供参赛者实验使用，最后该项竞赛的一百万美元奖金由BPC团队获得，该团队在矩阵分解方法的基础上引入了隐性反馈[[[8]](#endnote-8)]，结合了近邻模型作为局部修正，又引入了近邻模型中的相似度矩阵分解思想，由此，矩阵分解方法在实践中得到了很大程度的完善，也推动了协同过滤技术的更深发展。

协同过滤的基本思想是根据具有类似兴趣的用户的偏好来对用户进行预测进而完成推荐，利用一个用户对其它项目的评分以及整个用户集过去的评分来预测这个用户对未评分项目的评分。协同过滤的原理是：根据用户群体的历史行为，找到与目标用户兴趣偏好相似的若干其他用户作为目标用户的邻居, 然后根据邻居用户的偏好信息计算出目标用户的兴趣列表。其基本思想就是日常生活中, 人们往往会选择兴趣爱好相似的朋友的推荐。协同过滤就是把这一思想运用到推荐系统中来, 利用其他用户对某一项目的评价和用户之间的相似度来向目标用户进行推荐。

协同过滤是推荐系统应用最为广泛的技术之一，它利用信息之间的关联程度把不相关的信息过滤掉，只保留最有价值的部分信息。基于协同过滤的推荐算法基于以下事实：用户对跟它兴趣相似的其他用户感兴趣的东西往往也会感兴趣。与基于内容的推荐不同的是，协同过滤推荐不用关心项目的具体内容，也不需要给项目属性作标签，所以能够应对复杂的难以表达的场景，而且协同过滤推荐的结果具有新颖性的特点，即常常能给用户推荐新奇的项目，进而提高了用户的体验。

基于协同过滤的个性化推荐技术一般可以分为基于内存的协同过滤(Memory-based CF)、基于模型的协同过滤(Model-based CF)。

##### 2.1.2.1 基于内存的协同过滤推荐

基于内存推荐是协同过滤推荐方法中广为流行的一类方法，也是最早被研究和应用的方法，其原理是通过直接计算用户或项目的相似度然后利用 K 近邻(KNN)算法来寻找邻居并以此产生推荐，基于内存推荐按照计算相似度的对象的不同可以分为基于用户和基于项目两种[[[9]](#endnote-9)]。

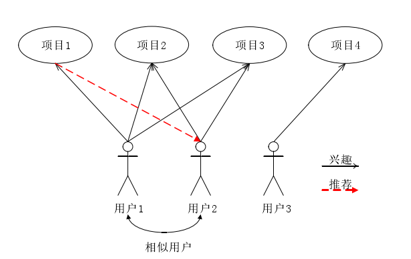


图2-5 基于用户的协同过滤推荐原理图

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 项目1 | 项目2 | 项目3 | 项目4 |
| 用户1 | 1 | 1 | 1 | 0 |
| 用户2 | 0 | 1 | 1 | 0 |
| 用户3 | 0 | 0 | 0 | 1 |

表2.1

基于用户的推荐算法的基本思想是根据用户的历史行为计算用户之间的相似度，然后取与目标用户相似度最高的 K 个近邻用户，计算近邻用户对该项目的评分的加权平均，即为该用户对项目的预测评分。由图 2-5和其对应的表2.1，可以得到，用户 1 和用户 2 都对项目2和项目3感兴趣，因此可以认为用户1与用户2之间的相似度较高，对于用户2来说，用户1即为用户2的近邻，由用户 1对项目1感兴趣可以推断用户2可能对项目1也感兴趣，基于上述推断可以对用户2推荐项目 1。

在实际的应用场景下，由于需要在线上有较快的响应速度，一般将推荐的过程分为线上和线下两部分，将比较耗时的各用户之间的相似度计算由线下来完成，线下通过用户之间的相似度计算来得到各用户的近邻，并将邻域信息存储下来；而线上部分就只需要查询邻域表来预测目标用户感兴趣的项目即可，这可以大大的提高响应的速度。

线下部分的执行过程如下：

对每一个用户i

对每一个其他的用户j

计算用户i和用户j之间的相似度sim(i,j)，并保存为用户相似矩阵

以给用户u推荐项目为例，基于用户的k-近邻算法的线上部分的主要流程如下：

(1) 访问用户相似矩阵，获取k个与用户u相似的用户集合N（u），作为用户u的近邻

(2) 对于N(u)中的每一个用户ui，获取与用户ui有关联的项目集合，对项目集合中的每个项目，计算用户u的偏好值

(3) 对项目集M（u）中的项目进行加权，去重和排序，并取top-N个项目推荐给用户

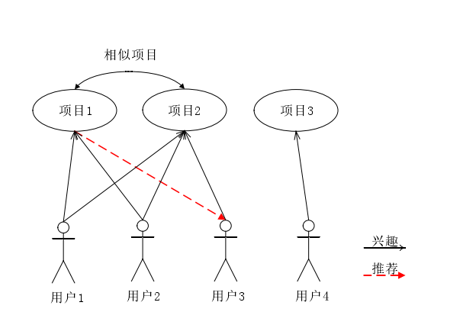


图2-6 基于用户的协同过滤推荐原理图

基于项目的推荐算法[[[10]](#endnote-10)]与基于用户的推荐算法原理类似，基于项目的推荐算法给用户推荐和该用户之前偏好的项目相似的项目，如图 2-2 所示，用户1和用户2都对项目1和项目2感兴趣，可以认为项目 1 和项目 2 的是相似项目，因为项目1和项目2之间的相似度较高，现在，用户 3 对项目 2 感兴趣，那么可以推断用户3也对项目2感兴趣，因此可以向用户3推荐项目2。

在实际的系统场景下，基于用户的推荐算法往往不如基于项目的推荐算法效果[[[11]](#endnote-11)]。不难看出，随着用户数量的增长，基于用户的推荐算法的计算时间也随着增长，这就增加了基于用户的推荐算法的运行时间，同理，随着项目数量的增长，基于项目的推荐算法的计算开销也随之增加，基于项目的推荐算法的运行时间也随之增长。一方面，在大多数的系统中，用户的规模都是要远远大于项目的规模，也就是说系统中的用户数量往往比项目数量多很多，因此基于项目的推荐算法的性能往往带来比较显著的提升。另一方面，项目之间的相似性相比较用户之间的相似性来说更加稳定，用户的认知很可能会随着时间的推移而发生变化，进而导致之前的用户相似度的计算不准确。相反，随着时间的推移，对每件项目来说，其接触的用户越来越多，项目间的相似性会渐渐趋于收敛，最终不再发生剧烈的抖动。因为无论是基于项目的推荐算法，还是基于用户的推荐算法都高度依赖相似性计算的准确度，所以基于项目的推荐算法的性能往往要优于基于用户的推荐算法。

毋庸置疑，相似性度量是基于用户的推荐和基于项目的推荐中最重要的一项工作之一，相似性度量可以用来搜索相似的用户或项目，它决定着能否发现的近邻用户或项目是否准确以及有多准确，也影响着进一步的推荐结果的准确性。相似性度量一般立足于衡量对象的描述特征或属性，通常由各对象属性集合或特征向量间的相关系数、相似系数以及距离等来计算得到。常用的相似性度量的方法有四种，如下：

1. 欧氏距离

用户之间的欧氏距离是一种 Lp 范式，当 p = 1 时称为曼哈顿距离。可见， 欧氏距离越大，相似度越小。p = 2 时，欧氏距离 dis( A，B) 为

 (2-1)

欧氏距离衡量的是空间各数据点的绝对距离，跟各个数据点所在的位置坐标直接相关，欧氏距离能够体现个体数值特征的绝对差异，更多的用来分析维度的数值大小的差异。

1. 余弦相似度

余弦相似度是用向量空间中两个向量夹角的余弦值作为衡量两个个体间差异的大小的度量。把用户看作 n 维的向量，就可以根据向量化的用户来进行相似度的衡量。如果两个用户向量的方向一致，即夹角接近为零，那么这两个用户向量就相近。与欧式距离类似，这个夹角同样可以用来度量相似度。余弦值越大，表示相似度越高，最大为 1，最小为-1。1表示这两个用户完全相似，-1表示这两个用户完全不同。与欧式距离相比较来说，余弦距离更多的是从方向上区分差异，而对绝对的数值不敏感，同时它也修正了可能存在的度量标准不统一的问题。余弦的计算公式为：

 (2-2)

1. 皮尔逊相关系数

皮尔逊相关系数从本质上来看可以说是余弦相似度在维度值缺失情况下的一种改进，它可以用来度量两个变量之间的相关程度，其取值范围为[-1,1]，1表示这两个变量完全正相关，-1则表示这两个变量完全负相关，0表示不相关。其计算方法如下：

 （2-3）

由上式可以看出，皮尔逊相关系数实际上是两个向量的协方差比上其各自的方差的乘积，其中，协方差反映了这两个向量变化趋势的一致性，比上方差的乘积则是用来对协方差进行归一化。

1. Jaccard 相似系数

Jaccard相似系数是衡量两个集合相似度的一种指标，它可以用来比较二元向量的相似性和分散性的概率。用户间的相似性可以通过这两个用户所对应的向量集合的交集与并集之比来度量， 那么这两个用户间的相似性 sim( A，B)可以表示为:

 （2-4）

基于Jaccard系数的相似性度量不关注用户对项目的偏好值的高低，即评分的大小，而是关心系统中用户是否对项目集中的项目做出过评分，也即表达过偏好。它是由两个用户都打了分的项目的数量除以这两个用户至少有其中一个用户打过分的项目的数量得到，从根本上来说，Jaccard系数是两个被打出评分的项目集合的交集与并集的比值，如图 2-7 所示。

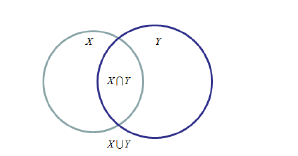


图2-7

基于内存的协同过滤技术其原理通俗易懂，也易于实现，最重要的是它不关注项目的内容，特征等信息，因此不需要处理项目的内容和特征等信息，这使得基于内存的系统过滤技术通用性的很强，适用于较多的场景。然而，这类方法也存在着一定的缺陷：首先，因为在现实的场景下，得到的用户数据往往是高度稀疏的，这会导致基于内存的协同过滤技术计算用户或项目的相似度时得到的结果不够准确，进而导致了推荐的性能显著下降；另外，在处理历史信息较少的用户或项目，如系统中新增加的用户或项目，协同过滤技术无法对其展开与系统中的其他用户或项目的相似度计算，进而对其进行处理，也就无法将新的项目推荐给用户或者为系统中新增加的用户推荐项目，一定程度上可以说，推荐算法对这类用户和项目是失效的；最后，随着互联网技术的高速发展，系统中的数据规模越来越大，在面对大规模的数据时，基于记忆的协同过滤方法所需要的计算成本非常高，甚至可能无法计算，这意味着算法无法很好的适应大规模数据的系统，即算法的扩展性不强。

##### 2.1.2.2 基于模型的协同过滤推荐

基于模型的协同过滤推荐是协同过滤推荐的重要推荐算法之一，基于模型推荐事先利用训练数据集离线学习一个预测模型，再将模型应用到在线系统进行推荐。与基于内存的协同过滤推荐算法不同的地方在于，基于模型的协同过滤推荐不是利用启发式规则预测用户对项目的评分，而是采用机器学习和统计的方法对已有数据建立模型，利用建立好的模型来进行评分预测。基于模型的协同过滤推荐性能的好坏与建立的模型有比较大的关系，好的模型能挖掘出更多的用户和项目之间的潜在关系进而获得更准确的推荐结果，但是在现实场景下由于建立模型相对比较耗时，一般都只是在离线完成模型的建立。经典的基于模型的协同过滤推荐方法主要有奇异值分解法[[[12]](#endnote-12)]、贝叶斯分类法[[[13]](#endnote-13)]、聚类法[[[14]](#endnote-14)]、关联规则法[[[15]](#endnote-15)]等。

**奇异值分解模型**用于协同过滤的主要思想是用户对项目的偏好是基于用户对这些项目的隐含特性的喜好的,而且在项目之间也存在一些共有的特征。用户对某一项目评分高是因为他们对这些潜在的，共有的特征的评分较高,所以通过把系统中用户对项目的评分用线性代数方法分解为一些潜在特征,然后根据用户对这些潜在特征的感兴趣程度来预测用户对他所未评分项目的偏好。

具体来说，用户-项目的评分矩阵R可以用公式(2-5)来表示：

R = PQ’ (2-5)

其中P(N\*K)，Q(M\*K)分别为表征所有用户和所有项目的因子矩阵，K代表了每个用户和每个项目的特征可以用前K个奇异值所对应的K个奇异向量表征，其取值代表了保留了K个维度的特征信息，K的取值很重要，如果太小,就无法得到原始评分矩阵中重要特征,如果太大,就会失去降维的意义,因此需要通过多次试验来确定要保留的维数，一般满足K<<Min(N,M)。式(2-6)实际上是矩阵理论中矩阵奇异值分解公式(2-7)的变形。

R=PΣQ (2-6)

奇异值分解模型中的奇异矩阵P、Q中各个元素作为待估计参数，以用户-项目评分矩阵中的现有数据来作为训练数据对训练模型，也就是求解优化问题(2-7)，其中μ是权重系数，来缓解模型的过拟合。

 (2-7)

优化问题(2-7)求解后得到P、Q奇异矩阵，然后带入公式(2-6)即可得到预测的用户对项目的评分值。奇异值分解模型将评分预测问题转化为了奇异矩阵的参数估计问题，由于奇异矩阵P、Q中的待估计参数远小于原始用户-项目评分矩阵的元素个数，因此该方法能起到降低矩阵维数的作用，[13]中的实验结果表明奇异值分解方法在某些场合下能改善推荐系统的可扩展性问题,在稀疏的评分矩阵上效果比较好,但是如果原始的评分矩阵稀疏程度较大,其推荐效果甚至不如基于用户的协同过滤推荐。

**贝叶斯分类模型**方法的基本思想是把用户对项目的评分分成若干个类，如评分范围属于[1,5]的可以分成5类，每个类代表评分值1、2、3、4、5，然后可以利用目标用户与项目的属性特征来计算该目标用户对该项目的评分属于各个类别的概率，最后取概率最大的评分类中评分值。贝叶斯分类方法的具体数学描述如下：若用户-项目评分矩阵中的每个数据样本有n个属性(用户和项目)，则每个数据样本(评分)可以用一个特征向量表示为，又如果所有的数据样本的分类数为m，分别用表示，对于待分类的数据样本X，根据贝叶斯定理(2-8)可以计算其属于某个类的条件概率：

 （2-8）

其中，P（Ci）是样本属于Ci类的先验概率，P（X|Ci）是样本在条件为Ci类情况下属性为X的条件概率，由于P（X）对于所有的类别都是常数，最大化后验概率P（Ci|X）可以转化为最大化P（Ci）P（X|Ci）。为了更方便的计算P（X|Ci）的值，贝叶斯分类器通常都假设各个类之间是相互独立的，也就是说各个属性的取值与其他属性无关。因此有：

 （2-9）

先验概率P（x1|Ci）, P（x2|Ci）,…, P（xn|Ci）能够从训练数据集中统计求出。基于此，对于一个类别未知的样本，可以先分别计算该样本属于每一个类别的概率，然后选择其中概率最大的类别作为其类别，即：

 (2-10)

**聚类模型**主要有三类：用户聚类、项目聚类和用户-项目联合聚类。用户聚类一般是利用聚类算法将用户集聚类成若干类，使得每一类中是具有较高相似性的用户，在推荐过程中，先在目标用户所在的聚类中心上寻找目标用户的若干个近邻用户，然后根据这些近邻用户对某项目的评分加权即可得到目标用户对该项目的预测评分。同理，项目聚类是对项目集进行聚类，通过在项目子集中寻找某项目的近邻来实现目标用户对该项目的评分预测。用户-项目联合聚类又叫用户-项目协同聚类，考虑到用户和项目的相关性，对用户集和项目集都进行了聚类，将用户聚类和项目聚类两个步骤依次循环迭代直至最终收敛。应用于聚类模型的聚类算法中最常见的是 K-means 算法[[[16]](#endnote-16)]，K-means是机器学习领域中的一种非监督学习算法。以项目聚类为例，K-means 聚类算法的基本步骤如下：

Step1.设定聚类中心数 K，并从用户-项目评分矩阵中寻找到评分数最多的 K

个项目作为初始聚类中心点，记为

Step2.对于项目集中每个项目 v，计算 v 与各个聚类中心的距离或相似度，得到聚类Ci，将该项目添加到聚类Ci中。

Step 3. 修 改 聚 类 中 心 ， 计 算 每 个 聚 类 的 均 值 ， 作 为 新 的 聚 类 中 心，如果聚类中心未发生改变或迭代步数达到最大的设定值，则算法终止，得到 K 个聚类，否则跳转到 Step 2 继续迭代。

基于聚类模型的推荐在聚类中心上搜索目标用户/项目的近邻，相比在全局上做近邻搜索，最近邻的寻找范围大大的缩小了，这大大减少了推荐过程的计算，提升了推荐的速度，在一定程度上有效的缓解了协同过滤推荐系统面临的扩展性问题，然而，在聚类中搜索近邻这种做法依赖于聚类模型的优劣，相比在全局上搜索近邻，聚类模型在改善了扩展性的同时也牺牲了一定的推荐准确性。

**关联规则**挖掘因为其能够发现不同商品在销售过程中的相关性，在零售业有着广泛的应用。作为数据挖掘领域的一个重要分支，关联规则挖掘旨在发现数据集中项集之间的关联。若是项的集合，D是数据库中事务的集合，每个事务T是不同项的集合，且有。规则是形如的蕴含式，其中A,B是一个项集，即,并且。规则在事务集D中成立，具有置信度和支持度，同时满足最小置信度阈值和最小支持度阈值的规则即为强关联规则。

关联规则的挖掘过程可以分为两步：第一步在样本集上生成所有的频繁项集，这些项集的出现频率至少满足大于等于定义的最小支持度。第二步由第一步得到的频繁项集生成强关联规则，规则必须同时满足最小支持度和最小置信度。在以上的两个过程中，第一步要在整个数据库规模上求解，故如果数据库规模很大，则解空间也非常大，导致计算量很大，会影响算法的效率。

关联规则法用于协同过滤系统是[13]首先提出，通过利用Apriori算法[[[17]](#endnote-17)]挖掘用户的历史历史记录的关联性来进行推荐。Apriori算法首先从聚集树发现匹配用户当前访问操作路径的关联规则,然后再根据推荐度因子的大小确定推荐项,推荐度因子定义为关联规则的置信度乘以距离因子，关联规则模型的生成可以通过离线的方式进行,因此可以保证系统进行有效的，实时的推荐。

#### 2.1.3基于混合技术的推荐技术

基于内容的推荐与基于协同过滤的推荐都有着各自的优点和缺点，因此，将这两类方法结合打开了新的思路。基于混合技术的推荐方法可以分别使用基于内容的推荐方法和基于协同过滤的推荐方法，然后将两种方法生成的结果进行结合；也可以将一些内容特征作为协同过滤中所使用的信息的一部分，或者反过来把用户和项目的历史数据作为内容特征用于基于内容的推荐技术等等。研究表明，与纯粹的基于内容推荐或协同过滤推荐相比，混合推荐技术可以取得更有效的推荐性能，同时，也有助于应对和处理协同过滤推荐中的冷启动和稀疏性等问题。

### 2.2 哈希学习技术

协同过滤系统为目标用户在项目集上检索偏好的项目，这个过程可以看做是一个相似性检索问题：检索用户感兴趣的项目[[[18]](#endnote-18)]。传统的相似性度量如Jaccard相似系数、余弦相似系数和欧式距离等尽管已经广泛使用在推荐系统中，但是其在高维海量的数据上计算量过大且对于高稀疏度的数据其相似度计算精度低。研究表明，哈希学习在相似性检索方面有着独特的优势，它通过机器学习机制将数据映射成二进制串的形式，能显著减少数据的存储和通信开销，从而有效提高学习系统的效率。由于哈希学习学到的二值码能保持原空间中的近邻关系，其在应用快速相似性检索的领域具有强大的生命力，已经被广泛应用到信息检索[[[19]](#endnote-19)]、计算机视觉[[[20]](#endnote-20)]和推荐系统等领域。

对于高维数据点x，我们可以通过一系列的哈希函数计算得到该数据点对应的k位二进制编码，这样我们就将原始的高维数据映射到了低（k）维空间。哈希学习一般包含三个主要部分：哈希函数的学习与设计、哈希编码的生成和利用哈希编码来进行相似性检索[[[21]](#endnote-21)]。其中，最主要的也就是最为核心的部分是哈希函数的学习与设计，在大多数的研究中，哈希编码的生成也包含在哈希函数的学习过程中。

哈希可以看作是某种概率降维，将数据从原始空间映射到特定长度的海明空间，并确保相似数据点在编码之后距离也非常近。常见的哈希函数学习方法分为数据独立的方法和数据依赖的方法。其中，数据独立的哈希学习方法中最经典的为局部敏感哈希[[[22]](#endnote-22)](Locality Sensitive Hashing, LSH)，这一类的哈希学习方法不需要对给定的数据集进行挖掘分析，而是直接在数据上随机投影使得投影后的哈希编码能在一定程度上保留原始空间上的相似度。这类哈希学习方法计算的速度较快，效率也就很高。然而，由于其不对所给的数据进行挖掘，其性能一般不如数据依赖的方法。

数据依赖的哈希学习方法通过使用机器学习方法，在给定的数据集上来训练以得到对数据编码的哈希函数。根据利用数据中信息的形式和程度的不同，可以分为非监督式[[[23]](#endnote-23)][[[24]](#endnote-24)][[[25]](#endnote-25)][[[26]](#endnote-26)]、半监督式[[[27]](#endnote-27)][[[28]](#endnote-28)]和监督式[[[29]](#endnote-29)][[[30]](#endnote-30)][[[31]](#endnote-31)][[[32]](#endnote-32)]的哈希学习，顾名思义，非监督哈希学习一般倾向于获取数据的分布和流行结构等特性；半监督式哈希学习同时利用已标注的和没有标注的数据来训练哈希函数；监督式哈希学习则利用度量学习、核学习和深度学习等方法来训练哈希函数。其中，谱哈希[[[33]](#endnote-33)]是非监督哈希学习中比较有代表性的方法之一，谱哈希通过把数据点之间的相似度构建为一张图，进而把哈希编码问题转化成了图的平衡划分问题。然后利用拉普拉斯特征映射的思想，通过计算该图的拉普拉斯矩阵最小的若干个非零特征值对应的特征向量来解决上述问题。

以上介绍的哈希学习技术主要应用在文本数据和图像数据之上，哈希学习首先由zhou 等在2012年推介到推荐系统领域，在推荐系统中，从数据集中直接训练出用户-项目的二值码是一个NP-难问题[16]。为了从训练集上得到所需的二值码，Zhou等采用了两步学习策略，即先对原空间的样本采用度量学习进行降维，得到低维空间的实数向量的表示，再对得到的实数向量进行量化得到二进制哈希码，在[16]中，作者设计了采用数据集的中位数作为阈值和正交变换的两种量化方法，设计并实现了CFCodeReg和CFCodepair两个分别对应于平方损失和成对损失的度量学习模型，并在MovieLens、EachMovie和Netflix数据集上取得了比谱哈希、矩阵分解模型更好的推荐结果。Zhou等作为将哈希学习引入到协同过滤系统中的先驱，填补了协同过滤系统中哈希学习技术的空白，所采用的两步学习策略给后来的研究者提供了依据与指示。

zhang等[[[34]](#endnote-34)]认为在度量学习阶段，内积并不等于相似度，用内积来表示相似度的做法会导致最后的倾向度预测的精度的损失，随后他们提出了正则化后的用户、项目特征来计算余弦相似性，并分别的基于量级和相位来量化得到二进制的哈希码，取得了更高质量的推荐效果。在[[[35]](#endnote-35)]中，liu等认为在用户与项目这两个维度内部实体之间的联系在生成简约二值码的过程中起着更重要的作用，而不仅仅是用户与项目之间的相互联系，在他们的度量学习阶段，所构造的目标方程包含了每个维度内实体哈希后的损失和整体的维度之间的实体哈希后的损失两个部分，在采用了奇异值分解的技术对目标问题进行优化得到简约的二值码过程中，也得到了对于之后加入的用户或项目进行哈希的哈希函数。以上基于二阶段的哈希方法均取得了精确度的提升，然而，在[[[36]](#endnote-36)]中，zhang等认为上述的采用二阶段过程的哈希学习方法损失了大量的信息，也导致降低了最后系统的性能。作者设计了实值矩阵分解方法与量化为二值码的矩阵分解方法的对比实验，得到二阶段过程会降低算法的性能的结论。最后作者采用了直接解决离散最优化的问题，通过将两个阶段联合起来，并引入连续值的辅助矩阵，建立辅助矩阵与原矩阵的联系，将约束条件转移到连续值的辅助矩阵上，从而将优化问题分解为包含了辅助矩阵的四个分问题，然后设计了并行离散优化算法和基于小规模的奇异值分解算法解决上述的四个分问题，得到平衡的正则化后的二值码，取得了比其他基于二阶段哈希学习方法更好的推荐效率。

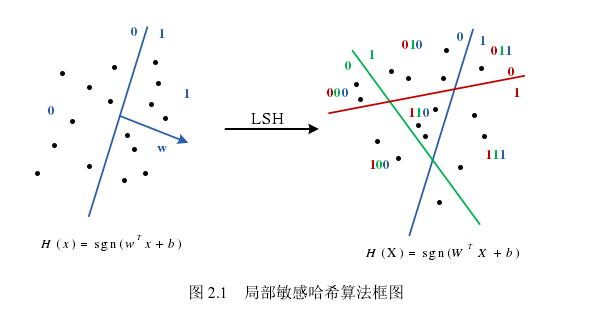
接下来我们通过介绍具体的算法的实现过程，来详细阐述数据独立的哈希学习方法和数据依赖的哈希学习方法的实现细节和发展现状。

#### 2.2.1 局部敏感哈希

局部敏感哈希是最经典的近似最近邻检索算法，由Indyk等人在1998年提出，为近似最近邻检索问题而制定的主存算法。该算法的思想是把相似的点尽可能高概率的映射到一样的哈希桶内，将原始空间中的高维数据的距离计算问题转换为海明空间中的基于哈希码的距离计算问题，这样可以极大的减少计算量，因为海明空间中的编码为0和1，结构非常简单，易于计算，通过这样的方式可以实现比较快速的近似最近邻检索过程。局部敏感哈希的原理是认为在原始空间中数据点之间的碰撞概率与他们之间的距离成反比，如果两个数据点之间的欧氏距离越小，那么他们碰撞的概率就越大，反之亦然。接下来介绍（）-敏感的哈希算法，其中，B(q, r)是以q为中心，r为半径的构成的闭合球形区域，h(.)是哈希函数，和是检索半径，和是不同的碰撞概率，根据局部敏感哈希的算理，则有：

1.如果，则有

2.如果， 则有



局部敏感哈希的编码过程如图所示，可以看出，LSH是通过生成特定数目（等于编码长度）的超平面来对原始数据样本进行划分的，然后给位于某一超平面同侧的样本赋予同样的编码值，不同侧的样本赋予不同的编码值，由于它属于数据独立的哈希方法，其超平面的生成是随机产生的，在实际应用中，该超平面是从一系列满足标准正态分布的向量集合中随机抽取K个得来的，其哈希函数可以描述为



其中W是构成超平面的投影矩阵，X是原始数据样本集，b是偏移量。虽然LSH在近似最近邻检索问题上有很大的计算优势，尤其是当编码位数较高时，其检索的准确度也较高，但是由于它基于随机投影的性质，使得它不能很好的获取原始数据的分布特性，也就限制了该算法的应用场景。在局部敏感哈希提出后，研究者们不断提出了LSH算法的各种变种和改进版本，如贝叶斯LSH（BayesianLSH）,自适应LSH(Query Adaptative LSH)等众多其他LSH算法，这些算法都或在一定程度上提升了LSH算法的检索性能，或是拓展了LSH算法的应用场景。但是他们都属于数据独立的哈希方法，缺少对数据集分布特性的描述，因此，仍然限制了他们的应用前景。

#### 2.2.2 迭代量化哈希

哈希学习发展至今，迭代量化哈希算法[[[[37]](#endnote-37)]]（Iterative Quantization，ITQ）是最有代表性的哈希算法之一，由于它最早考虑到不同维度信息量分布不均问题，并且提出了对主成分分析后的低维特征迭代量化的策略，在检索性能和速度上都取得了较大的提高。它的主要思想为：

1. 首先在原始空间上应用主成分分析技术[[[[38]](#endnote-38)]](Principal Component Analysis, PCA)对高维样本进行降维，从而将其高维特征映射到低维空间，得到低维空间的特征表示，记X为原始空间数据，降维后的维度为c，W为投影矩阵，V为应用PCA后得到的原始高维数据的低维表示，其中V = XW。
2. 然后对V量化进行编码，考虑到数据方差分布不均的问题即投影后的低维数据可区分性差，直接对V进行量化无法很好的代表样本的真实分布情况，因此ITQ算法提出了数据旋转的策略，因为若W为最优的投影矩阵，则WR也是如此，其中R为的正交矩阵，基于此对PCA降维后的样本进行旋转，使得旋转后的数据在各个主方向上方差尽可能的均衡，建立如下损失方程：

(2-11)

其中B为期望得到的哈希编码。求解上式可以通过随机初始化R，固定R求解B，再固定B求解R的步骤迭代进行直至达到局部最优解。

ITQ具有计算复杂度低，检索性能高的优点[[[39]](#endnote-39)]，但它也有一个明显的缺点即为了提升检索性能所做的平衡方差的工作造成了信息量的丢失。

### 2.3 本章小结

本章主要对目前的推荐系统和哈希技术相关领域的的国内外研究现状和成果做了介绍和总结。首先，介绍了个性化推荐系统，包括基于内容的推荐，基于协同过滤的推荐和基于混合技术的推荐，其中重点介绍了协同过滤相关技术，基于协同过滤的推荐又主要包括基于内存的推荐和基于模型的推荐，我们对基于内存的推荐的两种主要的方法做了比较详尽的介绍，也对这两种方法的优缺点做了分析。接下来，我们又介绍了基于模型的推荐中的代表方法，给出了几种基于模型的推荐方法的主要形式。最后，我们介绍了哈希学习的发展和现状，并且给出了局部敏感哈希和迭代量化哈希这两种经典的哈希方法的介绍。

## 第三章 两阶段联合的哈希算法

### 3.1问题描述

在传统的基于协同过滤的方法中，用户的向量的长度为系统中的项目的数量，同理，项目的向量长度也为用户的数量，在大规模系统，用户或者向量的数量可能会非常大，这就导致了用户或项目的向量非常高维，这对于推荐过程的相似度计算来说是一个巨大的挑战，而且，系统中的用户或项目的向量往往是稀疏的，这也会导致相似度计算的准确度的降低。为了解决上述问题，本文研究了哈希方法在协同过滤推荐系统中的应用，哈希方法的主要目标是通过学习一系列的哈希函数，来为系统中的用户或项目编码，从而将原始的高维向量降到低维空间。

为了进一步的将所得的哈希编码应用到推荐过程，学习到的哈希编码必须保留评分数据上用户对项目的偏好，即用户偏好的项目与该用户的海明距离小，相反的，用户没有表达出偏好的项目应该与该用户有着较大的海明距离。遵循上述的思想，我们首先对评分数据采用降维然后量化的方式生成其中一个视角的哈希码，然后建立偏好与海明距离的模型，通过模型建立度量方程，对度量方程求解得到另一视角的哈希码，从而完成对两个视角的哈希编码，并且用户与项目编码之间的海明距离反映了用户对项目的偏好程度。

### 3.2 基于用户视角的两阶段联合哈希算法

假定系统中用户数量为m，项目数量为n，用户对项目的评分构成的评分矩阵为Sm，其中表示用户i对项目j的评分，分值越高表示用户越偏好该项目，向量=()为用户i的向量表示，向量=(,,…)为项目j的向量表示，哈希后的用户空间为U：{}，哈希后的项目空间为V: {,,…}。tr(.)表示矩阵的迹，||.||F表示矩阵的Frobenius范式，sgn()为符号函数，输入大于等于0时输出1反之输出-1。

1）算法的第一阶段在用户视角降维，将用户映射到海明空间，首先在用户视角应用主成分分析技术，得到c位的用户的特征向量，对之量化即可得到c位的用户的二值码，为了减少量化过程的信息损失，引用[37]中的迭代量化策略。具体的做法如下：

(1). 在用户空间应用PCA，得到c位的用户的特征向量F

(2). 建立量化过程的损失方程：

(3-1)

其中R为正交的旋转矩阵，量化后的用户空间为U，随机初始化R后，交替执行如下两个过程直至得到局部最优解，得到最优的U：保持R不变, 按照U=sgn(FR)来更新U；保持U不变，对U进行奇异值分解（Singular Value Decomposition, SVD）为, 按照来更新R。

2). 算法的第二阶段通过系统中的评分来约束海明空间中的用户与项目的距离，即以海明空间中用户与项目的相似度来预测用户对项目的偏好。参照[18]中用户与项目的相似度定义，如下：

sim(i,j) =

=( + c -) (3-2)

= (c +) = +

其中I()为指示函数，如果条件为真则返回1，否则返回0。由上面的定义可知，当用户与项目的海明距离越小时，相似度越接近1；距离越大时，相似度越接近于0，基于这个性质可以建立如下的损失方程：

(3-3)

s.t. V

通常情况下，需要在上式加上正则项αβ 来缓解过拟合，然而在本文的情景中，为了最大化信息熵和方差，对待求的二值码添加了去相关和平衡的约束，即U1=0，V1=0，U=mI, V=nI ，在约束条件下，正则项为常数故可以省略。另外，此处的S为原始评分放缩后的值，若原始评分S[mins, maxs],则放缩后的S = 2c(S - c)/(maxs - mins) - c。

在约束条件下，直接由上式求解最优的V 是一个NP-难问题，在此引入辅助集合V’={X|X1=0, =nI}, 距离d(V,V’)=,基于此松弛（3-3）式，如下：

(3-4)

s.t. V

通过调整足够大的参数α，使得X逼近V即d(V,V’)=0, 进而将难以求解的V的离散约束转移到连续的容易求解的X上，由约束条件的性质，上式可化简为：

(3-5)

s.t. V,

=nI, X1 = 0

将求解V,X的问题分解为求解V和X的两个子问题，交替优化这两个子问题直至收敛即可求得V的最优解。

在求解V的子问题中，保持X不变，对于V有：

(3-6)

其中为S中已知的用户对项目的评分集合。由于的值是离散的，采用逐位更新的策略更新，更新规则如下：

= sgn (K(, ))

其中为的第k位，为中除了的其他部分，更新时保持不变， =） + α , 为X中第j行第k位，为中除了的其他部分，K(x, y) 函数的功能是：当x时，K（x, y）=x，否则K（x, y）=y。

对于X的子问题，保持V不变，由（3-5）式可得：

(3-7)

s.t. =nI, X1 = 0

欲求解上式，只需构造一个中心矩阵C，SVD（CV）=PΣ，X=P。

UTSH算法的主要步骤总结如下：

输入：评分矩阵，二值码位数c，权衡系数

输出：用户的二值码,项目的二值码

1. 

2. 随机初始化正交矩阵R

3. 交替执行①，②直至收敛

①

②,

4.初始化V，X，将S放缩

5.交替执行③，④直至收敛

③ for j=1 to n :

重复以下过程直至收敛

for k=1 to c :





④构造中心矩阵C

,

6.返回U,V

### 3.3 基于项目视角的两阶段联合哈希算法(ITSH)

基于项目视角的两阶段哈希算法为先从项目视角哈希，再通过评分约束用户与项目的海明距离，进一步得到用户的二值码的算法，由于其过程与上述基于用户视角的两阶段哈希算法对称，在此不再赘述。

### 3.4 本章小结

本章主要研究了两阶段联合的哈希算法，首先在评分上对其中一个视角应用主成分分析和迭代量化技术，生成该视角的二值码，也实现了对评分数据的全局特征的提取，接下来，利用评分和已经得到的二值码，建立损失方程，求解最优化问题，来迭代生成另一视角的二值码，进而实现对评分数据局部特征的捕捉。通过以上的过程，我们就生成了用户和项目的二值码，提出了两阶段联合的哈希算法。

## 第四章 基于两阶段联合哈希的推荐算法

传统的基于内存的协同过滤推荐技术通常是直接利用系统中的评分数据来计算用户或者项目的相似度。然而，在实际的系统中，无论是用户视角上还是项目视角上，用户向量与项目向量都是高维且稀疏的，基于这些高维稀疏向量的相似度计算不仅需要非常高的计算开销，进一步导致了计算推荐结果的速度下降，也因为其稀疏性降低了相似度计算的准确度带来了推荐质量的降低。

考虑到上述问题，我们可以用上文第三章中提出的基于两阶段联合的哈希技术来提高推荐系统的性能，利用训练好的用户与项目的二值码来进行高效的推荐，进而实现推荐算法在速度和准确度上的提升。

### 4.1 推荐结果生成

利用两阶段联合的哈希算法生成的保留了用户对项目的偏好的二进制哈希码，本文提出的基于两阶段联合哈希的推荐算法将用户之间或项目之间的相似度计算，转化为了用户与项目的二进制哈希码之间的相似度计算。

通过上一章中提出的基于两阶段联合的哈希算法，我们可以为系统中的用户和项目生成保留了用户与项目的偏好的二进制哈希码U,V，对于查询用户来说，我们可以直接用该用户对应的K位二进制与项目集上项目的二进制哈希码来进行相似度计算，而不是传统方法那样计算用户或项目的高维向量之间的相似度，接下来，我们会详细说明用二进制哈希码来做相似度计算的优势。事实上，对于二进制哈希码之间的相似度计算，一般是计算这两个二进制哈希码之间的海明距离，而海明距离衡量的是两个二进制编码对应位不同的数量。

为了进一步的减小存储的开销和计算开销，我们可以将训练得到的用{-1，+1}来编码的用户与项目的二进制哈希码转换为01编码，一方面，01编码与{-1，+1}编码实际上是等价的，另一方面，01编码的每一位的存储只需要用一个位，这样就可以大大的减少存储的开销，更进一步地，我们把转换后的01编码按照8位一组的方式对用户与项目的二进制哈希码进行分组，这样，每一组哈希码都对应着一个0-255之间的整数，那么K维的哈希码就可以用K/8个整数来表示。实际上，两个二进制哈希码之间的海明距离就是这两个编码进行异或运算后的编码中1的个数，为了方便计算，先建立一张大小为256的映射表，然后将0-255之间的每个整数与它所对应的二进制编码中1的个数建立映射。这样对于两个K维的二进制哈希吗的海明距离的计算，就只需要进行K/8次异或运算，然后将每一次异或的结果所对应的二进制编码中1的个数相加，从而大大的提高了用户与项目的二进制哈希码之间的相似度计算。

在以上快速相似度计算的基础上，对于不同系统中项目数量规模大小的情况，我们提出了两种策略。对于系统中项目数量较少的情况， 可以直接计算查询用户与项目集上的每一个项目的二进制编码之间的相似度，然后选取海明距离较小的项目推荐给查询用户。对于系统中项目数量较多的情况，如果直接计算查询用户与整个项目集上的项目之间的相似度，这依然是一个计算量非常大的策略，此时，我们使用哈希查询[]的方法生成查询用户的候选项目集合，然后与候选项目集合中的每一个项目进行海明距离的计算，选取海明距离最小的若干个项目推荐给查询用户。具体的做法是：首先，用二进制哈希码为键为用户建立一张哈希表，来完成二进制哈希码—项目的映射。对于查询用户来说，我们不会直接计算该用户与整个项目集上的项目的海明距离，而是根据该用户的二进制编码和设定好的阈值生成一个以该用户为中心的汉明球，然后我们就可以通过建立的哈希表找到汉明球内编码所对应的项目，进而得到查询用户所对应的候选偏好项目集合。然后在候选项目集合上来查询该用户可能偏好的项目来为该用户推荐，这种通过候选集来查询的策略大大减小了所需要的海明距离计算量，也进一步大大提高了推荐的速度和效率。

### 4.2 算法整体结构

在上文中，我们已经介绍和分析了上一章的两阶段联合的哈希算法和4.1节提出的由项目集上生成推荐的方法，将这两种方法整合起来，就构成了我们的基于两阶段联合哈希的推荐算法，主要结构如图4.1所示：



图4-1

本文提出的基于两阶段联合哈希的推荐算法可以分为训练和推荐两个过程。在训练过程中，我们根据系统中的评分数据由两阶段联合的哈希算法分别生成用户与项目的二进制哈希码。对于查询用户，我们可以直接取出该用户的二进制哈希码，然后利用4-1中提到方法快速生成该用户的偏好项目的候选集合，然后在该集合上搜索与该用户具有最小海明距离的若干个项目推荐给用户。

### 4.3 本章小结

本章主要提出了基于两阶段联合哈希的推荐算法。算法主要利用了第三章提出的两阶段联合的哈希算法，通过哈希技术将系统中的用户与项目训练生成二值码，然后利用二进制哈希码的快速的相似度计算性质，实现了高效的对查询用户预测偏好项目的检索，进而实现了快速高效的为查询用户提供可能偏好项目的推荐。

本章首先描述了实际应用中传统的基于内存的协同过滤技术，分析了提出的算法在top-k推荐任务上的优势。然后，介绍和分析了推荐结果的生成过程和其中的细节。最后，将算法的各个阶段整合起来，提出了一种基于两阶段联合哈希的协同过滤推荐算法，给出了算法的主要流程。

## 第五章 实验设计与分析

为验证算法的有效性，对UTSH与ITSH、ITQ和二值化的矩阵分解算法binMF[[[40]](#endnote-40)]进行仿真试验。其中ITQ为分别在用户和项目视角上哈希，来获得用户和项目的二值码;binMF为二值化的基于交替最小二乘法的矩阵分解算法，即对矩阵分解算法生成的特征表示以中位数作为阈值进行量化，得到用户和项目的二值码。

### 5.1 实验数据集

仿真实验数据选择MovieLen-1M数据集[[[41]](#endnote-41)]。MovieLens-1M 数据集共包括了6040个用户对3900部电影的1000209个评分，可以看出只有大约4%的用户-电影存在评分。评分是从1到5的整数，评分越低表示用户对该电影的偏好程度越低，反之，评分越高则表示用户对该电影偏好程度较强。通常，从数据集中随机选取80%的记录作为训练数据，余下的作为测试数据来验证算法效果。由于选取了归一化折损累计增益（Normalized Discounted Cumulative Gain, NDCG）来作为评估指标，这要求过滤掉数据集中记录过少的用户，以确保这一指标的有效性。过滤掉评分数低于5的电影的评分，并且选取评分数超过20的用户，选取每个用户的80%记录作为训练数据，余下的20%作为验证数据。对数据集进行5次同样比例的随机划分，并取5次实验结果的均值用来评估。

### 5.2 评价指标

通常的推荐算法采用均方根误差RMSE（Root-mean-square-error）来评估算法的性能，这个指标衡量算法在训练集上生成的预测评分与验证集中的真实评分的差距。然而在现实的推荐系统中，通常只为目标用户推荐预测评分较高的项目，而预测评分低的项目通常不会被推荐给用户，因此，RMSE在推荐任务上并不是一个最优的度量标准。

本文选取[[[42]](#endnote-42)]中提出的归一化折损累计增益NDCG作为实验的度量标准，它是一个建立在折损累计增益上的指标。具体来说，对于任意一个用户，算法为该用户返回的前K个推荐结果在验证集中的实际评分为{r1,r2,r3,…,rK},则对应的折损累计增益为：

DCG@K= （4-1） 根据该用户在验证集中的实际评分，选取其中最大的K个评分并按照降序排列，即为理想的项目评分{R1，R2，…,RK}, 由此可计算此用户的理想折损累计增益为：

IDCG@K= （4-2）

最终，可以得到该用户的归一化折损累计增益为：

NDCG@K= （4-3）

在接下来的实验中，计算在不同算法上所有用户的归一化折损累计增益，并取均值作为当前算法的归一化折损累计增益。

### 5.3 实验结果

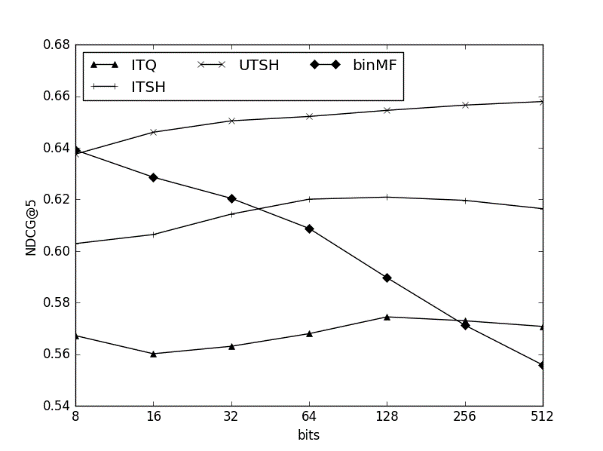


图5-1 top-5推荐时各算法的折损累计增益比较

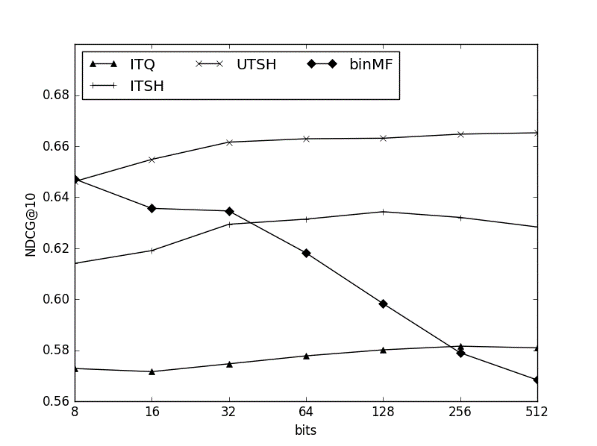


图5-2 top-10推荐时各算法的折损累计增益比较

实验的结果如图5-1、图5-2，

（1）提出的UTSH算法NDCG要高于其他算法，其曲线增长平缓，在小于128位时就有较高的ndcg值，反应了算法受编码的位数影响较小，且用较少位数的编码就能取得较好的性能，说明UTSH算法的推荐性能好而且存储代价小。

（2）binMF算法的性能随着位数增加呈现比较明显的下降，一是因为训练集的稀疏性导致算法在量化阶段损失了大量信息，二是NDCG着重衡量top-k推荐的质量，而binMF算法着重于对整体未知评分的预测，可知矩阵分解算法在高稀疏性的数据集上进行top-k推荐任务中表现不如本文所提出算法。

（3）在三种哈希算法中，在NDCG@5情况下，ITSH比ITQ提高了7.53%，UTSH也比ITQ提高了12.69%。实际上，ITSH算法与ITQ算法在项目视角的哈希学习过程是基本一致的，性能上的差异体现在用户视角上采用的监督式哈希方法，由于数据的稀疏性，直接在评分数据集上应用ITQ编码会损失大量的信息，为了保留更多的评分中的信息，ITSH在第二阶段用评分约束用户与项目的海明距离，通过监督式的哈希算法获得了高效的用户编码。同理，UTSH在项目视角采用了监督式的哈希算法，也取得了更好的性能。

（4）UTSH 比ITSH的NDCG高出了5.58%（NDCG@5），这说明数据的稀疏性对ITQ算法的性能的影响较大，也进一步体现了两阶段哈希算法的优势，TSH不仅充分的利用了评分数据的信息，而且针对评分数据各视角的稀疏性特点进行了针对性的处理，进一步提高了推荐的质量。

### 本章小结

在本章中，我们通过在数据集上的实验验证了本文提出的两阶段联合哈希的协同过滤算法的效果。我们首先介绍了实验所采用的数据集和实验结果的评价指标。然后，我们介绍了拿来作为基准的一些现有的协同过滤推荐算法。最后，通过在数据集上的仿真实验结果的对比，我们验证了本文所提出的算法的有效性，证明了本文算法相对于传统的协同过滤推荐算法的优势。

## 第六章 总结与展望

### 4.1 本文总结

针对在大数据时代背景下，传统的依赖于相似性度量和近邻检索的推荐算法在面对系统存在的海量高维数据时表现出计算量大，推荐效率低的问题，提出了一种基于两阶段联合哈希技术的推荐算法研究，主要包括用户与项目的二进制哈希码的学习和基于生成的哈希码的推荐两方面的工作，并在数据集上验证了所提出的算法的有效性。

对于用户与项目的二进制哈希码的学习来说，我们采用先对评分数据从用户或项目视角应用主成分分析技术和迭代量化技术生成该视角的二值码，在这个过程中，对评分数据进行降维和量化，这是对评分数据在整体上的一种处理，这有利于获取评分数据的全局特征；然后用评分约束用户与项目在海明空间的距离，即建立评分与用户/项目相似度的损失函数，求解上述的最优化问题，即可生成另一视角的二值码，这一过程着重于对评分中每一个评分的考察，可以很好的挖掘评分数据的局部特征，通过以上的两个过程就完成了用户和项目的哈希码的学习；对于基于生成的哈希码的推荐来说，我们利用学习得到的用户与项目的哈希码来完成进一步的推荐，具体来说，对于系统中项目数量较少的情况， 可以直接计算查询用户与项目集上的每一个项目的二进制编码之间的相似度，然后选取海明距离较小的项目推荐给查询用户。对于系统中项目数量较多的情况，则使用哈希查询的方法来完成推荐。通过在数据集上的实验，我们考察了所提出的算法与现有方法的对比，实验的结果表明了本文所做的工作相对现有方法在推荐性能上的优势。

本文主要提出了先对用户或项目视角进行哈希，用评分来约束哈希后的距离，进而对另一视角进行哈希编码的方法。一方面，将传统的相似度计算问题转化为高效的二值码检索问题，极大的减少了计算和存储开销；另一方面，基于两阶段联合哈希的推荐算法不仅能够有效降低稀疏性对推荐性能的影响，也为利用除评分外的信息来编码提供了可能，这有利于进一步提高推荐系统的性能。

### 4.2 未来研究方向

本文主要提出了一种针对评分数据来对用户和项目进行编码，利用高效的哈希码之间的相似度计算来完成相似度检索的任务的方法，在研究的过程中，我们也发现了本文所做的工作的不足之处，这有待于接下来的工作或研究中予以解决或改进。

首先，本文所做的工作是完全依赖于评分数据的，没有将评分以外的信息加入到编码过程中，对推荐性能的提升有限。在当今的大数据时代，系统中的数据呈现出多源异构的特征，因此，利用好评分以外的多视图的信息来生成二值码，以进一步解决稀疏性问题无疑是将来的一个非常重要的研究方向。

此外，利用编码后的用户/项目的哈希码来进行推荐可以大大的减少计算量，但是，算法在离线的对用户/项目进行编码的过程依然是非常耗时的，这对于实时性要求较高的场景下，必然是一个挑战。因此采用分布式的技术来进行编码的训练是非常有必要的，哈希编码的分布式实现将是进一步要完成的工作。

## 参考文献

## 在校研究成果

作者在攻读硕士学位期间的研究成果

[1]张辉宜,侯耀祖，陶陶.一种两阶段联合哈希的协同过滤算法[J].计算机工程,2018(已录用)

## 致谢

写到这里，论文工作基本完成，也宣告着我的学生时代即将结束。这三年来，无论是在学习上还是思想上，自己的成长是真真切切感受得到的。在学习上，自己对计算机科学技术的发展有了更为全面和深入的了解，也提高了自己的科研水平和实践能力，为自己下一步的职业规划和发展起到了良好的铺垫作用。在思想上，接受了师长的教诲和指导，也与同学朋友不断的深入交流，丰富了自己的精神世界，也让自己更加坚毅和温和。这期间也有过迷茫，有过焦虑，但很庆幸自己坚持到了现在，我想这一刻总算没有辜负家长的期许，自己的决心和师长朋友的殷切关怀。

首先感谢我的导师张辉宜教授，他从一开始就给了我选择的权利，让我能够以兴趣作为指导，并且结合自己的学术基础，选择更加适合自己的学术研究方向。在研究的过程中，也多谢他不断催促和提醒，让我能够战胜自己的懒惰和拖延，按时完成学术研究。更要感谢的是他对我研究方法上的指导，让我懂得并且掌握做研究和做事的思路和方法。张老师对我课题上的指导不仅在我的研究生生涯起着至关重要的作用，也会在我之后的人生中起着指引的作用。

无论何时何地，我都要感谢我的家人，感谢父母的理解和支持，感谢弟弟的成熟与担当，让我能够以比较放松的状态来完成研究。也要感谢他们对我的期许，鞭策着我学习更多的知识和技能，不断完善和提高自己。

在三年的学术生涯中，如果说跟有些人朝夕相处，在一块的时间和精力超过了老师，超过了女朋友，甚至超过了父母家人，那么这些人肯定就是同学和朋友了。感谢实验室的师弟张恒同学，室友华冰涛同学和辛文飞，王良萸，刘东旭同学，他们在我困难时给了我帮助，在闲暇时丰富了我的生活，希望我们的友谊一直持续下去。

感谢我的母校，在这里已经度过了将近七年的时光，七年可以说是非常非常久了，我在这里的每一点进步和成长，都依托于这个环境和平台，都有它的印记。

最后，感谢我的女朋友，对我很重要的李同学，有了她我的研究生生活才告别了苦闷和焦躁，也有了她我的奋斗目标才更加明确和深远。感谢她一直以来的陪伴与理解，让我的三年研究生生涯‘日长飞絮轻’。

感谢各位专家对论文的审阅！

1. []李武军, 周志华. 大数据哈希学习:现状与趋势[J]. 科学通报, 2015, 60(5):485-490. [↑](#endnote-ref-1)
2. []Wang J, Liu W, Kumar S, et al. Learning to Hash for Indexing Big Data—A Survey[J]. Proceedings of the IEEE, 2015, 104(1):34-57. [↑](#endnote-ref-2)
3. []https://www.taobao.com/ [↑](#endnote-ref-3)
4. []http://www.dangdang.com/ [↑](#endnote-ref-4)
5. []Ricci F, Rokach L, Shapira B, et al. Recommender Systems Handbook[M]// Recommender systems handbook. Springer, 2011:1-35. [↑](#endnote-ref-5)
6. []Zhao B Y, Kubiatowicz J D, Joseph A D. Tapestry: An Infastructure for Fault-tolerant Wide-area Location and Routing[C]// University of California at Berkeley, 2001. [↑](#endnote-ref-6)
7. []Resnick P, Iacovou N, Suchak M, et al. GroupLens:an open architecture for collaborative filtering of netnews[C]// ACM Conference on Computer Supported Cooperative Work. ACM, 1994:175-186. [↑](#endnote-ref-7)
8. []Koren Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model[C]// ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2008:426-434. [↑](#endnote-ref-8)
9. []孙小华. 协同过滤系统的稀疏性与冷启动问题研究[D]. 浙江大学, 2005. [↑](#endnote-ref-9)
10. []Sarwar B, Karypis G, Konstan J, et al. Item-based collaborative filtering recommendation algorithms[C]// International Conference on World Wide Web. ACM, 2001:285-295. [↑](#endnote-ref-10)
11. []金登科. 基于多视图锚点图哈希技术的推荐算法研究[D]. 浙江大学, 2016. [↑](#endnote-ref-11)
12. []Sarwar B. Applicatin of dimensionality reduction in recommender systems[J]. In Acm Webkdd Workshop, 2000. [↑](#endnote-ref-12)
13. []孟宪福, 陈莉. 基于贝叶斯理论的协同过滤推荐算法[J]. 计算机应用, 2009, 29(10):2733-2735. [↑](#endnote-ref-13)
14. []Sarwar B M, Riedl J, Konstan J. Recommender Systems for Large-scale E-Commerce: Scalable[C]// Conference on Computer and Information Technology. 2002. [↑](#endnote-ref-14)
15. []Agrawal R, Imielinski T, Swami A N. Mining Association Rules Between Sets of Items in Large Databases, SIGMOD Conference[M]// Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data. 1993:207--216. [↑](#endnote-ref-15)
16. []Hartigan J A, Wong M A. Algorithm AS 136: A K-Means Clustering Algorithm[J]. Applied Statistics, 1979, 28(1):100-108. [↑](#endnote-ref-16)
17. []Agrawal, Rakesh, Srikant, et al. Fast algorithms for mining association rules[J]. Proc.of VLDB, 1994, 15(6):487-499. [↑](#endnote-ref-17)
18. []Zhou K, Zha H. Learning binary codes for collaborative filtering[C]// Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2012:498-506. [↑](#endnote-ref-18)
19. []Zhang D, Wang F, Si L. Composite hashing with multiple information sources[C]. In: Proceedings of the 34th ACM Conference on Research and Development in Information Retrieval (SIGIR), Beijing, 2011, 225–234. [↑](#endnote-ref-19)
20. [] Xu H, Wang J, Li Z, et al. Complementary hashing for approximate nearest neighbor search[C].In: Proceedings of the 13rd IEEE International Conference on Computer Vision (ICCV), Barcelona, 2011, 1631–1638. [↑](#endnote-ref-20)
21. []杨根茂. 基于哈希加速的近似最近邻检索算法研究[D]. 浙江大学, 2015. [↑](#endnote-ref-21)
22. []Datar M, Immorlica N, Indyk P, et al. Locality-sensitive hashing scheme based on p-stable distributions[C]// Twentieth Symposium on Computational Geometry. ACM, 2004:253-262. [↑](#endnote-ref-22)
23. []Liu W, Wang J, Kumar S, et al. Hashing with Graphs[C]// International Conference on Machine Learning, ICML 2011, Bellevue, Washington, Usa, June 28 - July. DBLP, 2011:1-8. [↑](#endnote-ref-23)
24. []Kulis B, Grauman K. Kernelized locality-sensitive hashing for scalable image search[C]// IEEE, International Conference on Computer Vision. IEEE, 2010:2130-2137. [↑](#endnote-ref-24)
25. []Shen F, Shen C, Shi Q, et al. Inductive Hashing on Manifolds[C]// IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2013:1562-1569. [↑](#endnote-ref-25)
26. []Kulis B, Grauman K. Kernelized Locality-Sensitive Hashing[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2012, 34(6):1092-104. [↑](#endnote-ref-26)
27. []Jain P, Kulis B, Grauman K. Fast image search for learned metrics[C]// Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. IEEE, 2009:1-8. [↑](#endnote-ref-27)
28. []Jain P, Kulis B, Dhillon I S, et al. Online metric learning and fast similarity search[J]. Advances in Neural Information Processing Systems, 2008:761-768. [↑](#endnote-ref-28)
29. []Kulis B, Jain P, Grauman K. Fast Similarity Search for Learned Metrics[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2009, 31(12):2143. [↑](#endnote-ref-29)
30. []Liu W, Wang J, Ji R, et al. Supervised hashing with kernels[C]// Computer Vision and Pattern Recognition. IEEE, 2012:2074-2081. [↑](#endnote-ref-30)
31. []Norouzi M, Fleet D J, Salakhutdinov R. Hamming distance metric learning[C]// International Conference on Neural Information Processing Systems. Curran Associates Inc. 2012:1061-1069. [↑](#endnote-ref-31)
32. []Shen F, Shen C, Liu W, et al. Supervised Discrete Hashing[J]. 2015:37-45. [↑](#endnote-ref-32)
33. []Weiss Y, Torralba A, Fergus R. Spectral hashing[C]// International Conference on Neural Information Processing Systems. Curran Associates Inc. 2008:1753-1760. [↑](#endnote-ref-33)
34. []Zhang Z, Wang Q, Ruan L, et al. Preference preserving hashing for efficient recommendation[C]// International ACM SIGIR Conference on Research & Development in Information Retrieval. ACM, 2014:183-192. [↑](#endnote-ref-34)
35. []Liu X, He J, Deng C, et al. Collaborative Hashing[C]// IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2014:2147-2154. [↑](#endnote-ref-35)
36. []Zhang H, Shen F, Liu W, et al. Discrete Collaborative Filtering[C]// International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 2016:325-334. [↑](#endnote-ref-36)
37. [37]Gong Y, Lazebnik S. Iterative quantization: A procrustean approach to learning binary codes[C]// IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2011:817-824. [↑](#endnote-ref-37)
38. [38]Raiko T, Ilin A, Karhunen J. Principal Component Analysis for Large Scale Problems with Lots of Missing Values[C]// European Conference on Machine Learning. Springer-Verlag, 2007:691-698. [↑](#endnote-ref-38)
39. []邓慧茹. 面向大规模视觉检索的哈希学习[D]. 西安电子科技大学, 2014. [↑](#endnote-ref-39)
40. []Zhou Y, Wilkinson D, Schreiber R, et al. Large-Scale Parallel Collaborative Filtering for the Netflix Prize[C]// Proc. Int’l Conf. Algorithmic Aspects in Information and Management, Lncs. 2008:337--348. [↑](#endnote-ref-40)
41. []Harper F M, Konstan J A. The MovieLens Datasets: History and Context[M]. ACM, 2015. [↑](#endnote-ref-41)
42. []Balakrishnan S, Chopra S. Collaborative ranking[C]// ACM International Conference on Web Search and Data Mining. ACM, 2012:143-152. [↑](#endnote-ref-42)