本科毕业设计（论文）

动态推荐算法的研究与改进

Research and Improvement of Dynamic Recommendation Algorithm

学　　院：电子信息工程

专　　业：通信工程

学生姓名：黎斯思

学　　号：15211195

指导教师：陈一帅

北京交通大学

2019 年 5 月

学士论文版权使用授权书

本学士论文作者完全了解北京交通大学有关保留、使用学士论文的规定。特授权北京交通大学可以将学士论文的全部或部分内容编入有关数据库进行检索，提供阅览服务，并采用影印、缩印或扫描等复制手段保存、汇编以供查阅和借阅。

（保密的学位论文在解密后适用本授权说明）

学位论文作者签名：　　指导教师签名：

签字日期：　　年　　月　　日　　签字日期：　　年　　月　　日

中文摘要

摘要：大数据时代中，“信息过载”问题十分严重。推荐系统能够缓解“信息过载”问题，已得到广泛应用。传统推荐算法既不能很好地处理冷启动问题，也不能适应物品流行度及用户兴趣随时间动态变化的情况，推荐性能不佳。因此，动态推荐算法成为一个重要的研究热点。

多臂老虎机（Multi-Armed Bandit）算法是一种重要的在线学习算法。它在与用户不断交互实验的过程中动态调整优化自己的策略，得到最大收益，是一个动态的、在线更新的算法。其中，基于上下文的老虎机算法（Contextual-Bandit）中的 LinUCB 算法是一个经典算法。它引入特征向量，采用线性模型拟合用户对物品的兴趣。本文以新闻推荐为推荐算法工作的场景，针对 LinUCB 算法进行了深入研究和改进。主要贡献如下：

1. 实验评估了 LinUCB 算法在 Yahoo 新闻推荐数据集上的性能，发现了算法存在的利用和探索问题，即：当置信区间上界相差不大时，忽略了收益均值对性能的影响，损失了推荐系统的利用利益。
2. 为了解决该问题，将 LinUCB 算法与 Greedy 算法结合，提出了 GLinUCB 算法。该算法的基本思想是：在置信区间上界的差值与收益均值的差值之比满足小于参数 β 的条件时，主要考虑均值对系统性能的贡献。通过大量的实验，找到了 Yahoo 新闻推荐数据集下模型的最佳超参数。实验结果表明：改进后的 GLinUCB 算法推荐准确率为 6.67%，相较于 LinUCB 算法获得了 2.14%的性能增益。
3. 针对用户特征分布不均给推荐系统带来的挑战，实验评估了一种最新提出的基于 LinUCB 算法的分级算法，找到了当前数据集下模型的最佳超参数，并利用可视化手段对推荐过程进行了详细分析。实验结果表明，这种多推荐器系统推荐准确率为 6.75%，相较于单推荐器系统能够获得 3.37%的性能增益。

本文系统研究了 Contextual-Bandit 动态推荐算法的改进，发现了现有算法的不足，提出了新的算法，具有一定的理论价值和应用价值。

图 9 幅，表 6 个，参考文献 15 篇。

关键词：动态推荐；新闻推荐；推荐系统；上下文多臂老虎机；老虎机算法

ABSTRACT

ABSTRACT: In the era of big data, the problem of "information overload" is very serious. The recommendation system can alleviate the "information overload" problem and has been widely used. The traditional recommendation algorithm can not handle the cold start problem well, nor can it adapt to the popularity of the item and the dynamic change of the user's interest with time. The recommended performance is not good. Therefore, the dynamic recommendation algorithm has become an important research hotspot.

The Multi-Armed Bandit algorithm is an important online learning algorithm. It dynamically adjusts and optimizes its own strategy during the process of continuous interaction with users, and gets the most benefit. It is a dynamic, online update algorithm. Among them, the LinUCB algorithm in the context-based Contextual-Bandit algorithm is a classic algorithm. It introduces feature vectors and uses a linear model to fit the user's interest in the item. In this paper, the news recommendation is the recommended algorithm, and the LinUCB algorithm is deeply researched and improved. The main contributions are as follows:

1. The experiment evaluates the performance of LinUCB algorithm on the Yahoo news recommendation data set, and finds the use and exploration of the algorithm. That is, when the upper bound of the confidence interval is not much different, the impact of the average value of the performance on the performance is neglected, and the loss is lost. Recommend the benefits of the system.
2. In order to solve this problem, the LinUCB algorithm is combined with the Greedy algorithm, and the GLinUCB algorithm is proposed. The basic idea of the algorithm is that when the ratio of the difference between the upper bound of the confidence interval and the mean of the benefit satisfies the condition less than the parameter, the contribution of the mean to the performance of the system is mainly considered. Through a lot of experiments, I found the best hyperparameters for the model under the Yahoo News recommendation data set. The experimental results show that the improved GLinUCB algorithm has an accuracy of 6.67%, which is 2.14% better than the LinUCB algorithm.
3. Experiments evaluated a newly proposed classification algorithm based on LinUCB algorithm, found the optimal hyperparameters of the model under the current data set, and used the visualization method to analyze the recommendation process in detail. The experimental results show that the recommended accuracy of this multi-recommended

system is 6.75%, which is 3.37% performance gain compared to the single recommender system.

This paper systematically studies the improvement of Contextual-Bandit dynamic recommendation algorithm, finds the shortcomings of existing algorithms, and proposes a new algorithm, which has certain theoretical value and application value.

Figure 9, table 6,reference 15

KEYWORDS：Dynamic Recommendation System; News Recommendation; Contextual Bandit; Bandit

目　　录

1 引言

1.1 研究背景及意义

近年，随着互联网、移动终端设备技术的迅猛发展，“信息过载问题”变得非常严重。在当今时代，每个人既是信息的接收者也是信息的制造者，这使得网络生产信息的能力越来越强大，网络数据呈现爆炸式的增长，如何从海量数据中快速匹配出用户感兴趣的信息变得极具挑战。

大数据持续增长，人们对信息的使用率却在降低，这就是所谓的“信息过载”问题。一方面，虽然通过百度、谷歌等搜索引擎查找关键词可以对信息进行过滤从而获得需要的信息，但互联网仅仅是被动地向用户提供与关键词相关性大的内容，当信息分类不准确或者用户输入的关键词过少时，会增加用户的检索时间并影响检索结果，无法满足个性化与定制化的用户需求[1]。另一方面，一些电子商务平台、新闻门户以及视频网站等，作为信息的提供者，如何在众多商品中向用户提供更精准的推荐服务，提升用户体验并挖掘用户的潜在价值，这对提高企业的商业利益至关重要[2]。因此“信息过载”仍然是互联网面临的主要挑战。

推荐算法作为解决“信息过载”问题的一个重要技术，已经被广泛应用于互联网应用。推荐算法通过分析用户的行为数据（例如用户的浏览记录、鼠标的移动轨迹、点击记录等等），能够挖掘用户的潜在需求，捕捉用户的兴趣爱好，从而在其系统备选的服务中，选择满足用户个性化需求的商品进行推荐，成为解决“信息过载”问题的关键技术。Anderson 在其著作《长尾理论》[3]中给出这样的论断：“我们即将离开检索进入到推荐时代”。这将是一个互联网比你更了解你想要什么的时代。目前，推荐算法在互联网各领域取得了广泛的应用，并产生了巨大的商业利润。

推荐准确率是衡量一个推荐算法优劣最直观的标准，也是推荐系统的目标所在。如果推荐的服务符合用户的兴趣和需求，用户会大概率接受系统推荐的服务，并产生正向行为，例如在电商平台上购买推荐的商品、在新闻网站上点击查看推荐的，或是在视频网站点击观看推荐的视频。相反，如果推荐的服务用户不感兴趣，那么会极大的降低用户的使用体验，造成系统的用户流失，影响企业的长远利益。因此，如何提高推荐准确率成为了每个研究者孜孜不倦追求的首要目标。

目前，传统推荐算法如协同过滤和基于内容的推荐算法等，被广泛应用于个性化推荐系统中，但仍存在着一定的弊端。大部分传统推荐算法的基本思路是利用已经收集到的用户行为矩阵，通过计算相似度，对未知的物品评分进行预测，然后选取具有最高评

分的物品向用户进行推荐[4,5]。一方面，传统的推荐算法都基于一个假设：用户的兴趣不变，其建立的用户兴趣模型，主要依靠静态的数据，没有考虑用户行为数据自身的时间因素和因果因素。另一方面，传统的推荐算法的关键在于历史用户行为数据，这就导致算法在系统中出现新物品新用户的时候，数据稀疏带来冷启动的问题。以上两个问题影响着传统推荐算法的性能。

在真实的场景中，推荐系统是一个动态的系统。用户的兴趣会随季节时令、环境以及年龄的影响，随着时间的变化而改变。例如，当某一项运动的赛季结束，球迷们很有可能去关注另一项运动的赛事，而不会过多关注处在淡季的运动；青少年进入中学之后，可能不再满足于童话类的书籍，而是更加喜爱情感类或是科幻类的书籍……此外，用户的兴趣也受系统的影响，其行为数据中也包含系统本身的动态性。例如，在新闻推荐平台上，文章的种类是在时刻更新的，新闻的流行度具有高度的动态特性，新上架的热点文章很有可能激起用户的短期兴趣，并具备发展成为长期兴趣的可能。因此，动态推荐算法能够很好的解决系统冷启动的问题，考虑用户兴趣变化的动态推荐算法成为当下推荐领域的一个研究热点。

本文将以 Yahoo 新闻数据集作为数据背景，以新闻推荐作为具体的动态推荐场景，对基于上下文的多臂老虎机算法（Contextual-Bandit）作深入研究，并结合 Greedy 算法思想对其进行改进，提出一种半贪婪的上下文多臂老虎机（GLinUCB）算法；针对数据集用户特征不均的现象，对多级推荐器结构进行了实现。通过在数据集上进行大量的实验，结果表明，我们提出的改进算法 GLinUCB 和多级推荐结构，能够有效的提高推荐准确率。

1.2 国内外研究现状

动态推荐系统能够根据用户的行为数据，分析用户兴趣的变化，推荐用户可能感兴趣的物品。基于上下文的多臂老虎机算法本身具有不断探索用户兴趣变化的属性，能够发掘用户潜在的兴趣，为用户提供持续贴切的推荐服务。另外，在传统推荐算法中加入对用户兴趣迁移以及时间、地点等动态信息，也能在一定程度上适应传统推荐系统对算法动态性的要求。目前，主流的动态推荐算法分为两类：第一类是多臂老虎机算法，以基于上下文的多臂老虎机算法为代表，其中的 LinUCB 算法比较成熟，已被广泛应用在动态推荐系统上；第二类是对传统推荐算法的改进，加入时间敏感算法，弥补传统推荐算法对于用户兴趣动态变化带来的性能不足。

1.2.1 多臂老虎机算法

多臂老虎机算法（Multi-Armed Bandit），起源于赌博学，能够有效解决如何做出选择的问题。以新闻推荐场景为例，多臂老虎机算法将每一个待推荐文章看作是一个老虎机，从备选的文章中选择某个特定的文章进行推荐，相当于扳下对应老虎机的摇臂，这一动作对推荐系统带来的收益（点击率），等价于老虎机吐出来的金币。系统通过 Bandit 策略选择推荐的文章，从而达到收益最大化的目的。Bandit 算法是一种在线更新算法，能够根据用户的实时反馈调整自己的推荐策略，从而捕捉用户兴趣迁移以及备选文章上下架的动态变化。Bandit 算法的探索-开发策略能够有效应对冷启动，因而在工业界得到广泛的应用。

Bandit 算法具有多种变种算法。ϵ-greedy 算法是一种最基本的 Bandit 算法，该算法的推荐策略是将 ϵ 作为权衡探索和开发的一个概率参数，具有复杂度较低的优点，但是准确率有待提升。置信区间算法（Upper Confidence Bound）作为另一类算法，与简单的随机探索策略不同，UCB 算法根据文章的期望收益与置信区间，选择两者之和最高的文章进行推荐。在置信区间算法的基础上，Yahoo 的数据科学家们[6]提出了 LinUCB（Linear Upper Confidence Bound）算法，引入用户的特征，充分利用推荐场景的上下文信息，能够满足个性化推荐的要求，具有较高的推荐准确率，并将其成功应用到 Yahoo 新闻的个性化推荐系统中。由于创造性地引入了用户特征，这一类算法被称为基于上下文的多臂老虎机算法（Contextual-Bandit）。Cheng S 等人[7]在此基础上提出了一种融合矩阵分解的 MFLinUCB 算法，根据用户对商品的真实评价与预测评价之间的误差，应用矩阵分解算法更新用户和商品的特征，在新的特征作为输入的情况下，使用多臂老虎机策略进行推荐，取得了不错的效果。此外，Gentile C 等人[8]认为分析用户的社交关系有助于提高推荐准确率，在 LinUCB 算法上进行改进并提出了 CLUB（Cluster of Bandits）算法，引入用户协同思想，根据用户特征聚类的结果进行 Bandit 推荐，并根据每次反馈结果调整用户聚类。Qingyun Wu 等人[9]考虑了离散的时间因素对于 Bandit 算法的影响，提出了能够感知推荐环境动态变化的基于上下文的 Bandit 算法 dLinUCB（Dynamic Linear Bandit with Upper Confidence Bound）。同时，也有基于动态记忆网络的设计思路来计算用户偏好的 Bandit 推荐算法[10]，以上是对传统 Contextual-Bandit 算法的改进算法，其对数学模型进行了修正和优化，使算法性能得到了一定程度的提升。但是由于 Contextual-Bandit 算法依赖于一定的数学假设，在某些不符合假设的场景下表现效果不尽人意，因此数学模型的表达能力有限，这也是算法本身最大的不足之处。

1.2.2 传统推荐算法的改进

主流传统推荐算法大致分为三类：基于内容的推荐算法、协同过滤的推荐算法和混合推荐算法。基于内容的推荐算法通过计算用户兴趣特征向量与待推荐项目属性向量的

相似度，对用户产生项目预测评分或者得到 Top-N 项目序列。协同过滤的推荐算法是 Goldberg 等人[11]在 1992 年提出，是目前研究和应用最广泛的推荐算法，通过用户对项目的历史评分数据，通过计算用户之间的相似度或者项目之间的相似度，运用最近邻居算法计算用户的项目预测评分。混合推荐[12]将若干推荐算法通过一定的方式组合，能够适应不同的推荐场景，增强推荐结果。传统推荐算法具有推荐结果直观，算法简单等优点，但是由于其对于用户历史行为数据依赖性大，容易出现数据稀疏和冷启动的问题，没有考虑时效性对推荐系统的影响，在动态性较高的新闻推荐场景下很难取得理想的效果。一些学者提出了考虑时间因子的推荐算法，尝试利用时间信息来提高推荐准确率。ZimDars 等人[13]尝试将协同过滤推荐作为时间序列预测问题来考虑，这一想法启发了其他学者，Adomavicius 就是其中之一。Adomavicius 和 Koren 之后提出了一系列考虑时间信息的模型，对传统推荐算法进行了改进。Ding[14]在基于用户或者物品的协同过滤算法的基础上，增加时间指数衰减函数，对相似用户或者物品评分进行衰减处理，给予距离当前时刻最近的物品评分更多关注，而距离时间较远的物品相对较少的关注。这种方式能够直接的获取时间序列对推荐系统的影响效果。在 SVD++等算法中引入时间信息，也能够缓解原始矩阵分解没有考虑时间变化的问题。然而，考虑时间因素的改进算法大多只适用于用户兴趣变化具有一定规律并且变化节点时间距离较长变化速率较慢的推荐场景，传统算法面临的数据稀疏和冷启动的问题也仍然存在，因此具有一定的局限性。

综上所述，Contextual-Bandit 算法能够有效解决冷启动问题，并且由于其具有在线更新反馈的特性，在动态推荐场景下，相较于传统推荐算法具有更高的研究和应用价值。因此，本文将详细研究 Contextual-Bandit 算法，并针对算法存在的不足之处进行改进。

1.3 本文结构安排

本文组织结构如下：

第二章主要介绍了与本论文相关的技术背景。包括推荐系统中的 E&E 问题，以及常用于解决 E&E 问题的 Bandit 算法。随后对引入用户和物品特征的 Contextual-Bandit 算法进行了介绍，其中重点介绍了 LinUCB 算法的原理，并给出了 LinUCB 算法的伪代码。最后对本文用到的分析方法，包括 PCA 降维和 K-means 聚类技术进行了简要介绍。

第三章分析了新闻推荐场景的特点，以及推荐算法在此场景下面临的冷启动和用户兴趣高度动态变化的问题。指出了传统推荐算法在该场景下可能存在的问题以及动态新闻推荐算法的必要性。接着介绍了本文使用的数据集：Yahoo 新闻推荐的日志记录，同时分析了用户特征在空间内的分布情况。

第四章对 LinUCB 算法中利用和探索决策原理进行了介绍，并对其实现过程中出现的利用和探索决策不合理的情况进行了说明。针对其出现的上述问题，创新性地提出了

GLinUCB 算法，给出了算法的基本思想和算法流程。并且得到了算法的最佳参数，评估了算法的性能表现，并对于实验结果进行了可视化分析和比较。

第五章介绍了用户特征敏感的分级推荐算法，给出了算法的基本思想和伪代码。并且评估了该算法的性能表现，对结果进行了可视化分析。

第六章对本文的主要工作进行了概括，总结了本文的主要贡献，并对未来的研究方向进行了展望。

2 技术背景

本章将介绍相关技术知识。首先引入推荐系统中经典的 E&E 问题以及 Bandit 算法的简介。然后重点介绍 Contextual-bandit 算法的基本思想和核心原理，详细介绍 LinUCB 算法的实现过程，最后将介绍本文分析和处理数据用到的 PCA 降维方法和 K-means 聚类方法。

2.1 推荐系统中的 E&E 问题

E&E（Exploit&Explore）问题的产生源于每个用户对于不同分类的物品感兴趣程度是不一样的，系统中待推荐物品的种类也是多样的。传统推荐算法在获取用户历史行为数据之后，已经知道在已知物品中当前用户可能最感兴趣的物品，但是当系统中出现全新的物品，系统无法获取此物品的历史信息，也就无法判断此物品是否符合用户的兴趣，因此传统推荐算法倾向于继续推荐之前用户最感兴趣的物品。一方面，这将导致某类的商品被不断重复展示，形成了对用户取向的“追打效应”。另一方面，优秀的新物品将缺少展示的机会，曝光率将大大降低，甚至可能永远不会在用户的推荐列表中，即使它可能更加符合用户的兴趣。通过推荐已知的收益最高的物品来保证获得当前较高的收益，还是尝试探索未知的新物品，以一定的概率获得更优秀的推荐内容，寻求得到更高收益的可能性，这就是推荐系统中的 E&E 问题。一方面是利用（Exploit）已知的内容，取得已知的最大收益，另一方面是探索（Explore）未知的内容，去寻求更多的可能。

推荐系统中的 E&E 问题可分为三类。第一类是冷启动问题：对无历史信息的用户进行个性化推荐或是将新物品推荐给可能对其产生兴趣的群体时，面临着推荐准确率不高的难题。第二类是老用户的推荐选择问题，在物品具有分类特性的情况下，如何确定推荐的物品使系统获得最大收益。第三类是推荐策略选择问题，即如何比较推荐策略间的优劣。

Exploit 是一种贪婪算法，其优点在于可以充分利用已知高收益物品进行推荐，但是这也导致算法只能达到局部最优，错过潜在的更高收益机会。Explore 则倾向于随机算法，其优点在于能够探索潜在可能产生高回报的物品，缺点是同时具有可能探索到低回报的物品。对推荐系统来说，只考虑 Exploit 将会埋没推荐备选池中的优质资源，对已上架的商品过度推荐，如果只考虑 Explore 将会让用户很难体会到个性化的服务，影响用户的使用体验。

E&E 问题的解决思路基于对优秀的推荐系统的定义。优秀的推荐系统应该在不断利用已知高收益的物品的同时，不断探索新的物品推荐给用户，从而利用用户已知的兴趣，

探索其未知的兴趣。在保证用户感受到系统带来的精准推荐服务的前提下，拓展用户的潜在兴趣，给用户带来新鲜感和惊喜感，提高用户的使用体验和对系统的忠实度。因此推荐算法需要兼顾利用和探索，才能获得最大的收益。

2.2 Bandit 算法

多臂老虎机（Bandit）算法能够很好的解决 E&E 问题。Bandit 算法源于赌博学的 N 臂老虎机问题：有 N 个老虎机放在赌徒面前，从每一台老虎机得到的收益均不相同，每一轮他可以选择扳下一台老虎机的摇臂，并且获得所选老虎机带来的收益。经过多轮选择后，可以得到各个老虎机收益的统计信息，此时赌徒可以根据统计信息决定选择哪一台老虎机的摇臂，并且决定每个臂被选择多少次，最终使赌徒获得最大的收益。Bandit 算法假设每一台老虎机每一轮提供的收益于自身收益随机分布的函数相关，通过一定的选择策略使得算法能够精确计算老虎机被选择的顺序和次数。这其中就体现了利用（Exploit）和探索（Explore）之间的权衡，因此能够很好的解决推荐系统的 E&E 问题。

关于 Bandit 算法的研究比较多，主要包含以下几类：

1. ϵ-greedy 算法

该算法以一个介于 0 和 1 之间相对较小的参数 ϵ 来折中 Bandit 算法的利用和探索力度。每轮选择，算法以 1-ϵ 的概率进行利用（Exploit），选择当前收益最大的老虎机，以 ϵ 的概率进行探索（Explore），从其他老虎机中随机选择。当物品收益的不确定性较大时，例如概率分布较宽时，则需要更多的探索，此时需要更大的 ϵ 值，例如概率分布比较集中时，则少量的尝试就能获取近似真实的收益，此时需要的 ϵ 较小。由于多轮实验后，物品的收益已经近似真实收益，此时不再需要探索，可以令 ϵ 随实验次数的增加而逐渐减小。ϵ-greedy 算法在贪心算法中引入探索机制，是一种启发式的算法，但是由于 ϵ 参数的大小往往不好确定，系统取得的效果有限。

2. 置信区间上届算法（UCB）

UCB 算法使用置信区间来度量估计的不确定性，即置信区间越宽，不确定性越大，反之区间越窄，不确定性越小。每个物品的收益均值有一个置信区间，经过多轮选择后，物品收益均值的置信区间会变窄，即经过多次试验后，物品的收益均值估计的不确定性降低，同理，被选中次数较少的物品仍保持较大的置信区间，具有较大的不确定性。UCB 算法则根据每轮估计均值的置信区间，对物品进行排序，选择置信上限最大的的物品进行推荐。如果某些物品的置信区间很宽，则会倾向于被多次选择，这也是算法进行探索（Explore）的部分，对于已经进行过多次选择的物品，其收益的大小比较容易确定，算法将更加倾向于选择收益均值更大的物品，这是算法进行利用（Exploit）的部分。置信上限作为算法的关键部分，是收益均值与其标准差之和，计算公式如下：

$$Z = \bar{x}_j(t) + \sqrt{\frac{2\ln t}{T_{j,t}}} \tag{2-1}$$

式中 t——已经进行的选择次数；

$\bar{x}_j(t)$——物品 j 在前 t 次选择后得到的评价收益；

$T_{j,t}$——在前 t 次选择中 j 物品被选择的次数。

UCB 算法将开发和探索融为一体，能够智能调节探索的力度，因此能够获得较好的性能[15]。

3. Softmax 选择策略

Softmax 选择策略的基本思路是以高概率选择高收益物品，以低概率选择低收益物品。Softmax 算法中物品选择概率的分配是基于 Boltzmann 分布，概率计算公式如下：

$$P(k) = \frac{e^{\frac{Q(k)}{\tau}}}{\sum_{i=1}^{K} e^{\frac{Q(i)}{\tau}}} \tag{2-2}$$

式中 k——第 k 个物品；

$Q(k)$——物品 k 的估计平均收益；

τ——放缩参数；

K——物品总数。

同样是以概率的方式实现利用和探索的统一，Softmax 相较于 ϵ-greedy 引入了指数函数，放缩物品收益的差距，从而确定推荐的权重，根据推荐的权重确定物品被推荐的概率。当物品的收益差距缩小时，即 τ 较大时，算法更趋向于探索，当物品的收益差距被放大时，即 τ τ 较小时，算法趋向于利用。

4. 汤普森采样（Thompson sampling）

汤普森采样算法假设每个物品产生的收益均值，服从一个概率分布，通常为 Beta 分布，然后每轮选择，对每个物品利用其 Beta 概率分布抽取一个随机量，最终选择拥有最大随机量的物品，随后根据物品的实际收益情况更新 Beta 分布参数。通过不断的反馈修正算法的选择，从而得到每个物品接近正式的收益期望分布。

以上几类 Bandit 算法都能应对推荐系统中的 E&E 问题，由于具有在线更新的特性，也能较好的适用于动态推荐场景，Bandit 算法得到广泛的应用。

2.3 上下文 Contextual-Bandit 推荐算法

Bandit 算法虽然能通过不断的与环境进行交互，很好的处理冷启动问题，但是由于传统 Bandit 算法完全基于统计特性，没有考虑用户和物品的特征，推荐系统的个性化特

点不能得到充分的体现，因而达不到较好的推荐效果。基于上下文的多臂老虎机算法（Contextual-Bandit）则在 Bandit 算法上进行了改进，将用户的特征和物品的特征考虑在内，并取得了不错的表现。

2.3.1 Contextual-Bandit 基本原理

Contextual-Bandit 算法认为物品所获得收益期望与用户和物品的特征之间存在某种关系（例如，线性关系）。这使得算法能够充分利用用户和物品的上下文信息，根据当前用户的特征为每个待推荐物品计算其收益均值，然后权衡利用和探索的力度，给用户推荐其最感兴趣的物品，从而实现动态的个性化推荐。其算法过程如下：

一般来说，Contextual-Bandit 算法进行的是离散的试验，在第 $t(t=0,1,2\cdots\cdots)$ 轮试验中：

1. 算法先观察当前用户 u_t 和备选推荐的物品 A_t 以及他们的特征向量 $\mathrm{X}_{t,a}$，其中 $a \in A_t$，向量 $\mathrm{X}_{t,a}$ 即包含用户特征，也包含物品特征，也就是推荐场景的上下文信息；
2. 根据之前试验得到的收益，算法选择 $a_t \in A_t$，并获得对应的收益 r_{t,a_t}，其均值取决于用户 u_t 和物品 a_t；
3. 算法根据当前观察到的（X_{t,a_t}，a_t，r_{t,a_t}）更新推荐的策略，其他未被选择的物品不考虑更新。

2.3.2 LinUCB 算法

Yahoo 的科学家们在 2012 年介绍了动态新闻推荐领域的 Contextual-Bandit 算法，并假设物品的收益期望与用户和物品的特征之间为线性关系，提出了 LinUCB 算法。算法的基本思想是通过用户和物品特征与收益期望之间的线性关系，计算当前的收益期望，然后使用岭回归进行拟合，再根据 UCB 思想对用户进行个性化推荐。

根据 2.3.1 节中的介绍，假设物品 a 的收益均值与 d 维向量 $\mathrm{X}_{t,a}$ 以某一未知系数 θ_a^* 成线性关系，也就是对第 t 轮试验而言，

$$\mathrm{E}[r_{t,a} \left| \mathrm{X}_{t,a} \right.] = \mathrm{X}^{\mathrm{T}}{}_{t,a}\theta_a^* \tag{2-3}$$

其中对于不同的物品 a，参数不一样。设 D_a 为一个 $m \times d$ 矩阵，行对应 m 个训练输入（例如 a 物品的前 m 次推荐内容），$b_a \in R^m$ 是与之对应的反馈向量（例如对应的 m 次点击/未点击反馈）。对训练数据应用岭回归可得到系数的估计为：

$$\hat{\theta}_a = (D_a^{\mathrm{T}} D_a + I_d)^{-1} D_a^{\mathrm{T}} c_a \tag{2-4}$$

I_d 为 $d\times d$ 维单位矩阵，c_a 中的向量相互独立与 D_a 中的行向量相关。以至少1-δ 的概率有：

$$\left|\mathrm{X}_{t,a}^{\mathrm{T}}\hat{\theta}_a-\mathrm{E}[r_{t,a}\left|\mathrm{X}_{t,a}\right]\right|\le\alpha\sqrt{\mathrm{X}_{t,a}^{\mathrm{T}}(D_a^{\mathrm{T}}D_a+I_d)^{-1}\mathrm{X}_{t,a}} \tag{2-5}$$

其中 $\delta>0$，$\mathrm{X}_{t,a}\in R^d$，α=1+$\sqrt{\ln(2/\delta)/2}$ 是一个常量。上式给 UCB 的引入提供了一个契机，在每轮试验中，选择

$$a_t\overset{def}{=}\arg\max_{a\in A_t}(\mathrm{X}_{t,a}^{\mathrm{T}}\hat{\theta}_a+\alpha\sqrt{\mathrm{X}_{t,a}^{\mathrm{T}}A_a^{-1}\mathrm{X}_{t,a}}) \tag{2-6}$$

上式中 $A_a\overset{def}{=}D_a^{\mathrm{T}}D_a+I_d$。即每轮选择均值和置信区间上届最大的物品进行推荐。

LinUCB 经过每轮推荐后，根据用户的真实反馈，以及当前的上下文信息，对参数θ进行更新，从而使θ向量不断得到优化，提高模型的准确度，因此 LinUCB 是一个在线的算法。在新闻推荐场景中，该算法具有能够实时跟踪新闻的流行度，并以此更新推荐策略的优点。

LinUCB 的具体实现过程如下：

算法 1 Linear UCB

Algorithm1 Linear UCB

Algorithm 1: Linear UCB Bandit Algorithm

0: Input: a

1:**for** t=1,2,3….T, **do**

2: Observe the current feature x_t

3: **for** all a Î L:

4: **if** a is new **then** :

5: A_a ¬ I_d

6: b_a ¬ $0_{(d,1)}$

7: **end if**

8: q_a ¬ $A_a^{-1}b_a$

9: $p_{t,a}$ ¬ $q_a x_t+a\sqrt{x_t^T A_a^{-1}x_t}$

10: **end for**

11: Choose arm $a_t=\arg\max_{a\hat{\mathrm{I}}\mathrm{L}}p_{t,a}$ and observe a real-valued payoff $r_{t,a}$

12: A_a ¬ $A_a+x_t x_t^T$

13: b_a ¬ $b_a+r_{t,a}x_t$

14: **end for**

其中算法的第 4-6 行是对新物品的初始化，第 8-9 行计算物品收益期望的 UCB 值，第 11 行选择 UCB 值最大的文章进行推荐，并观察实际的收益情况，第 12-13 行根据实际收益情况对算法进行更新。

其中的符号含义如表 2-1 所示。

表 2-1 LinUCB 符号对照表

Table 2-1 Symbol description of LinUCB

符号	含义
L	候选物品集合
a	推荐集合中待推荐的物品
x_t	t 时刻用户特征
q_a	每个推荐物品的权重向量
$r_{t,a}$	在 t 时刻推荐 a 得到的真实回报
A_a	新闻 a 被过去被推荐过的特征记录矩阵
b_a	用物品 a 的回报反馈值

2.3.3 岭回归

岭回归（Ridge Regression）是一项线性回归技术，在最小二乘法的基础上增加了一个正则项。其中最小二乘法的目标函数为：

$$\min\left\|X\theta-Y\right\|^2 \tag{2-7}$$

通过梯度下降的方式经过多次迭代可得到局部最优解，也可以通过解线性方程组的方式得到闭式解：

$$\theta=(\mathrm{X}^{\mathrm{T}}\mathrm{X})^{-1}X^{\mathrm{T}}Y \tag{2-8}$$

由于矩阵 $\mathrm{X}^{\mathrm{T}}\mathrm{X}$ 为奇异矩阵时不满足可逆条件，影响最终解的精确度。岭回归通过在目标函数中加入正则项来解决这个问题：

$$\min\left\|X\theta-Y\right\|^2+\left\|\alpha\theta\right\|^2 \tag{2-9}$$

该目标函数的的解为：

$$\theta=(\mathrm{X}^{\mathrm{T}}\mathrm{X}+\alpha\mathrm{I})^{-1}X^{\mathrm{T}}Y \tag{2-10}$$

岭回归通过加入正则项，放弃最小二乘法的无偏性，虽然损失了部分信息，降低了一定的精度，但是得到的回归系数更为符合实际、更可靠。

2.4 PCA 降维方法

PCA（Principal Component Analysis），即主成分分析法，是一种常用的数据降维方

法。一般情况下，用户和物品的特征具有多个变量，往往是高维的，变量之间可能存在相关性，从而会增加问题分析的复杂性。如果分别分析每一个变量，得到的结果是孤立的，不能充分利用数据中的信息，所以如果单纯的缩减变量，会损失很多有效信息，影响分析结果。PCA 的主要思想是将高维特征映射到低维上，低维坐标轴具有正交特性，低维特征是在原有高维特征的基础上，根据方差最大原则重新构造形成的全新向量。因此降维后的特征相当于只保留包含绝大部分方差的维度特征，因此也被成为主成分。PCA 方法简单易于实现，因此被广泛应用于数据特征的降维处理。

2.5 K-means 聚类方法

K-means 是最简单的聚类算法之一，但是应用十分广泛。选取适当的 k 参数，将数据分类后能够更加直观方便的分析数据的分布等，然后分类研究不同聚类下数据各自的特点。K-means 算法的基本原理如下：

1. 随机选取 k 个聚类中心点；
2. 遍历所有数据，根据一定的距离计算公式，将每个数据划分到离它最近的中心点内，以中心点为核心形成簇；
3. 根据每个簇内的数据，计算平均值，作为新的中心点
4. 重复步骤 2、3，直到中心点收敛不再发生变化，或执行了指定次数的迭代。

2.6 本章小结

本章介绍了推荐系统中的 E&E 问题以及 Bandit 算法，重点介绍了 Contextual-Bandit 算法的基本思想及其重要的一种算法 LinUCB 的原理和实现过程，以及其中应用到的岭回归方法。此外，本章还介绍到了本文应用到的一些数据处理方法，PCA 降维方法和 K-means 聚类方法，其具体应用将在后面章节得到体现。

3 新闻推荐场景及数据介绍

本章基于著名的新闻门户网站 Yahoo 新闻的推荐日志，分析新闻推荐领域的动态特性，以及用户特征的分布规律。阐述了在高度动态变化的推荐场景下，传统推荐算法的不足，以及引入动态推荐算法的必要性。同时分析了用户特征分布不均匀的特点对当前推荐算法可能带来的问题。

3.1 新闻推荐系统

3.1.1 新闻推荐系统简介

目前，推荐系统在互联网领域的应用大致分为两大类，一类是商品提供平台，如亚马逊、淘宝和京东等，电商平台根据用户的注册信息、浏览、收藏和购买记录向用户推荐其可能感兴趣的商品。另一类是信息提供平台，即视频、图片、文章、新闻领域的推荐平台。 对于电商平台而言，夏季裙子和短裤类夏装相较于冬装对用户更具有吸引力，用户兴趣以及商品流行度相对稳定的时间跨度以季节为单位，可见用户的购物兴趣以及商品的流行程度虽然会随时间变化，但其变化通常比较缓慢，因而资源在推荐系统中存在的生命周期较长。对于新闻推荐系统而言，由于新闻具有突发性、高时效性的特点，使得其相对于其他推荐系统更具有动态性。

新闻推荐系统的动态性表现在两个方面：第一，新闻的突发性会使备选推荐文章池时刻处于变化之中，发生突发事件时，有关报道的新闻作为新文章加入到备选推荐文章池中；第二，新闻系统具有高时效性的特点，表现在一则新闻的存在的生命周期很短暂，短则几个小时，长则只有几天。由于每篇新闻其本身的流行度和变化趋势都不相同，备选推荐文章的上下架都是不同步的，如果不能较好的处理，会导致系统的冷启动问题，由此增加了算法对于动态变化的要求。

综上，新闻推荐场景具有高度动态性，推荐算法需要充分考虑到新闻流行度的变化，并根据此变化快速调整推荐的策略，更新文章的推荐优先级。

3.1.2 Yahoo 数据集介绍

本文选取 Yahoo 门户网站新闻推荐系统的离线日志数据作为算法的数据集。数据集记录了 2009 年 5 月 1 日到 2009 年 5 月 10 日 Yahoo 首页“Today”栏目的推荐日志，这

是一个采用随机推荐策略的离线日志数据集，详细的记录了当时的推荐详情。数据集一共包含十天的数据，其中每天的数据量越为 400 万到 500 万条，数据集包含了以下信息：

1. 时间戳（Timestamp）：记录了用户访问推荐系统的具体时间戳，也是推荐器执行随机推荐策略的时间，以秒为单位。
2. 新闻 id（News_id）：当前随机推荐策略具体推荐的文章。
3. 用户特征向量（User_feature）：当前访问推荐系统的用户的特征向量，其维度为六维。前五维为原始的用户历史数据采用 Logisitc 回归（LR）拟合得到的特征向量。第六维为常数 1，是偏置项。为提高算法效率，本文只考虑真正于用户信息相关的前五维分量。
4. 推荐结果（Result）：记录当前用户对此次推荐文章的反应，值为 1 或者 0。1 代表用户接受本次推荐，即产生了一次点击文章的行为。0 代表用户没有接受此次推荐，忽略了当前文章或者对当前文章不感兴趣。根据此项结果，可以统计推荐系统的点击率，并根据此来评估总体的推荐性能。
5. 推荐候选集（Candidate_Set）：列出当前时间点可用来被推荐的备选推荐文章，以及他们各自的特征向量。由于新闻系统的动态性变化，不同时间对于不同用户，其候选集是不同的。

本文将基于 Yahoo 数据集对 LinUCB 算法进行测试，并探寻动态推荐算法的一般模式和规律，并对算法进行改进，通过此离线数据集对算法进行评估。

3.2 用户特征分布

由 3.1.2 小节对数据集结构的介绍可知，用户的前五维特征包含用户特征的主要信息，第六维为偏置项。我们随机抽取了 5000 条数据，对用户的前五维分量做了降维处理和可视化分析。

我们用 PCA 主成分分析法，对随机抽取的 5000 名用户的前五维向量进行了降维处理。将用户的五维特征降为二维，可视化后得到的用户特征在空间内的分布如下图所示：

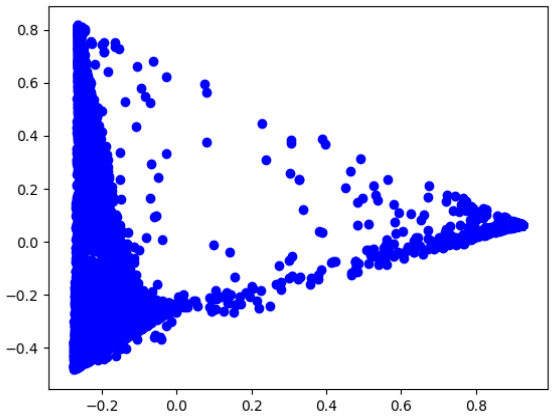

图 3-1 用户特征二维空间分布

由上图可知，用户特征在二维空间分布是不均匀的，用户集中分布在几个簇内。如果当前使用一个推荐器进行推荐，对于 LinUCB 算法而言，相当于使用分布不均匀的用户特征对算法的θ参数向量进行拟合，由于推荐策略是由所有用户共同训练更新得到的结果，用户特征的不均匀性必将对推荐系统的性能产生影响。

3.3 本章小结

本章对新闻推荐场景进行了详细介绍，并分析了其高度动态性的特点会带来的问题。一方面，新闻推荐平台备选文章具有更新速度快的特点，可能带来严重的冷启动问题。另一方面，新闻文章流行度的变化较快，需要推荐策略及时进行更新。新闻推荐以上两个问题，对推荐算法提出了更高的要求。通过对用户特征的降维和可视化分析，我们发现用户的特征具有不均匀的特点，该特点也对我们后续提出分级结构改进算法提供了一个启发。

4 GLinUCB 算法

本章针对在 LinUCB 算法实现过程中发现的利用和探索策略不合理问题，对算法进行了改进，结合 Greedy 思想提出了 GLinUCB 算法，该算法能够较为合理的权衡推荐策略中利用和探索的比重，可以使算法更好地适用于不同的推荐场景，提高推荐的准确率。最后我们针对数据集和算法的特点提出了离线评估方法，该方法能够有效评估算法性能，实验结果表明，GLinUCB 算法的推荐准确率得到较大的提高。

4.1 LinUCB 算法应用中的问题

4.1.1 LinUCB 算法中的利用和探索策略

作为 Contextual-Bandit 算法中最具代表性的 LinUCB 算法，因其对用户和待推荐文章的特征加以利用，且具有实时反馈更新的特点，在动态新闻推荐领域得到了广泛的应用。LinUCB 算法中采用了线性模型引入用户和物品的特征，充分利用推荐场景的上下文信息。然后根据 UCB 思想解决利用和探索的问题，以高斯分布为例，其基本思想如图 4-1 所示：

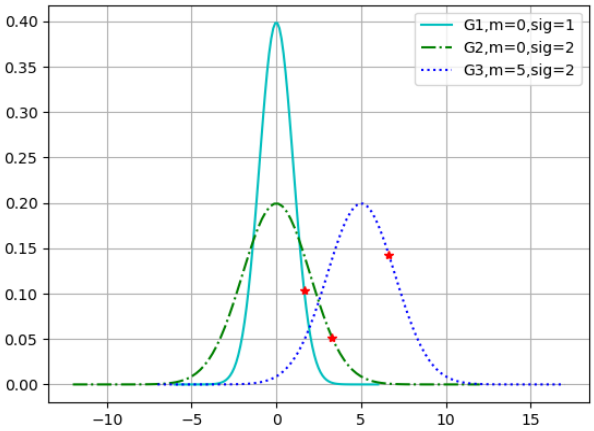

图 4-1 LinUCB 利用和探索策略示意图

上图中可观察到三条服从高斯分布的函数，其中函数 1 的均值为 0，标准差为 1，函数 2 的均值为 0，标准差为 2，函数 3 的均值为 5，标准差为 2。在置信水平为 0.9 的条件下，即 δ=0.1 的条件下，置信区间上届如图中红色的标点所示。在本文新闻推荐场景下，标准差的大小与待推荐文章可能获得收益的不确定性成正比，在收益均值相等的情况下，置信区间上界大的函数即为置信区间较宽的函数。也就是说当收益均值不等时，

算法倾向于选择均值与不确定性之和较大的文章进行探索，即在函数 1 和函数 2 中选择函数 2 代表的文章；在不确定性相等的情况下，选择收益均值较大的文章进行利用，即在函数 2 和函数 3 中选择函数 3 代表的文章进行推荐。总的来说，UCB 的思想即为选择置信区间上届最大的文章进行推荐。

置信区间上界同时考虑了均值大小和置信区间大小，通过置信度δ计算得到探索系数α的值。探索系数α作为算法的超参数，通过赋予其不同的值，可以调整推荐策略探索的力度，因此能较好地解决利用和探索的问题。

4.1.2 LinUCB 算法中的探索和利用问题及验证

4.1.1 介绍的思路虽然能够通过调整算法的α参数，在一定程度上解决探索和利用力度之间的矛盾，但是也有可能出现一些不合理的情况。

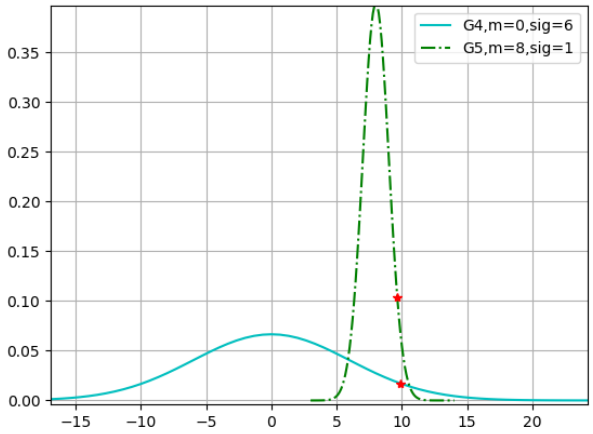

图 4-2 LinUCB 利用和探索不合理决策示意图

如上图所示，函数 4 的均值为 0，标准差为 6。函数 5 的均值为 8，标准差为 1。二者的置信区间上届十分接近，函数 4 的区间上届为 9.8，函数 5 的置信区间上届为 9.6，仅仅相差 0.2，然而两者的均值相差为 8，两者均值的差距是置信区间上届差距的 40 倍。虽然如此，UCB 的算法仍将会选择函数 4 代表的文章进行推荐，以损失较大的利用利益，换取微弱的探索利益，因此在这种情况下，利用和探索的不合理决策，必将影响到算法的性能表现。

应用 LinUCB 算法对 Yahoo 数据集中第一天的数据进行测试，通过筛选数据中点击行为为“1”，且当前随机策略推荐的有效文章 id 等于 LinUCB 算法推荐得分（置信区间上届值）列表排名第二的文章 id 的记录，得到共 11424 个数据。分析这些数据中推荐得分列表中前两篇文章，定义 degree 参数为他们的均值差距与得分差距的比值，假设得分前两篇文章分别为 A 和 B，degree 计算过程如下：

$$\deg ree = \frac{average_A - average_B}{rating_A - rating_B} \tag{3-1}$$

对其取对数，若 degree 为负数，则取其绝对值后再取对数的负数。可视化结果如下图：

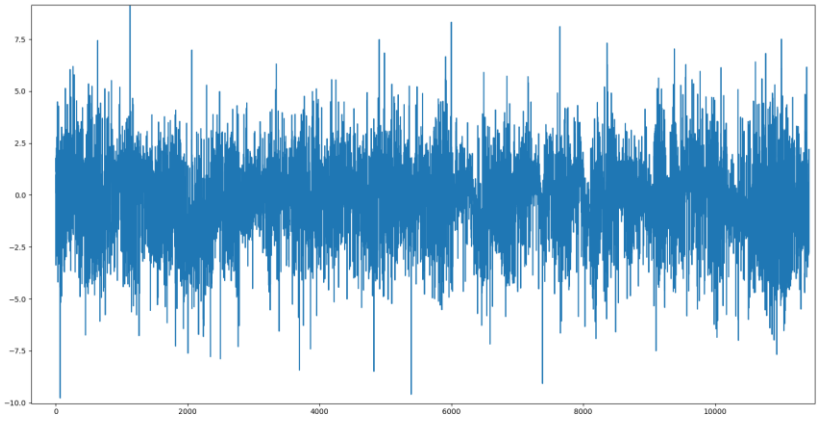

图 4-3 LinUCB 算法下 degree 值可视化结果

统计可知共有 5075 个结果为负数，即上图中位于负半轴的部分。负数数据占比为 44.4%。由此可知，LinUCB 算法中利用和探索决策的不合理，在新闻推荐场景下是真实存在的，也验证了上述分析的合理性。

4.2 算法的实现

4.2.1 Greedy 思想的引入

针对 4.1.2 节提到的置信区间上届相差较小的情况下可能出现的问题，我们联想到 Greedy 算法能够充分实现利用的优点，由此在 LinUCB 算法的基础上创造性地引入 Greedy 算法。在上述置信区间上届相差较小的情况出现时，推荐算法以 Greedy 策略为主，确保获得较大的利用收益。我们将使用 Greedy 思想改进后的算法命名为 GLinUCB 算法。

4.2.2 GLinUCB 算法流程

本小节将介绍 GLinUCB 算法的具体实现流程。根据上述几个小节分析，GLinUCB 对原有算法最大的改进之处在于文章的选取不再绝对满足于置信区间上界最大原则。当均值差与总分差之比，即 degree 参数满足一定条件时，则在推荐得分列表的前两篇文章中使用 Greedy 推荐策略。算法的流程如下图所示。

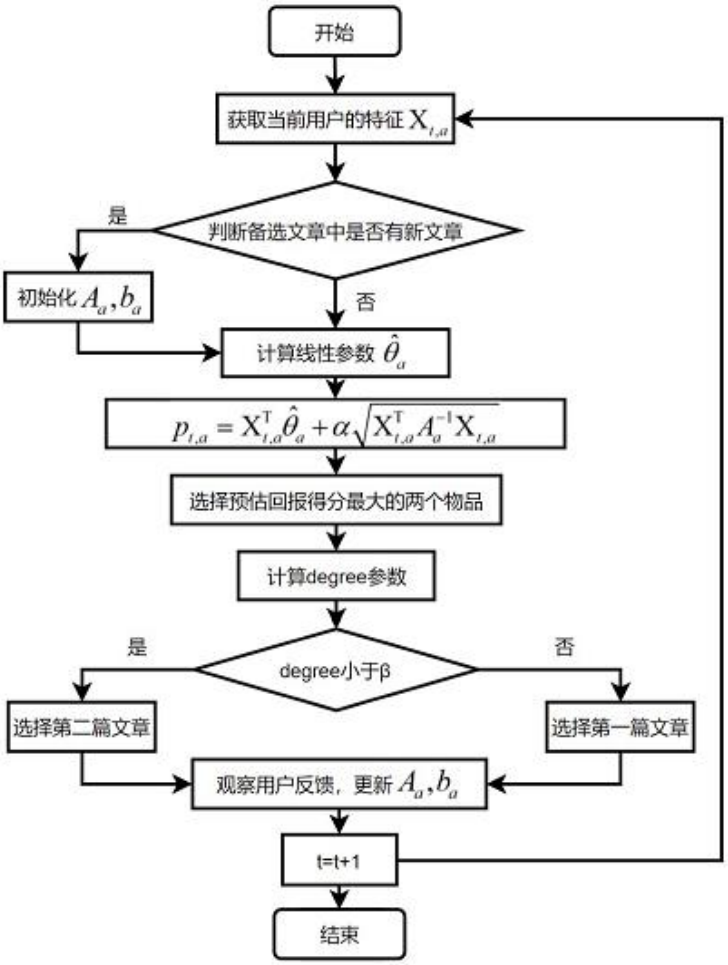

图 4-4 GLinUCB 算法流程图

其中的 β 参数为判断最大置信区间上届算法失效的临界参数，此时由于置信区间上届相差较小，而收益均值差别较大，考虑第二篇文章的期望大于第一篇文章，β 的值通常为负数。值得注意的是 LinUCB 算法基本原理中 $X_{a,t}$ 矩阵应由用户特征向量和文章的特征向量共同决定，此处由于备选推荐文章特征不随时间变化的特点，将用户特征作为 $X_{a,t}$ 处理。

4.3 算法评估

4.3.1 评估方式

本节将对本文使用的评估方式进行介绍。本文主要采用离线评估方式对动态推荐算法进行性能测试。由于 GLinUCB 算法是一个在线的，动态的算法，算法不断根据用户的反馈来优化推荐策略，所以传统的划分训练集和测试集的评估方法不再适用。另外，

由于数据集为一条一条的历史推荐日志，随机推荐策略选择的推荐文章与 GLinUCB 算法选择的推荐文章不一定相同，这种情况下，GLinUCB 算法无法得知用户的真实反馈，此条记录是无效数据，不参与算法的更新以及算法的评估。因此评估的过程中需要对数据进行筛选，匹配有效的数据。其评估算法如下所示：

算法 2 策略评估算法

Algorithm 2 Policy_Evaluator

Algorithm 2: Policy_Evaluator

0: Input: α

1:**for** t=1,2,3….T, **do**

2: Observe the current article chosen by random policy in log: R_t

3: Observe the current article chosen by GLinUCB policy: a_t

4: **if** a_t is same as R_t:

5: $matches \leftarrow matches+1$

6: Observe the current reward r_{t,a_t}:

7: **if** r_{t,a_t} is 1:

8: $clicks \leftarrow clicks+1$

9: **end if**

10: **end if**

11:**end for**

12: $ctr = clicks / matches$

13:Output: ctr

其中α是模型的超参数，在 LinUCB 算法模型下，由文献[4]可知，当α取 0.2 时推荐系统能够取得最佳性能，因此我们评估 GLinUCB 算法性能时也将α设置为 0.2，方便与 LinUCB 算法的比较。评估模型的输出为 CTR（click to rate）指标，定义为点击行为次数与数据匹配次数之比。由于我们设定一次点击行为能够获得 1 的回报；否则回报为 0。通过这种收益的定义，文章的预期收益均值，恰好是点击率，选择具有最大点击率的文章相当于最大化用户的预期点击次数，与我们的老虎机算法制定中的总预期收益最大化是一样的。

在算法的具体实现过程中，我们发现每次实验所得的 CTR 数据都不相同，经过多次调试发现：在推荐系统初次推荐的时候，由于缺乏有效的历史数据，每篇文章的得分相同，都为 0，因此系统初始状态推荐的策略是随机的。直到系统产生第一次有效推荐，即当前推荐文章与数据记录中的真实推荐文章相同时，才会对矩阵 A_a 和 b_a 进行更新，从而打破文章得分都为 0 的处境。由于初始状态模型的随机性会影响最终的 CTR 结果，我们决定在每篇文章得分都相同的情况下固定第一次推荐选择的文章，命名为首选文

章，由此确保每次实验所得结果是具有参考意义，并且是可重现的。

4.3.2 评估结果

本小节使用 4.3.1 小节中介绍的评估方式对算法进行性能测试。随机选取六篇文章作为六次实验的首选文章，通过评估算法对数据筛选，得到约 20 万条有效数据，对六次实验的结果取平均值，得到实验结果如下：

表 4-1 GLinUCB 算法性能

Table 4-1 The performance of GLinUCB

算法	临界参数 β	CTR
随机策略	--	3.98%
LinUCB	--	6.53%
GLinUCB	-0.5	6.46%
	-1	**6.67%**
	-2	6.61%

由上表可知，当 β=-1时，GLinUCB 算法相较原 LinUCB 算法具有较大的性能提高，算法增益为（6.67%-6.53%）/6.53%=2.14%。同时我们也发现，当 β=-0.5 时，相较于原 LinUCB 策略，CTR 反而变低，推荐系统性能反而降低。这是因为当 β 的取值越接近 0，触发 Greedy 策略所需要的均值差占得分差的比例越小，算法越趋向于利用，由此会导致推荐系统性能的下降。相反，β 的取值越远离 0，算法中的 Greedy 策略应用越少，失去对探索和利用力度的调控。

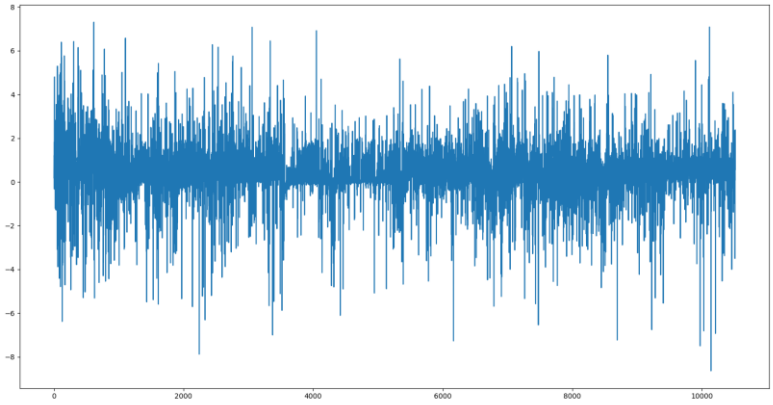

图 4-5 GLinUCB 算法下的 degree 值可视化结果

对 GLinUCB 算法做 4.1.2 小节的处理，共获得 10515 条数据，如上图所示，其中负数数据为 3017 条，占比 28.7%，相较于 4.1.2 小节的结果有明显的减少，说明 GLinUCB

算法能够有效缓解 LinUCB 算法倾向于放弃大的利用利益而选择小的探索利益的问题。

4.4 本章小结

本章首先详细介绍了 LinUCB 算法实现过程中利用和探索的原理，并发现了其利用和探索过程中出现的问题：当置信区间上界相差不大时，LinUCB 忽略了收益均值对性能的影响，损失了推荐系统的利用利益。启发式地引入 Greedy 思想对算法进行了优化，提出了 GLinUCB 算法。该算法的基本思想是：在置信区间上界的差值与收益均值的差值之比满足小于参数 β 的条件时，主要考虑均值对系统性能的贡献。通过多次实验，得到了当前 GLinUCB 算法 β 参数的最佳值为-1，它能够在超参数 α 的基础上，更加精确的权衡推荐策略利用和探索的力度，避免在文章得分相差很小的情况下，损失较大的利用收益。试验结果表明，GLinUCB 算法在原 LinUCB 算法的基础上能够进一步提高推荐的准确率，并获得了 2.14%的增益，

5 分级推荐算法

在第三章中，通过对用户特征 PCA 降维和可视化处理，我们分析了用户的特征向量在二维空间内的分布，发现了用户的特征向量在空间内分布极不均匀，及由此可能带来的问题：当所有用户共用一个推荐器，且用户特征不均匀的情况下，无法拟合获得最佳的参数，从而影响推荐准确率。针对以上问题，我们在本章将介绍一种对用户特征敏感的分级推荐方法。该方法采用两级结构，根据用户的不同的特征而决定采用不同的推荐器。

5.1 基本思路

本节将介绍分级推荐算法建立多模型的基本思路。由于用户特征在空间中集中在几个簇内，若对于这几个不同的簇使用不同的推荐模型，那么可以分别得到其各自的最佳模型参数，各个推荐模型效果达到最佳的同时，也会提高系统整体的推荐性能。

5.1.1 多推荐器的引入

经过对第三章中用户特征分布的观察，我们发现了用户特征集中分为几个簇类，因此我们选择引入多个推荐器，分别适用于不同的用户群体。

每个推荐器使用其对应用户群体的特征来拟合模型参数。也就是说，当系统输入当前用户特征时，系统选择当前用户所属群体对应的推荐器，将用户特征作为此推荐器的输入，此推荐器根据 LinUCB 算法产生收益得分最高的文章进行推荐，由系统与用户的交互结果更新此推荐器的模型参数。在这种情况下，每个推荐器处理的用户特征相对而言分布更加均匀，由每个簇的样本点拟合出来的模型参数也更加合理。针对不挑剔的用户群体，多推荐器的引入，相当于为其量身定做了一个推荐器，这种方法能丰富推荐模型预测的细节，使得模型的预测更具有针对性，并且能够有效的提高推荐系统整体的推荐准确率。

5.1.2 用户类别分类器的引入

经过上节的分析我们通过引入多推荐器解决了用户特征分布不均带来的模型参数非最优问题，但同时也引出了另外一个问题：如何根据用户特征选择其对应的推荐器。本小节我们将引入用户类别分类器，判别当前用户所属群体和推荐器。

用户类别分类器作为各推荐器的上一级，通过判别模块，决定第二级到底采用哪一个推荐器。理想情况下，由于样本数据间的差异，第二级各个推荐器对同一推荐对象的推荐结果会有较大差别，例如，某用户甲属于群体 A，对应于 1 号推荐器，当系统为甲选择 1 号推荐器时，1 号推荐器能够选择推出更符合甲兴趣的新闻文章，若选择其他推荐器，则推荐效果大打折扣。另外，由于 LinUCB 算法本身是一个动态，在线的算法，在本文的推荐场景中，分类器选择第二级推荐器的策略也是沿时间轴在线、动态地向前推进的，因此考虑用户类别分类器仍采用 LinUCB 算法。对于第一级分类器而言，其备选的物品集合不再是新闻文章，而是第二级的多个推荐器集合。

通过引入用户类别分类器，使模型能够有效判定用户所属类别，从而保证推荐器对于用户特征是敏感的。

5.2 算法框架和实现

经过上述介绍，本小节将对算法的框图及实现流程进行详细说明。

5.2.1 模型框架

分级算法为一个两级结构。第一级为一个用户类别分类器，第二级为多个推荐器。我们沿用参考文献[7]的命名，将其分别称为 Master 模型和 Slave 模型。Master 模型和 Slave 模型采用的都是 LinUCB 算法。对 Master 而言，其输入为当前到访用户的特征，备选物品集合为第二级的 Slave 对象。对 Slave 而言，被 Master 选中的 Slave 模型承担当前的推荐任务，其输入为当前到访用户的特征，备选物品集合为当前备选推荐文章集合。因此分级算法的基本框架是一个“主从”式的两级结构，Master 负责甄别用户特征所属的类别，Slave 用于对当前的用户推荐。

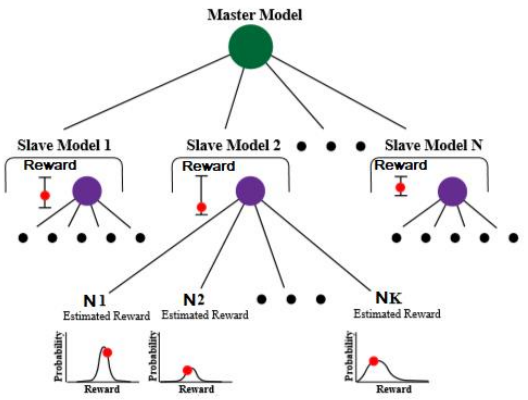

图 5-1 用户特征敏感的分级模型结构

上图中形象地展示了分级算法的结构。图中红色的点代表期望分布的置信区间上界。Master 对用户进行分类也是依据 LinUCB 算法计算得到的收益期望，选择置信区间上限最大的 Slave 进行推荐。

如果将模型简化，只有一个 Master 分类器和两个 Slave 推荐器，即只考虑 Slave1 和 Slave2，推荐场景如上图所示，我们可以观察到当前用户输入到 Master 分类器中，Slave2 的收益期望的置信区间上届大于 Slave1，因此选择 Slave2 对当前用户进行推荐。Slave2 根据 LinUCB 算法计算得到 k 篇备选文章的收益期望的置信区间上界，并选择其中收益期望置信区间上界最大的文章，作为当前推荐文章。

由于最终产生的推荐结果是 Master 选择的结果和 Slave2 从新闻备选文章中选择的结果相互串联得到的，因此，我们应该由用户的反馈同时更新 Master 和选定的 Slave。即使用（推荐的新闻，用户特征，用户反馈）这一个三元组去更新选定的 Slave1，使用（选定的 Slave，用户特征，用户反馈）这一个三元组去更新 Master。这样，第二级的推荐器，第一级的类别选择器都持续得到动态更新，从而不断提升推荐的性能。

5.2.2 算法伪代码

经过以上算法的示例说明，本小节我们给出算法的伪代码：

算法 3 两级的新闻推荐算法

Algorithm 3 A two-level method for news recommendation

Algorithm 3: A two-level method for news recommendation

0: Input: a

1: Initialize **Master** model and **Slave** models

2: **for** t =1,2,3….T, **do**

3: Observe the current user feature $X_t \in R^d$

4: Feed X_t into **Master M** and choice the picked **Slave PS**

5: Feed X_t into **PS** and recommened some particular news a_i

6: Observe the user's feedback r_t

7: Update **M** with (X_t , **PS**, r_t)

8: Update **PS** with (X_t , a_i, r_t)

9: **end for**

该算法对于每个到访的用户，首先将其特征送入 Master 分类器中，Master 分类器由当前用户特征以及历史数据计算得到每个 Slave 推荐器可能带来的收益期望分布，在其中选择收益期望的置信区间上界最大的推荐器。然后将用户特征送入选定的推荐器，

得到最佳推荐新闻文章。接着观察用户对此文章的反馈，并以用户特征、选定的推荐器和反馈更新 Master 分类器，以用户特征、推荐的新闻和反馈更新 Slave 推荐器。

5.3 算法评估

本节我们使用 4.3.1 小节的评估方法对分级推荐算法进行测试，并评价算法的性能表现。

已知 LinUCB 算法中的超参数α的最佳值为 0.2，此处我们将 Slave 推荐器的参数也设置为 0.2，将 Master 分类器的α参数作为分级算法整体的超参数。

表 5-1 分级推荐算法性能

Table 5-1 The performance for level policy

算法	Master 分类器α	CTR
随机策略	--	3.98%
LinUCB	--	6.53%
分级推荐算法（LinUCB）	0.1	6.6%
	0.2	6.66%
	0.3	**6.75%**
分级推荐算法（GLinUCB）	0.3	**6.83%**

由上表可知，Master 分类器在α参数等于 0.3 的情况下能够有很好的表现。相较于 LinUCB 获得了（6.75%-6.53%）/6.53%=3.37%的性能增益。将分级算法中第二级的 Slave 推荐器应用第四章提出的 GLinUCB 算法，取β参数值为-1，同时保持第一级 Master 推荐器为 LinUCB 算法，得到系统的 CTR 性能指标为 6.83%，相较于 LinUCB 单推荐器性能得到了很大的提升。

为了更加直观地了解分级推荐算法中分类器的工作详情，我们随机抽取了 5000 个用户特征样本，并对其进行 PCA 降维处理，得到分类器对用户特征的分类如右图所示结果。其中绿色样本点为 Slave1 推荐器处理的数据，红色样本点为 Slave2 推荐器处理的数据。左图为使用 K-means 聚类方法将样本分为两类的结果。由左右图对比可知 Master 分类器能够实现对用户类别的划分，实现效果近似聚类，但是与聚类效果略有不同。这是由于 Master 分类器不是简单的对用户特征按空间距离进行分类，用户对推荐文章的反

应会作为反馈改变 Master 分类器的模型参数，从而影响分类器的决策。

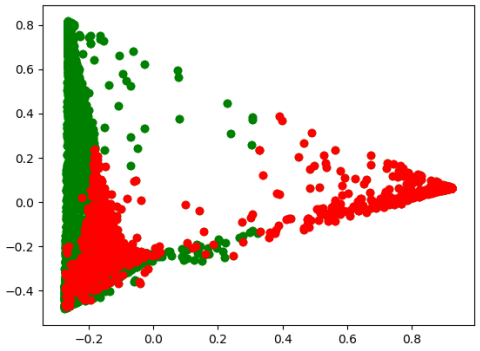

图 5-2 用户特征 K-means 聚类效果

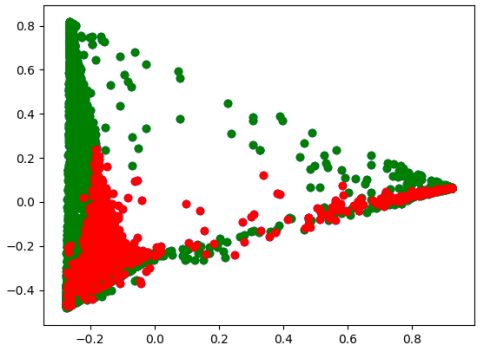

图 5-3 用户特征分类器判别效果

选取首选文章 id 为 109508，观察 LinUCB 和分级推荐算法的 CTR 的变化曲线，得到如下图所示结果：

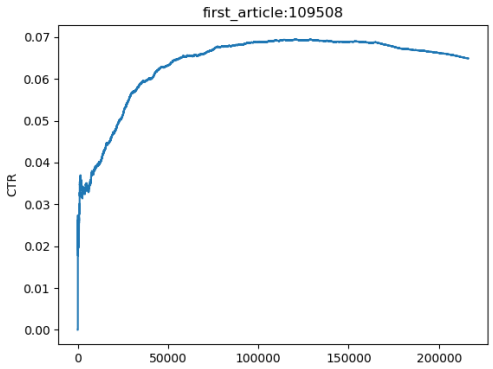

图 5-4 LinUCB 算法 CTR 曲线

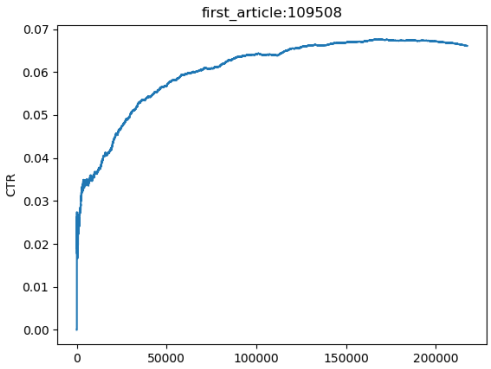

图 5-5 分级推荐算法 CTR 曲线

左图为 LinUCB 的 CTR 曲线，右图为分级推荐算法的 CTR 曲线。对比两图可发现，分级算法下 CTR 曲线的上升速度变缓，处理大约 15 万条数据后算法收敛，CTR 维持在 6.8%附近，而单级 LinUCB 算法在处理了 7.5 万条数据后达到收敛，最终 CTR 保持在 6.6%附近。从最后的结果来看，分级算法的推荐准确率较高，但是从算法的收敛过程来看，分级算法收敛速度要慢于 LinUCB 算法。这是由于分级算法对用户特征进行了分类处理，导致每个 Slave 推荐器能够利用的数据相对于单个推荐器减少了一半，因此 Slave 推荐器数目的增加，将导致算法可利用的数据比例降低，从而导致收敛速度减慢。由此可知，Slave 推荐器的数目并不是越多越好，我们应该针对数据量的大小以及具体的推荐场景合理选择 Slave 推荐器的数目。

5.4 本章小结

本章介绍了一种分级推荐算法，用于解决用户分布不均带来的问题，通过分类器的

判决处理，使得每个子推荐器处理的用户特征达到相对均匀，以此获得更佳的推荐表现。最后我们对分级算法性能进行离线评估，实验结果显示，相对于单个推荐器，多级推荐方式系统性能获得了 3.37%的增益，证明了算法的有效性。我们根据可视化结果，也证明了分类器能够有效判别用户特征。同时，我们根据得到的 CTR 曲线，也发现了分级策略在算法的收敛速度上表现出的不足。

6 总结及展望

6.1 总结

本文以动态的新闻推荐场景为背景，基于 Yahoo 网站提供的大规模在线数据集和 Python 平台，对主流的动态推荐算法 LinUCB 算法进行了深入研究，并针对 LinUCB 算法存在的不足之处对算法进行了两方面的改进和测试。针对 LinUCB 算法在利用与探索过程中可能出现的不合理决策情况，创造性地提出了 GLinUCB 算法；针对单个基于 LinUCB 算法的推荐器没有考虑用户特征在空间中的分布情况的问题，介绍了具有多个子推荐器的分级算法。改进后的两类算法性能表现优异，具有实用价值。具体贡献如下：

1. LinUCB 算法在决定推选哪一篇文章给用户时，仅仅依靠收益期望的最大置信区间上界这一唯一标准，这将导致最大置信区间上界差异较小时出现损失大的利用利益换取小的探索利益的问题。我们通过对数据的统计，验证了以上问题在推荐系统中是普遍存在的。因此我们具有创新性地提出了 GLinUCB 算法，在置信区间上界的差值与收益均值的差值之比满足小于参数 β 的条件时，主要考虑均值对系统性能的贡献，从而使用 Greedy 策略对 LinUCB 算法的推荐策略进行补充。通过离线评估方法对模型进行测试，找到了 GLinUCB 模型的最佳超参数 β。实验结果表明，GLinUCB 算法能够有效解决最大置信区间上界差异较小时出现舍弃损失大的利用利益换取小的探索利益的问题。GLinUCB 算法对推荐系统的推荐准确率有一个很大的提升，其 CTR 指标相对于 LinUCB 算法获得了 2.14%的增益。
2. 由于用户特征分布不均，使用一个推荐器处理全部用户的特征将无法使推荐效果达到最优，我们介绍了一种对用户特征敏感的多推荐器分级结构。分类器先对用户特征进行判别，而后选择子推荐器进行推荐，从而使每个子推荐器处理的是分布相对均匀的用户特征。通过离线评估方式对分级模型进行测试，成功地找到了分类器的最佳超参数，实验结果表明分类器很好的实现了判别功能，分级模型也对推荐系统的性能有很大的提高。其 CTR 指标相对于单个 LinUCB 推荐器获得了 3.37%的增益。

6.2 未来工作展望

LinUCB 作为一种经典的 Contextual-Bandit 算法，能够有效解决冷启动以及用户兴

趣动态变化的问题，目前在动态推荐领域取得了广泛应用。然而，经过我们的研究发现，LinUCB 这一类算法性能对于超参数的选择是很敏感的，在实际系统中，需要通过多次试验确定其最优值，这也是这类算法的缺点之一。另外，LinUCB 算法假定上下文特征与收益期望成线性关系，算法的实现是严格依照数学模型的，当这个关系更为复杂时，LinUCB 算法便不再适用，因此 LinUCB 算法的表征能力也有限。LinUCB 模型中特征的构建也会影响到算法的效果，而模型构建往往是一个复杂庞大的工程。

如何使算法能够根据自身的表现调整模型中的参数，从而使推荐结果更智能更准确将是未来研究中的一个重要方向。如何使模型不再局限于数学假设，具有更强的表征能力，也将是值得研究的一个问题。

参考文献

[1] 张志威.个性化推荐算法研究综述.信息与电脑.2018.

[2] 刘辉,郭梦梦,潘伟强.个性化推荐系统综述[J].常州大学学报(自然科学版).2017.

[3] Chris Anderson.The Long Tail: Why the Future of Business is Selling Less of More[J].Jounrnal of product innovation management.2007

[4] Herlocker J L , Konstan J A , Terveen L G , et al. Evaluating collaborative filtering recommender systems[J]. ACM Transactions on Information Systems, 2004, 22(1):5-53.

[5] Lu Z, Dou Z, Lian J, et al. Content-Based Collaborative Filtering for News Topic Recommendation[J]. 2015.

[6] Li L , Chu W , Langford J , et al. A Contextual-Bandit Approach to Personalized News Article Recommendation[J]. 2010.

[7] Cheng S, Wang B L, Mao L H, et al. Multi-armed bandit recommender algorithm with matrix factorization[J]. Journal of Chinese Computer Systems.2017.

[8] Li S . Online Clustering of Contextual Cascading Bandits[J]. 2017.

[9] Wu Q , Iyer N , Wang H . Learning Contextual Bandits in a Non-Stationary Environment[J]. 2018.

[10] 张佃磊. 基于自适应上下文多臂赌博机推荐算法研究[D]. 山东大学，2018.

[11] Goldberg et al. Using collaborative filtering to weave an information tapestry. COMMUN ACM, 1992.

[12] Burke R D . Hybrid Systems for Personalized Recommendations[C]// Intelligent Techniques for Web Personalization, IJCAI 2003 Workshop, ITWP 2003, Acapulco, Mexico, August 11, 2003, Revised Selected Papers. Springer-Verlag, 2003.

[13] Zimdars A , Chickering D M , Meek C . Using Temporal Data for Making Recommendations[J]. Proc.conf.on Uncertainty in Ai, 2013.

[14] Ding Y , Li X . Time weight collaborative filtering[C]// Proceedings of the 2005 ACM CIKM International Conference on Information and Knowledge Management, Bremen, Germany, October 31 - November 5, 2005. ACM, 2005.

[15] Peter Auer M L . Using Confidence Bounds for Exploitation-Exploration Trade-offs[J]. Journal of Machine Learning Research, 2002, 3(3):397-422.

致　　谢

值此本科学位论文完成之际，我首先想要感谢我的导师陈一帅老师。在本论文的写作过程中，陈老师非常耐心的给我提供了大量的建议，在指导的过程中循循善诱，培养了我发现问题和解决问题的能力。此外，陈老师乐于分享机器学习有关教程，并鼓励毕业设计小组的同学们互相学习和讨论，这使我对人工智能方向有了更加深入的了解，也对今后研究生阶段的学习充满信心。陈老师对待学术严谨细致，对待生活乐观豁达，永怀一颗赤子之心，是我今后工作学习的榜样。在此，借用这些朴实的话语，向陈老师表达我的由衷的感谢和深深的敬意。

感谢唐伟康学长耐心解答我的所有问题，在实验研究和论文编写上给予了我很大的帮助和支持。也祝贺学长毕业快乐，前程似锦，永远保持对代码的热爱。

感谢实验室的学长学姐们、毕业设计小组的同学们以及我可爱的舍友们，因为有你们的陪伴，生活充满轻松愉快的氛围。

感谢赵天次同学，在我生病情绪低落的时候鼓励我，让我重拾信心。

最后，我想感谢我的妈妈，是她在我背后一直默默地支持我，照顾我，让我能够顺利完成学业。