

深圳大学

本科毕业论文(设计)

题目: 融合内容信息的单类协同

过滤推荐算法研究

姓名: 徐留成

专业: 计算机科学与技术

学院: 计算机与软件学院

学号: 2012080173

指导教师: 潘微科

职称: 讲师

2016年4月27日

深圳大学本科毕业论文（设计）诚信声明

本人郑重声明：所提交的毕业论文（设计），题目《融合内容信息的单类协同过滤推荐算法研究》是本人在指导教师的指导下，独立进行研究工作所取得的成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式注明。除此之外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。本人完全意识到本声明的法律结果。

毕业论文（设计）作者签名：

日期： 年 月 日

目 录

| | |
|--|----|
| 摘要(关键词)..... | 1 |
| 1 引言 | 2 |
| 1.1 研究背景及意义 | 2 |
| 1.2 本文主要工作 | 2 |
| 1.3 论文组织结构 | 2 |
| 2 推荐系统概述 | 4 |
| 2.1 主要符号表 | 4 |
| 2.2 推荐系统纵览..... | 4 |
| 2.3 典型推荐算法概述 | 4 |
| 2.3.1 基于内容的推荐系统..... | 4 |
| 2.3.2 基于协同过滤的推荐系统..... | 6 |
| 2.3.3 混合式推荐系统 | 7 |
| 2.4 推荐系统评价指标 | 8 |
| 2.4.1 评分预测 | 8 |
| 2.4.2 TopN推荐 | 9 |
| 2.5 本章小结 | 9 |
| 3 预备知识..... | 10 |
| 3.1 Bayesian Personalized Ranking..... | 10 |
| 3.1.1 Pairwise Preference Assumption | 10 |
| 3.1.2 预测公式 | 10 |

| | | |
|-------|--|----|
| 3.1.3 | Likelihood of Pairwise Preference..... | 11 |
| 3.1.4 | 目标函数 | 11 |
| 3.1.5 | 随机梯度 | 12 |
| 3.1.6 | 迭代更新 | 13 |
| 3.1.7 | BPR算法 | 13 |
| 3.1.8 | 收敛缓慢的原因 | 13 |
| 3.2 | Latent Dirichlet Allocation | 14 |
| 3.2.1 | 数学模型 | 14 |
| 3.2.2 | 使用吉布斯采样估计LDA参数 | 15 |
| 3.3 | 本章小结 | 16 |
| 4 | 适应性采样策略 | 17 |
| 4.1 | 适应性采样策略概览..... | 17 |
| 4.2 | 类别分布 | 17 |
| 4.3 | 选取negative item v_j | 18 |
| 4.3.1 | 物品浏览概率..... | 18 |
| 4.3.2 | 如何对物品列表进行排序..... | 18 |
| 4.4 | 适应性采样算法 | 20 |
| 4.5 | 本章小结 | 20 |
| 5 | 融合内容信息的适应性BPR | 21 |
| 5.1 | Learning Content-aware Mappings | 21 |
| 5.2 | Parameter Inference of CA-BPR..... | 21 |
| 5.3 | 本章小结 | 22 |

| | |
|--------------------------|----|
| 6 实验论证..... | 23 |
| 6.1 数据集..... | 23 |
| 6.2 评测标准 | 23 |
| 6.3 实验过程与分析 | 25 |
| 7 结论与展望 | 27 |
| 7.1 本文的主要内容 | 27 |
| 7.2 进一步的研究工作 | 27 |
| 参考文献 | 28 |
| 致谢..... | 33 |
| Abstract(Key words)..... | 34 |

融合内容信息的单类协同过滤推荐算法研究

计算机与软件学院计算机科学与技术专业 徐留成

学号：2012080173

【摘要】 对于基于隐式反馈的个性化推荐算法而言，pairwise learning是一个非常重要的技术手段。pairwise learning 算法通常基于这样一个假设：对于一个用户而言，相比于未选择过的物品往往会更感兴趣于已选择过的物品。这种假设在推荐算法的学习过程中会衍生出大量的training pairs。而为了应对大规模的数据集，我们所研究的推荐算法往往都是基于均匀采样的随机梯度下降方法进行求解。不过，这种采取均匀采样的策略经常会导致算法收敛非常缓慢。在本文中首先讨论了均匀采样策略导致收敛缓慢的原因，并研究了通过在已有的BPR推荐框架中融合内容信息改进采样策略并最终提高推荐效果的方法。实验证明，相比于均匀采样策略，通过融合内容信息的适应性采样策略的确能够有助于提高推荐效果。

【关键词】 推荐系统; 协同过滤; 适应性采样

1 引言

1.1 研究背景及意义

互联网的出现和普及给用户带来了大量的信息,满足了用户在信息时代对各种信息的需求,但随着Internet的迅速发展而带来的网络上信息量的巨幅增长,使得用户在面对大量信息时无法快速从中获得对自己真正有用的那部分信息。换言之,在这种情况下人们对信息的使用效率反而降低了,这就是所谓的信息过载(information overload)问题。的确如此,面对信息的汪洋大海,人们往往感到无所适从,信息过载已经成为一个不容忽视的问题。

目前,应对信息过载的办法之一便是以搜索引擎为代表的信息检索系统,比如国外的Google¹、国内的Baidu²等,它们在帮助用户从巨大的网络资源中获取信息方面发挥着极其重要的作用。但对于使用搜索引擎的用户而言,在使用同一个关键字搜索信息时,在一段时间内所得到的结果都是相同的。另一方面来看,信息及其传播是多样化的,而用户对信息的需求是多元化和个性化的,那么通过以搜索引擎为代表的信息检索系统获得的结果显然不能满足用户的个性化需求,它们仍然无法很好地解决信息过载问题。

面对信息过载,另外一个非常有潜力的办法是个性化的推荐系统,它是根据用户的信息需求、兴趣等,将用户所感兴趣的信息、产品、服务等推荐给用户的个性化信息推荐系统。和搜索引擎相比,推荐系统通过研究用户的历史行为与兴趣偏好,进行个性化考量,由系统发现用户的兴趣点,从而引导用户发现自己的信息需求。一个优秀的推荐系统不仅能为用户提供个性化的服务,还能和用户之间建立密切关系,让用户对其推荐产生依赖。个性化推荐系统现已广泛应用于很多领域,其中最典型并具有良好的发展和应用前景的领域就是电子商务领域。目前,几乎所有大型的电子商务系统,如Amazon, eBay, 京东, 当当网上书店等,都不同程度地使用了各种形式的推荐系统。同时学术界对推荐系统的研究热度一直很高,逐步形成了一个独立的研究领域。

Internet为人们提供了极其丰富的信息资源,在这些海量、异构的Web信息资源中蕴含着具有巨大潜在价值的知识。根据用户访问的历史记录以及各种服务或商品之间的相关信息可以构建用户的兴趣模型,从而凭借该用户的兴趣模型对繁杂的信息进行过滤,然后向用户推荐其可能感兴趣的服务或商品。事实上,推荐系统已经成为目前解决信息过载最有效的工具之一。

1.2 本文主要工作

本文从推荐系统的概述展开,讨论了在推荐系统的学习算法中随机梯度下降方式中采用均匀采样策略而导致收敛缓慢的一些原因,并通过融合内容信息改进了均匀采样策略-适应性采样策略,然后将适应性采样策略放入已有的推荐算法框架中,加快原有推荐算法的学习。

1.3 论文组织结构

本论文共分为七章,内容如下:

第一章为引言,主要介绍了本论文的研究背景、意义,主要工作及论文的组织结构。

¹<https://www.google.com/>

²<https://www.baidu.com/>

第二章为推荐系统概览,并分类介绍了包括了基于内容、基于系统过滤与混合型推荐算法的一些典型的推荐学习算法。

第三章为预备工作,首先简要回顾了Bayesian Personalized Ranking(BPR)推荐算法,并对其局限性进行了一些探讨。

第四章为适应性采样策略,主要研究了通过融合内容信息提出了适应性采样策略改进已有的均匀采样策略。

第五章为整体的算法框架,将适应性采样策略融入已有的BPR推荐模型。

第六章为实验论证,主要内容为在适应性采样策略下的推荐算法的实验表现。

第七章为结论与展望,首先简要总结了本文的一些工作,并对接下来进一步的研究工作做了展望。

2 推荐系统概述

2.1 主要符号表

表1中列举了大部分在本文中使用的符号及其意义。

2.2 推荐系统纵览

自从20世纪90年代中期第一篇关于协同过滤(Collaborative Filtering)的研究文章^[40]出现以后,推荐系统就开始成为了一个重要且有趣的研究主题。协同过滤通过收集推荐系统中相似用户的偏好进行推荐,而生成近邻用户(neighbourhood formation)是协同过滤中非常重要的一个方面^[20;24]。近邻用户生成的目的是为每个用户找到一些相似的用户群或其最近邻,然后基于有着相似偏好的近邻用户推荐产品或服务^[9;50]。这里的近邻(neighbourhood)是指那些对于我们将要为之提供推荐建议的用户所感兴趣的物品有过相似交互行为的其他用户。在这里,我们把需要为之提供推荐的用户成为目标用户,那么通过比较目标用户与其近邻评分,就可以做出最终的推荐^[1;21]。当缺乏用户评分数数据的时候,协同过滤就会遇到所谓的稀疏性问题,这将导致推荐效果变得很差。因此,在推荐系统中预防稀疏性问题非常重要。为此一个很重要的途径便是从隐式反馈(比如用户的购买行为,上线时间,历史浏览记录)数据中提取用户的偏好信息来降低协同过滤对于用户评分数据的依赖,同时提高推荐效果^[4;18]。隐式反馈数据能够通过对于用户行为的观测提供更多的信息来降低评分数据不充分的影响^[35;50]。另一方面,协同过滤推荐技术的用户画像(user profile)通过用户对于物品的评分得以构建。为了降低协同过滤对于评分数据的依赖,用户行为(user activity)也已经成为研究调查的一个重要关注点,也就是说通过挖掘用户偏好的经验性知识来构建更加精确的用户画像(user profile)^[9;21;23;50]。

2.3 典型推荐算法概述

推荐系统通过识别用户的需求与偏好为其推荐合适的产品或服务。目前国内外关于推荐系统的研究下已衍生了很多推荐算法,这些推荐算法通常可以分为三类:基于内容的推荐(Content-based recommendations),协同过滤(Collaborative Filtering)和混合型(Hybrid approaches)推荐。

2.3.1 基于内容的推荐系统

基于内容信息的方法^[11;17;37]来学习个体的隐式表达(latent representation)并缓解冷启动(cold start)问题。比如,在FM^[37]中各种属性信息被放到特征矩阵中,然后通过对于评分数据回归分析相关属性。

基于内容的推荐系统从用户与物品的content profile之间的相似度出发进行推荐。他们从研究推荐系统中个体的内容信息角度进行分析。通常这类方法利用个体的内容信息,比如物品属性,用户文本,或照片的像素点,主要利用探索启发式(heuristics)的方法。在^[5;22;31]中,他们使用诸如cosine similarity的方式来衡量相似度,然后推荐在内容上与用户过去所喜欢的相类似的物品。在^[33]中,基于物品内容信息并由用户标注的标签:“相关(relevant)”或者是“不相关(irrelevant)”,作者学习了一个贝叶斯分类器来对没有标注的物品进行分类。近来,也有很多社交媒体(social media)相关的推

表 1: 主要符号表

| 常用符号 | 意义 |
|---|---|
| s | user number |
| t | item number |
| u | user |
| v | item |
| u_m | the specified user m |
| v_i | the specified item i |
| v_j | the specified item j |
| b_i | item bias |
| r_{ui} | real rating of user u on item i |
| \hat{r}_{ui} | predicted rating of user u on item i |
| \hat{r}_{uj} | predicted rating of user u on item j |
| e_i | entity, e.g., user u or item v |
| T | iteration number in the algorithm |
| $k \in \mathbb{R}$ | number of latent dimensions |
| $r(j)$ | the ranking place of the item v_j |
| \mathcal{P} | (user, item) pairs in training data |
| \mathcal{P}^{te} | (user item) pairs in test data |
| \mathcal{U} | the whole user set |
| \mathcal{I} | the whole item set |
| \mathcal{I}_u^{re} | recommended items for user u |
| \mathcal{I}_u^{te} | selected items by user u in test data |
| \mathcal{I}_u^{tr} | selected items by user u in training data |
| $\mathcal{I}_{u_m}^+$ | the set of items selected by the user u_m |
| $U \in \mathbb{R}^{s \times k}$ | user-specific latent matrix |
| $V \in \mathbb{R}^{t \times k}$ | item-specific latent matrix |
| $U_{u.} \in \mathbb{R}^{1 \times k}$ | user-specific latent feature vector |
| $V_{.v} \in \mathbb{R}^{1 \times k}$ | item-specific latent feature vector |
| $Y^e = [y_1^e, y_2^e, y_3^e, \dots]$ | latent representation of entities |
| $y_i^e \in \mathbb{R}^{1 \times k}$ | the latent vector of entity e_i |
| $\mathcal{C} = \{c_1, c_2, \dots, c_k\}$ | categories |
| $D_S := \{(m, i, j) v_i \in \mathcal{I}_{u_m}^+ \wedge v_j \in \mathcal{I} \setminus \mathcal{I}_{u_m}^+\}$ | the set of all pairwise preference |

荐系统关注content-based推荐方法并对其进行了很多研究。比如,在^[25;28]中通过基于可视性的内容相似度考虑它的最近邻标签,然后来为目标图像推荐标签。^[30]提出了一个在线视频的推荐系统,而该系统则利用了用户在用户与视频间点击数据的多模态的内容关联度。

但是,这些基于内容的推荐方法大都具有以下局限性:第一,它们必须有足够的信息构建一个分类器,并且显然会被推荐物品的特征所局限;第二,它们推荐的物品,在内容上往往与用户已经有过评分行为的物品很相似,显然这就会导致较低的推荐多样性。

2.3.2 基于协同过滤的推荐系统

协同过滤(Collaborative Filtering)方法通过挖掘用户的评分历史来预测用户的偏好。它们并不需要内容信息(content information),并且能够发现一些基于内容的推荐方法所不能发现的一些有趣的联系。通常来说,协同过滤基于这样一个基本的设想:相似的用户对于相似的物品有着相似的行为^[3;43]。这里的“相似”并不同于content-based方法中的内容相似度(content similarity),它指的是相似的评分偏好(similar rating preference)。

协同过滤方法可大致分为两类:memory-based methods, model-based methods。memory-based方法^[7;16;27;41]通常通过搜寻相似的用户或商品去进行推荐。而其相似度则是经由评分历史计算而得。memory-based方法也可进一步的被分为user-based和item-based两类方法。通过与当前用户有着相似偏好的其他用户进行推荐即为user-based,通过推荐与当前用户喜欢过的物品所相似的物品即为item-based。不过,当缺乏用户评分数据的时候,协同过滤就会遇到叫做稀疏性的一个问题,这将很容易导致推荐效果变得很差。因此,在推荐系统常常需要应对稀疏性这一大难题。应对稀疏性问题一个重要的途径便是从隐式反馈(implicit feedback)(比如用户的购买行为,上线时间,历史浏览记录)数据中提取用户的偏好信息来降低协同过滤对于用户评分数据的依赖,当然这往往同时也能够提高推荐效果^[4;18]。另外,相对于显式反馈,隐式反馈的数据更易采得也更丰富。隐式反馈能够通过对于用户行为的观测提供更多的信息来降低评分数据不充分的影响^[35;50]。这时其实也就是变成我们所谓的单类协同过滤(One-class Collaborative Filtering)问题。

OCCF问题的最典型特征是仅能够观测到正向采样(positive examples),比如用户的点击行为,浏览行为,同时数据分类往往非常不均衡,比如用户点击过物品可能只是占到整个物品集合的很小一部分。我们把用户未有过交互行为的物品,比如未点击过的物品,叫做negative examples。那么如何从大量未有过交互行为的物品集合中针对negative examples进行采样与建模是很多问题的关键所在。在前人的一些工作中,有几种直观的策略来处理这个问题。其实一个最常见的做法是将所有缺失的数据视作negative examples,显然这将导致推荐结果具有偏差,因为很多缺失数据很多可能是positive examples。另一种做法是所缺失的数据是做未知的,这将导致协同过滤模型仅利用了positive examples。近来的一些研究中,一些关于OCCF的研究人员将重点放到了对于negative examples的建模上^[19,34,35,44]。他们的一个基本的想法是将所缺失的数据视作是negative,但是给出了将其视作negative的一个概率权重。不过,他们当中的部分做法仅仅是通过简单地观测历史反馈的概率属性来区分negative examples。比如,^[19,34]他们计算了每个用户给多少物品评过,每个物品被多少用户评过,由此来计算一个权重。进一步的说,他们认为如果一个用户浏览过的物品越多,那么他没有浏览过的物品便更大可能是negative类型;如果一个物品被越少的用户浏览过,那么这个物品相关缺失数据便更小可能是negative,这种做法仍然是略显粗糙。

协作型方法^[39;48;51]通过处理大量的用户与物品间的交互信息,比如隐式反馈和显式的评分(也

叫作协同信息)。这些方法不同于memory-based方法, model-based方法采用机器学习与概率统计的技术从已有的用户评分去学习一个模型, 再将模型应用到推荐中。其中包括有隐语义模型(latent semantic models), 图模型(graphical models), 贝叶斯模型(Bayesian models), 聚类模型(clustering models)。在众多的model-based方法中, 低秩矩阵分解(low-rank Matrix Factorization)由于在可扩展性与精确度方面的优势已经获得了许多研究者的关注。其实分解的方法在个性化的推荐系统中很常见。他们可以被用来处理推荐系统中收集的各种信息, 比如隐式反馈^[19;39], 物品属性^[11;37], 用户画像^[17]和社交信息^[29]。其中矩阵分解基于用户的偏好可以被一小部分因子表示, 通过从user-item rating matrix来学习user与item一个低秩隐含因子, 然后利用它们去预测未被观测到的ratings。

矩阵分解^[13]及其一些扩展方法^[12;26;49]是用来处理协同信息的非常典型的分解方法, 它通过分解协同信息并试图在一个共享的隐式空间学习用户与物品的隐式表达。比如, 隐式矩阵分解^[19]通过为每个user-item pair计算一个适应性的信任权重来扩展基础的BPR处理隐式反馈。尽管通过扩展BPR能够应对隐式反馈问题, 但是由于在隐式反馈数据集中普遍存在的数据倾斜(data skew)问题(正反馈数量常常不到总数的1%), 他们很容易陷入过拟合问题。为了缓解数据倾斜与推荐系统的隐式反馈学习, Bayesian Personalized Ranking (BPR)^[39]和它的一些扩展方法^[32;34;38]被提出, 其所基于的假设为: 相比于未选择的物品用户更感兴趣已经选择的物品。这样假设会产生大量的训练数据, 因此对应的学习算法通常基于均匀采样用户物品对的随机梯度下降。但是不同的训练采样可能会对参数学习产生不同的影响, 均匀采样策略往往会产生大量低效的训练采样并导致收敛变得缓慢。尤其是当物品数量很大和物品的流行度有着长尾分布(long distribution)^[10]的时候, 均匀采样策略将会导致极其缓慢的收敛。因此, BPR的作者Rendle进一步研究了长尾效应并利用它提出了非均匀的物品采样器^[38]。对于给定的一个用户, 他们计划挑选出那些在某一领域很流行并且尚未被该用户选择过的物品来构成训练对。理论上, 这种采样方式很耗时, 因为它将物品的隐式因子当做物品流行度的指示器并且需要在每轮迭代的每个区域对物品进行重新排序。为了考虑运行效率, Rendle不得不减少重新排序的时间来妥协推荐性能。另一方面, 为了获得一个通用的加速BPR学习的方案,^[51] 尝试根据一个在两个不同未选择过的物品上的偏好差别来选取那些富含信息的训练对。但是, 由于真实世界的数据集里物品数往往极其庞大, 这种策略不得不在计算偏好差别上花费大量的时间。因此,^[38;51] 都陷入了平衡算法效率与性能表现的两难境地。在本课题中所研究的采样策略在效率与性能两方面都表现了很好的效果, 并且有潜力加速BPR的学习。

传统的协同过滤对于评分预测问题往往能够取得很好的效果, 比如Netflix的电影推荐。但是, 它受制于一个众所周知的问题: 冷启动, 当一个新的物品或用户进入系统时由于几乎无法获得任何评分记录, 在此种情况下推荐效果往往很不理想。为了缓解推荐系统中的冷启动问题, Map-BPR^[11]扩展了BPR框架, 他们学习了一个将内容信息空间映射到隐式空间的一个映射关系。然后, Map-BPR利用学习到了这个映射学习那些缺乏协同信息的新个体的隐式因子。不过, Map-BPR将隐式因子的学习分割为两个不相关的部分。这会导致在隐式反馈数据集中的个体的隐式因子仅仅指示协同属性而不会显示内容属性。为了获得更可信的隐式因子, 在本课题的研究方法在同一个学习过程中研究了通过协同信息与内容信息学习个体的隐式因子。

2.3.3 混合式推荐系统

混合方法尝试将基于内容与协同过滤的推荐方法结合起来应对它们的局限性。^[8]通过将基于内容与协同过滤的预测结果进行线性组合设计了一个混合推荐模型。^[42]提出从概率混合的角度将协同过滤与基于内容的推荐方法进行统一。近来也有很多工作都重点关注了社交媒体推荐(social me-

dia recommendation), 而他们中的大部分都采用了混合方法, 在挖掘社交媒体内容的同时考虑了用户的历史行为来获得更高的推荐准确度。^[47]为在线社交网络中的视频推荐(video recommendation)设计了一个组合式的社交内容推荐框架。他们的方法通过利用社交网络信息(social network information)与内容信息(content information), 提出一个user-content matrix填充冷启动中的user-video条目。^[44]研究利用了集成学习(ensemble learning)方法, 在音乐推荐中将基于物品协同过滤结果与基于内容方法的结果进行融合。

2.4 推荐系统评价指标

所谓评价指标主要包括“技术评价指标”和“业务评价指标”。技术评价指标包括诸如RMSE³、MAE⁴、NDCG⁵、MAP⁶、Recall、Precision等, 业务评价指标如成交转化率、用户点击率等。^[52]也介绍了推荐系统中的很多评测指标。这些评测指标可用于评价推荐系统各方面的性能, 它们包括用户满意度、预测准确度、覆盖率、多样性、实时性、健壮性等等。其中有些可以通过计算来定量衡量, 有些则只能定性描述, 有些可以通过离线实验计算, 有些需要通过用户调查获得, 还有些只能在线评测。这里主要介绍在技术评价指标中, 评分预测与TopN推荐的预测准确度定义。

2.4.1 评分预测

房间 Room (2015)



导演: 伦尼·阿伯拉罕森
 编剧: 艾玛·多诺霍
 主演: 布丽·拉尔森 / 雅各布·特伦布莱 / 琼·艾伦 / 肖恩·布里吉格斯 / 威廉姆·H·梅西 / 更多...
 类型: 剧情 / 家庭
 官方网站: roomthemovie.com
 制片国家/地区: 爱尔兰 / 加拿大
 语言: 英语
 上映日期: 2015-09-04(特柳赖德电影节) / 2016-01-15(爱尔兰)
 片长: 118分钟
 又名: 不存在的房间(台) / 抖室(港)
 IMDb链接: tt3170832

豆瓣评分



好于 96% 剧情片
 好于 96% 家庭片

图 1: 用户评分

很多提供推荐服务的网站都有一个让用户给物品打分的功能。那么, 如果知道了用户对物品的历史评分, 就可以从中习得用户的兴趣模型, 并预测该用户在将来看到一个他没有评过分的物品时, 会给这个物品评多少分。预测用户对物品评分的行为称为评分预测。

评分预测的预测准确度一般通过RMSE和MAE计算。对于测试集中的一个用户 u 和物品 i , 令 r_{ui} 是用户 u 对物品 i 的实际评分, 而 \hat{r}_{ui} 是推荐算法给出的预测评分, 那么RMSE的定义为:

$$RMSE = \frac{\sqrt{\sum_{(u,i) \in \mathcal{P}^{te}} (r_{ui} - \hat{r}_{ui})^2}}{|\mathcal{P}^{te}|}$$

³RMSE: Root Mean Squared Error, 均方根误差

⁴MAE: Mean Absolute Error, 平均绝对误差

⁵NDCG: Normalized Discounted Cumulative Gain

⁶MAP: Mean Average Precision, 平均准确率

MAE 采用绝对值计算预测误差，它的定义为：

$$MAE = \frac{\sum_{(u,i) \in \mathcal{P}^{te}} |r_{ui} - \hat{r}_{ui}|}{|\mathcal{P}^{te}|}$$

关于RMSE 和MAE 这两个指标的优缺点，Netflix 认为RMSE 加大了对预测不准的用户物品评分的惩罚(平方项的惩罚)，因而对系统的评测更加苛刻。研究表明，如果评分系统是基于整数建立的（即用户给的评分都是整数），那么对预测结果取整会降低MAE 的误差。

2.4.2 TopN推荐

猜你喜欢



图 2: TopN推荐

网站在提供推荐服务时，一般是给用户一个个性化的推荐列表，例如购物网站上的热门推荐，这种推荐叫做TopN 推荐。在现实场景下，TopN推荐也是更常见的一种推荐形式。

TopN 推荐的预测准确率一般通过准确率(precision)/召回率(recall)衡量。对于用户 u ，推荐列表 \mathcal{I}_u^{re} 的准确率定义为：

$$Precision_u = \frac{|\mathcal{I}_u^{re} \cap \mathcal{I}_u^{te}|}{|\mathcal{I}_u^{re}|}$$

其召回率定义为：

$$Recall_u = \frac{|\mathcal{I}_u^{re} \cap \mathcal{I}_u^{te}|}{|\mathcal{I}_u^{te}|}$$

2.5 本章小结

本章首先对推荐系统进行了概括性的介绍，然后主要从典型推荐算法与推荐系统的评价指标两方面对推荐系统的整个框架形成了一个粗略的认识。

3 预备知识

3.1 Bayesian Personalized Ranking

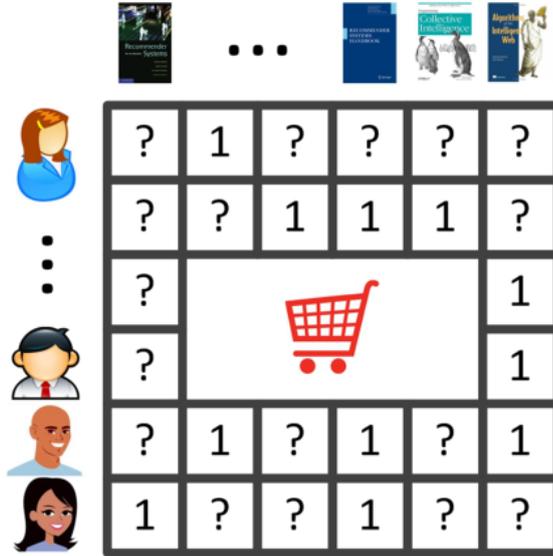


图 3: user-item 隐式反馈矩阵

在这一节，我们首先回顾BPR算法，然后讨论它的一些局限性，也就是其收敛缓慢与冷启动问题。通常用户与物品的隐式反馈可以表示为如图3所示的矩阵，矩阵的“1”表示用户已经对该物品有过交互行为，比如购买、点击等，矩阵的“?”则表示用户还未对该物品有过交互行为。

3.1.1 Pairwise Preference Assumption

BPR^[39]是一个应对隐式反馈很流行的推荐框架。它基于这样一个偏好假设：如果一个用户 u 已经选择了物品 i 但是没有选择物品 j ，那么在BPR中，我们认为相对于物品 j 用户 u 更喜欢物品 i ，并定义用户 u 关于物品 i 与 j 的偏好关系为：

$$p(i \succ_u j) := f(x_{uij}), \tag{1}$$

这里 $f(x) = 1/(1 + \exp(-x))$ ⁷， $x_{uij} := s(u, i) - s(u, j)$ ， $s(\cdot, \cdot)$ 可以是任何表示用户与物品相关程度的函数。在BPR^[39]中， $s(\cdot, \cdot)$ 为用户对物品的预测值，即 $s(u, i) = \hat{r}_{ui}$ ， $x_{uij} = \hat{r}_{ui} - \hat{r}_{uj}$ 。

3.1.2 预测公式

在BPR中，用户 u 对于物品 i 的预测值 \hat{r}_{ui} 公式为：

$$\hat{r}_{ui} = U_u \cdot V_i^T + b_i \tag{2}$$

⁷ $f(x)$ 即为sigmoid函数

3.1.3 Likelihood of Pairwise Preference

伯努利分布(Bernouli distribution)是关于布尔变量 $x \in \{0, 1\}$ 的概率分布, 其连续参数 $p \in [0, 1]$ 的概率.

$$(x|p) = Ber(x|p) = p^x (1-p)^{1-x} \quad (3)$$

若记事件 $(\hat{r}_{ui} > \hat{r}_{uj})$ 的概率为 $p(\hat{r}_{ui} > \hat{r}_{uj})$, 布尔变量 $\delta((u, i) \succ (u, j))$ 服从伯努利分布, 那么用户 u 的likelihood of pairwise preference 在^[39]中被定义为:

$$\begin{aligned} LPP_u &= \prod_{i, j \in \mathcal{I}} p(\hat{r}_{ui} > \hat{r}_{uj})^{\delta((u, i) \succ (u, j))} [1 - p(\hat{r}_{ui} > \hat{r}_{uj})]^{1 - \delta((u, i) \succ (u, j))} \\ &= \prod_{(u, i) \succ (u, j)} p(\hat{r}_{ui} > \hat{r}_{uj}) \prod_{(u, i) \preceq (u, j)} [1 - p(\hat{r}_{ui} > \hat{r}_{uj})] \end{aligned} \quad (4)$$

这里的 $(u, i) \succ (u, j)$ 表示用户 u 相比物品 i 更喜欢物品 j .

用 $f(\hat{r}_{uij})$ 来近似表示概率 $p(\hat{r}_{ui} > \hat{r}_{uj})$ ^[39], 对于公式4取其对数即 $\ln LPP_u$, 那么就有:

$$\begin{aligned} \ln LPP_u &= \ln \prod_{(u, i) \succ (u, j)} f(\hat{r}_{uij}) + \ln \prod_{(u, i) \preceq (u, j)} [1 - f(\hat{r}_{uij})] \\ &= \ln \prod_{(u, i) \succ (u, j)} f(\hat{r}_{uij}) + \ln \prod_{(u, i) \succ (u, j)} [1 - (1 - f(\hat{r}_{uij}))] \\ &= \ln \prod_{(u, i) \succ (u, j)} f(\hat{r}_{uij}) + \ln \prod_{(u, i) \succ (u, j)} f(\hat{r}_{uij}) \\ &= 2 \ln \prod_{(u, i) \succ (u, j)} f(\hat{r}_{uij}) \\ &= 2 \sum_{i \in \mathcal{I}_u^{tr}} \sum_{j \in \mathcal{I} \setminus \mathcal{I}_u^{tr}} \ln f(\hat{r}_{uij}) \end{aligned} \quad (5)$$

在这里 $\hat{r}_{uij} = \hat{r}_{ui} - \hat{r}_{uj}$, $f(x) = 1/(1 + \exp(-x))$, .

3.1.4 目标函数

基于上面的成对偏好假设, 可以从隐式反馈数据集中得到所有的偏好集合 $D_S := \{(u, i, j) | v_i \in I_u^+ \wedge v_j \in I \setminus I_u^+\}$, I_m^+ 表示被用户 u 选择过的物品集合, 三元组 (u, i, j) 表示用户 u 选择过物品 v_i 但是没有选择过物品 v_j . 我们把 v_i 叫做一个positive item, v_j 叫做一个negative item. 对于给定的集合 D_S , BPR的目标便是最大化所有user-item pair的似然偏好:

$$\arg \max_{\Theta} \prod_{(u, i, j) \in D_S} p(i \succ_u j), \quad (6)$$

公式(6)等价于最小化负的对数似然函数:

$$L_{feedback} = - \sum_{(u, i, j) \in D_S} \ln f(x_{uij}) + \lambda \|\Theta\|^2, \quad (7)$$

这里的 $x_{uij} = \hat{r}_{uij}$, Θ 表示算法中需要学习的模型参数集合, λ 表示超参数集合。在实际的算法学习中, BPR的学习算法经常采用均匀采样的随机梯度下降(Stochastic Gradient Descent)进行迭代学习。

更为具体的,公式(7)也就是最小化下面的目标函数(Objective Function):

$$\min_{\Theta} \sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{I}_u} \sum_{j \in \mathcal{I} \setminus \mathcal{I}_u} \Phi_{uij} \quad (8)$$

这里的 $\Phi_{uij} = -\ln f(\hat{r}_{uij}) + \frac{\alpha_u}{2} \|U_u\|^2 + \frac{\alpha_v}{2} \|V_i\|^2 + \frac{\alpha_v}{2} \|V_j\|^2 + \frac{\beta_v}{2} \|b_i\|^2 + \frac{\beta_v}{2} \|b_j\|^2$, $\Theta = \{U_u, V_i, b_i\}$ 的将要学习的参数集合。

3.1.5 随机梯度

对于一个随机采样而得的三元组 (u, i, j) , 对目标函数中的参数求其偏导即可得梯度。

在此之前先做一些准备工作, 对于函数 $f(x) = 1/(1 + e^{-x})$ 的导数:

$$f'(x) = -\frac{1}{(1 + e^{-x})^2} e^{-x} (-1) = \frac{e^{-x}}{(1 + e^{-x})^2} = \frac{1}{(1 + e^x)(1 + e^{-x})} = f(x)f(-x)$$

下面开始对参数 U_u 求其偏导:

$$\begin{aligned} \nabla U_u &= \frac{\partial \Phi_{uij}}{\partial U_u} = -\frac{\partial \ln f(\hat{r}_{uij})}{\partial f(\hat{r}_{uij})} \frac{\partial f(\hat{r}_{uij})}{\partial \hat{r}_{uij}} \frac{\partial \hat{r}_{uij}}{\partial U_u} + \alpha_u U_u \\ &= -\frac{1}{f(\hat{r}_{uij})} \frac{\partial f(\hat{r}_{uij})}{\partial \hat{r}_{uij}} \frac{\partial \hat{r}_{uij}}{\partial U_u} + \alpha_u U_u \\ &= -\frac{1}{f(\hat{r}_{uij})} f(\hat{r}_{uij}) f(-\hat{r}_{uij}) \frac{\partial f(\hat{r}_{ui} - \hat{r}_{uj})}{\partial U_u} + \alpha_u U_u \\ &= -f(-\hat{r}_{uij}) \frac{\partial f[(U_u \cdot V_i^T + b_i) - (U_u \cdot V_j^T + b_j)]}{\partial U_u} + \alpha_u U_u \\ &= -f(-\hat{r}_{uij}) (V_i - V_j) + \alpha_u U_u \end{aligned} \quad (9)$$

同样其他参数随机梯度如下:

$$\nabla V_i = \frac{\partial \Phi_{uij}}{\partial V_i} = -f(-\hat{r}_{uij}) U_u + \alpha_v V_i \quad (10)$$

$$\nabla V_j = \frac{\partial \Phi_{uij}}{\partial V_j} = -f(-\hat{r}_{uij}) (-U_u) + \alpha_v V_j \quad (11)$$

$$\nabla b_i = \frac{\partial \Phi_{uij}}{\partial b_i} = -f(-\hat{r}_{uij}) + \beta_v b_i \quad (12)$$

$$\nabla b_j = \frac{\partial \Phi_{uij}}{\partial b_j} = -f(-\hat{r}_{uij}) (-1) + \beta_v b_j \quad (13)$$

3.1.6 迭代更新

对于三元组 (u, i, j) 在采用SGD的BPR算法中的更新公式如下:

$$U_{u.} = U_{u.} - \gamma \nabla U_{u.} \quad (14)$$

$$V_i = V_i - \gamma \nabla V_i. \quad (15)$$

$$V_j = V_j - \gamma \nabla V_j. \quad (16)$$

$$b_i = b_i - \gamma \nabla b_i \quad (17)$$

$$b_j = b_j - \gamma \nabla b_j \quad (18)$$

这里的 γ 为学习率(learning rate).

3.1.7 BPR算法

如算法1即为采用SGD求解的BPR算法。

算法 1: The SGD algorithm for BPR

```

1 initialize the model parameter  $\Theta$ ;
2 for  $t_1 = 1, \dots, T$  do
3   for  $t_2 = 1, \dots, |\mathcal{P}|$  do
4     Randomly pick up a pair  $(u, v_i) \in \mathcal{P}$ ;
5     Randomly pick up an item  $v_j$  from  $\mathcal{I} \setminus \mathcal{I}_u^+$ ;
6     Calculate the gradients via Eq.(9-13);
7     Update the model parameters via Eq.(14-18);
8   end
9 end
    
```

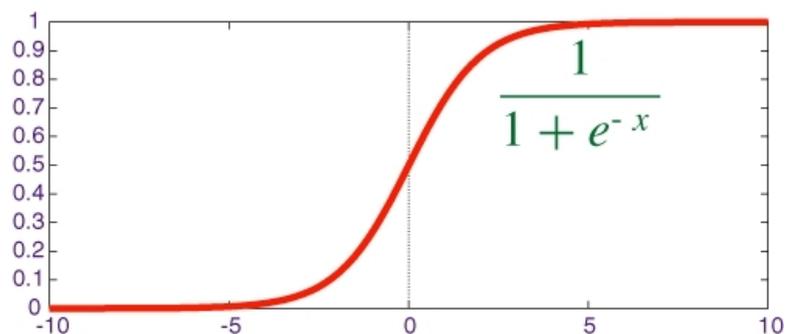
3.1.8 收敛缓慢的原因

由于上面的均匀采样方式会产生很多对于参数学习贡献微弱的train pairs,因此常常会导致收敛缓慢。确切的讲, 对于一个给定的训练采样 $(u, i, j) \in D_S$, 由公式7对随机梯度下降的任意一参数 $\theta \in \Theta$ 求其偏导:

$$\frac{\partial L_{feedback}}{\partial \theta} = -f(-x_{uij}) \frac{\partial (x_{uij})}{\partial \theta} = (f(x_{uij}) - 1) \frac{\partial (x_{uij})}{\partial \theta} \quad (19)$$

根据公式(19),如果 $f(x_{uij}) \rightarrow +1$,随机梯度将接近于0, 则训练采样 (u, i, j) 对于优化目标的贡献将会变得很小。

联系公式(19)与公式(1), 由图4 sigmoid函数图像可得, 当 $f(x_{uij}) \rightarrow +1$ 时, 也就是 $x_{uij} = \hat{r}_{ui} - \hat{r}_{uj}$ 越来越大, 即用户对于物品 v_i 与 v_j 的预测差值越来越大. 因此为了加速学习, 针对一个已有的user-item pair中的物品 v_i , 要采样的物品 v_j 应当是 v_i 相比有竞争力的物品, 更进一步说也就是由该用户对于 v_i 与 v_j 的偏好得分应该是相近的, 否则这个采样对于SGD便是低效的采样。

图 4: sigmoid函数 $f(x)$ 图像

从经验上来讲, 每个用户只会浏览一小部分的物品并对这些浏览过的物品提供一些交互反馈。如果均匀采样器均等地从整个物品集合中采样negative item. 对于一个user-item pair, 大部分均匀采样的物品并不具有可比性或者很难被相关的用户浏览。举个例子, iPhone与牙刷或iPhone与一个冷门的手机品牌可能会经常被均匀采样器采得。而由于这些低效的training pair对于SGD几乎作用很小, 整个训练过程便会收敛地极其缓慢。

除此以外, 与经典的分解技术相似, 如果一个用户或物品缺乏足够的反馈, 其对应的隐式表达往往不能够被很好的学习到。在现实世界数据集中, 用户行为与物品流行度的分布往往呈现长尾状。这就导致了大部分的用户和物品仅仅有很小部分的反馈数据。此外, 在真实的推荐系统中, 新的个体可能在任何时间被加入到推荐系统中。因此, BPR框架也很容易受制于冷启动问题。

3.2 Latent Dirichlet Allocation

Latent Dirichlet allocation(LDA), 隐含狄利克雷分布, 是一种主题模型(topic model), 它可以为文档集中每篇文档的主题按照概率分布的形式给出。同时它是一种无监督学习算法, 在训练时不需要手工标注的训练集, 需要的仅仅是文档集以及指定主题的数量即可。此外LDA的另一个优点则是, 对于每一个主题均可找出一些词语来描述它。

LDA首先由于2003年提出^[6], 目前在文本挖掘领域包括文本主题识别、文本分类以及文本相似度计算方面都有应用。

3.2.1 数学模型

LDA是一种典型的词袋(Bag-of-words)模型, 即它认为一篇文档(document)是由一组词(word)构成的一个集合, 词与词之间没有顺序以及先后的关系。一篇文档可以包含多个主题(topic), 文档中每一个词都由其中的一个主题生成。

另外, 正如Beta分布是二项式分布的共轭先验概率分布, 狄利克雷分布作为多项式分布的共轭先验概率分布。因此正如图5, LDA贝叶斯网络结构中所描述的, 在LDA模型中一篇文档生成的方式如下:

- 从狄利克雷分布 α 中取样生成文档 i 的主题分布 θ_i
- 从主题的多项式分布 θ_i 中取样生成文档 i 第 j 个词的主题 $z_{i,j}$

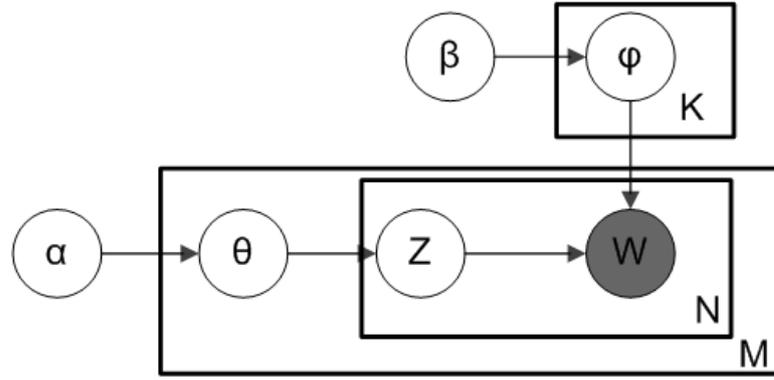


图 5: LDA 贝叶斯网络结构

- 从狄利克雷分布 β 中取样生成主题 $z_{i,j}$ 的词语分布 $\phi_{z_{i,j}}$
- 从词语的多项式分布 $\phi_{z_{i,j}}$ 中采样最终生成词语 $w_{i,j}$

因此整个模型中所有可见变量以及隐藏变量的联合分布是

$$p(w_i, z_i, \theta_i, \Phi | \alpha, \beta) = \prod_{j=1}^N p(\theta_i | \alpha) p(z_{i,j} | \theta_i) p(\Phi | \beta) p(w_{i,j} | \theta_{z_{i,j}}) \quad (20)$$

最终一篇文档的单词分布的最大似然估计可以通过将上式的 θ_i 以及 Φ 进行积分和对 z_i 进行求和得到

$$p(w_i | \alpha, \beta) = \int_{\theta_i} \int_{\Phi} \sum_{z_i} p(w_i, z_i, \theta_i, \Phi | \alpha, \beta) \quad (21)$$

根据 $p(w_i | \alpha, \beta)$ 的最大似然估计，最终可以通过吉布斯采样等方法估计出模型中的参数。

3.2.2 使用吉布斯采样估计LDA参数

在LDA最初提出的时候，人们使用EM算法(Expectation-maximization algorithm)进行求解，后来人们普遍开始使用较为简单的Gibbs Sampling，具体过程如下：

- 首先对所有文档中的所有词遍历一遍，为其都随机分配一个主题，即 $z_{m,n} = k \sim Mult(1/K)$ ，其中 m 表示第 m 篇文档， n 表示文档中的第 n 个词， k 表示主题， K 表示主题的总数，之后将对应的 $n_m^{(k)} + 1$, $n_m + 1$, $n_k^{(t)} + 1$, $n_k + 1$ ，他们分别表示在 m 文档中 k 主题出现的次数， m 文档中主题数量的和， k 主题对应的 t 词的次数， k 主题对应的总词数。
- 之后对下述操作进行重复迭代。
- 对所有文档中的所有词进行遍历，假如当前文档 m 的词 t 对应主题为 k ，则 $n_m^{(k)} - 1$, $n_m - 1$, $n_k^{(t)} - 1$, $n_k - 1$ ，即先拿出当前词，之后根据LDA中topic sample的概率分布sample出新的主题，在对应的 $n_m^{(k)}$, n_m , $n_k^{(t)}$, n_k 上分别+1。

$$p(z_i = k | z_{-i}, w) \propto k(n_{k,-i}^{(t)} + \beta_t)(n_{m,-i}^{(k)} + \alpha_k) / (\sum_{t=1}^V n_{k,-i}^{(t)} + \beta_t) \quad (22)$$

- 迭代完成后输出主题-词参数矩阵 Φ 和文档-主题矩阵 Θ

$$\phi_{k,t} = (n_k^{(t)} + \beta_t) / (n_k + \beta_t) \quad (23)$$

$$\theta_{m,k} = (n_m^{(k)} + \alpha_k) / (n_m + \alpha_k) \quad (24)$$

3.3 本章小结

本章首先介绍了采用SGD求解的Bayesian Personalized Ranking(BPR)推荐算法, 并且对可能导致其收敛缓慢的均匀采样策略做了讨论。然后简要介绍了LDA模型。

4 适应性采样策略

在这一章中, 我们结合了内容信息与隐式反馈提出了一个非均匀的物品采样器(a non-uniform item sampler)。在本章中所提出的适应性采样策略(adaptive sampling strategy)自动地模拟了真实的数据分布并且具有适应性地挑选更有针对性的train pairs。

4.1 适应性采样策略概览

在现实世界的场景中, 用户常常会浏览同一个目录下的多个物品, 然后做出他们的选择。那么很显然, 我们应该采样具有针对性的物品, 比方说针对iPhone, 相对于毛巾或者某低档品牌的手机, 采样高档Samsung或者LG显然更具有可比性与合理性。

因此, 在适应性采样策略中, 我们倾向于采样那些对于用户已选择过的物品更具有可比性同时有很大机会被相关用户浏览的物品。更确切的说, 对于一个user-item pair(u_m, v_i), 我们通过以下的步骤采样一个更加合理的负样本(negative item) v_j :

1. 根据用户 u_m 与物品 v_i 的所在目录分布(categorical distribution), 首先推断对于事件用户 u_m 选择物品 v_i 会发生在哪个目录下。
2. 对于给定的一个目录, 在该目录下我们进一步选择物品 v_j 作为negative item, 而该物品同时又具有较高的概率能够被用户 u_m 所浏览。

4.2 类别分布

在适应性采样中, 首先需要知道用户与物品的类别分布(categorical distribution)。不过在有些实际的应用场景中, 由于缺乏类别信息, 推荐系统并无法直接得到用户与物品的类别分布。为了应对这个问题, 我们利用了所谓的隐式表达(the latent representation of an entity)来近似指示其类别信息。

首先我们假设一个entity可能属于多个目录 $C = \{c_1, c_2, \dots, c_k\}$, 并且它的类别分布服从幂率(power laws)分布^[38]。用 $y_i^e \in \mathbb{R}^k$ 表示entity e_i 的latent vector, 而矩阵 $Y_e = [y_1^e, y_2^e, y_3^e, \dots]$ 是从内容信息(content information)与隐式反馈(implicit feedback)学习得到的entities's latent representation。以推荐系统中的一个经典场景为例: 在推荐系统有两种类型的实体(entity), 也就是说用户users, 比如消费者, 和物品items, 比如说电影, 书籍和歌曲等。明确起见, 本论文使用上标 u 与 v 分别表示与用户user和物品item相关的变量。比如, y_m^u 表示the latent vector of user u_m , Y^u 表示the latent representation matrix of user, y_i^v 表示the latent vector of item v_i 。为了联系categorical distribution与the latent vector of entity, 我们认为entity e_i 属于目录 $c \in C$ 的概率 $p(c|e_i)$ 为标准化因子的混合(a mixture over standardized factors), 并将其定义为:

$$p(c|e_i) \propto \exp\left(\frac{y_{i,c}^e - \mu_c}{\sigma_c}\right) \quad (25)$$

这里的 $\mu_c = E(y_{*,c}^e)$, $\sigma_c = \text{Var}(y_{*,c}^e)$ 分别表示all entity factors的经验均值与方差(empirical mean and variance over all entity factors)。假设在用户与物品上的类别分布是相互独立的, 那么

就可以进一步推断user-item pair(u_m, v_i)同属于一个category c 的联合概率 $p(c|u_m, v_i)$:

$$p(c|u_m, v_i) = p(c|u_m)p(c|v_i) \quad (26)$$

根据其联合概率, 就可以根据时间用户 u_m 选择物品 v_i 采样一个目录 c 。

4.3 选取negative item v_j

对于给定一个目录 c , 下一步的目标便是在该目录下选取一个negative item v_j , 而 v_j 同时将有很大概率会被用户 u_m 所浏览。

4.3.1 物品浏览概率

一个简单点的做法, 我们可以将entity e_i 在目录 c 下的排序得分(ranking score)视作为 $p(c|e_i)$, 再进一步从根据它们的排序得分直接选择物品。但实际上, 浏览概率(browsing probabilities)与排序得分(ranking scores)并不等同, 显然两者之间存在差距。在实际场景中, 对于出现在排序列表(ranking lists)中的物品, 那些排在靠前位置的物品相对于靠后位置的物品, 往往有着极大的概率被用户所浏览。比如在整个列表中排名前三位的物品的极有可能都会被用户所浏览, 而他们排序得分不同的影响在这种情况下将微乎其微。为了应对这个问题, 对于给定目录下的物品采样我们分为两步进行:

1. 首先, 我们先根据经验分布(empirical distribution)从候选物品(candidates)中采样一个排序的位置 r ;
2. 然后, 在该目录下对物品进行排序, 返回在位置 r 处的物品作为我们采样的negative item。

典型地, 经验分布大致服从analytical law, 比如Geometric^[46]或Zipf^[2] distribution。在这里, 我们应用Geometric distribution到从目录 c 的排序列表中选取位置 $r(j)$ 处的物品 v_j :

$$p(v_j|c) \propto \exp(-r(j)/\lambda), \lambda \in \mathbb{R}^+ \quad (27)$$

这里的 $r(j)$ 表示物品 v_j 的排序位置, λ 是用来调整概率密度的超参数(hyper-parameter)。

4.3.2 如何对物品列表进行排序

在获得negative item的排序位置后, 接下来的任务便是如何在这个位置安排对应的物品。^[38]中有一个简单的方法: 将物品的latent factors 当作其ranking scores, 然后根据它们的排序得分(ranking scores)对物品进行排序。但是由于物品的latent factors在每轮迭代都会被更新, 这种方法不得不在每轮迭代每个目录下对物品进行重新排序。这会导致一个很高的计算复杂度, 因为每轮迭代需要花费 $O(kt \log t)$ 的运行时间来进行重新排序, 这里的 t 指物品数。为此在^[38]中同样提出一个妥协性的做法: 每迭代 $t \log t$ 轮再进行重新排序。不过这种妥协会很容易导致局部收敛(local convergence)。此外, 每隔 $t \log t$ 轮进行更新, 实际上在很多未更新的时候的采样相当于从一个随机的物品子集中随机采样, 此时的采样反而会产生副作用。更进一步, 由于items' latent vectors 是被随机初始化, 那么

排序列表在首次重排序之前其实是相当于一个随机序列。如果这个随机的排序列表未被及时更新,那么采样器实际上会衰退为从一个物品子集中随机采样的采样器。因此,需要一个新的采样方法来平衡效率与推荐表现。

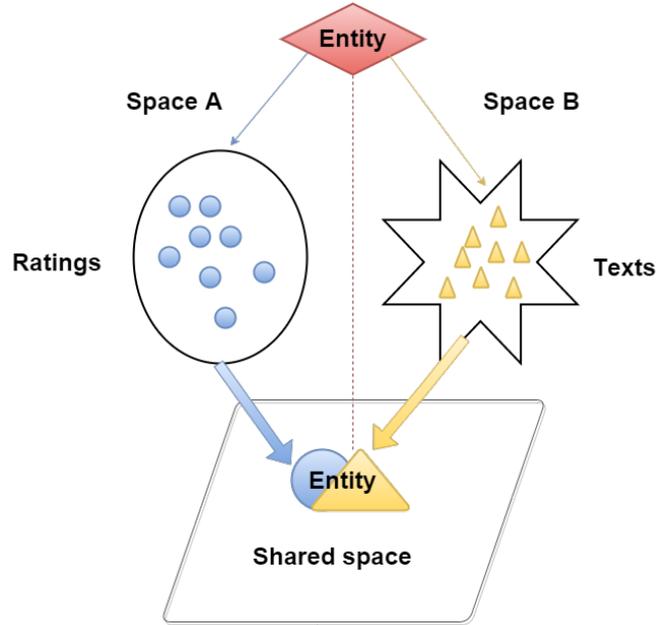


图 6: 将不同模态(different modalities)的entity映射到一个共享的隐式空间(a shared latent space). 在这里假设协同信息(collaborative information), 比如评分(rating), 和内容信息(content information), 比如文本(text), 分属于不同模态, 正如图中的space A, space B.

根据对于子空间的研究^[36;45],如果我们将一个entity从不同模态的映射到一个共享子空间,那么它在子空间中的表达应当是具有关联性的,比如互补(complementary)或是相似(similar)。如果我们独立地将一个item从content space和collaborative space映射到a shared latent space,那么我们就能够得到一个item在共享隐式空间的两个latent representations.如图3所示。为了避免采样器衰退为从一个物品子集中随机采样一个物品,我们通过物品的协同信息(collaborative filtering)来初始化排序列表(ranking lists)。具体来说,我们首先通过特征学习(feature learning)的方法从协同信息中学习物品的一个近似的隐式表达(latent representation),比如,用于图像的Conventional Neural Networks(CNN),用于文本的Latent Dirichlet Allocation(LDA)。那么,我们将latent factors视为在目录分布下的物品排序得分,然后在每个目录下对物品进行排序。最终,我们就根据这些排序后的结果对物品排序列表进行初始化。

此外,为了避免局部收敛的问题,同时平衡效率,我们只对于那些热门目录下的物品进行重排序。根据公式(26),首先对于一个user-item pair选定一个目录,然后进一步计算在每个目录下出现了多少所观测的user-item pairs.定义变量 $\rho \in \mathbb{R}^k$ 来表示目录的热度(popularity of categories)。在每次迭代中,我们根据目录的热度采样出一个热门目录(popular category) c :

$$p(c|p) \propto \exp\left(\frac{\rho_c - \mu}{\sigma}\right) \quad (28)$$

这里的 μ 和 σ 分别表示 ρ 的经验均值与方差(empirical mean and variance)。然后,我们将物品的current latent factors视为物品的新排序得分,并衡量在目录 c 下的new score vector,根据a similarity

function $sim(\cdot, \cdot)$ 与旧的score vector相比是否有较大变化.如果ranking score vector的变化超过了阈值 δ , 就用物品的latent representation matrix 的第 c 列 $y_{*,c}^v$ 来更新在目录 c 下的ranking scores, 并且对该目录下的物品进行重新排序.

4.4 适应性采样算法

总言之,在本论文中所研究的适应性采样策略如算法2所示, 对于一个user-item pair(u_m, v_i), 采样一个negative item v_j , 而 v_j 与 v_i 相比, 不仅具有可比性, 而且具有较高的几率为用户所浏览. 在算法2中, $index(c, r)$ 返回在排序列表 $l_c \in L$ 中位置在 r 处的物品. $x_c \in X$ 是在目录 c 下的ranking score vector, 而 x_c 正是由从协同信息学习而来approximate latent representation所初始化. 值得注意的是, 在整个学习过程中,本论文的适应性采样策略仅需要在一些热门目录重排序几次, 这不仅降低了计算复杂度同时避免了局部极值(local extremum)。

算法 2: Content-aware and Adaptive sampling

输入:

The observed user-item pair set \mathcal{P} ;
 The counters of category popularity ρ ;
 The latent representation matrixes Y^u and Y^v ;
 The ranking scores of items $X = \{x_1, x_2, \dots, x_k\}$;
 The orders of items $L = \{l_1, l_2, \dots, l_k\}$;

输出:

The training triple (u_m, v_i, v_j);
 The category popularity ρ ;
 Draw a category from $p(c|\rho)$;

- 1 Draw a popular category c from $p(c|\rho)$;
- 2 **if** $sim(x_c, y_{*,c}^v) > \delta$ **then**
- 3 Update x_c by $y_{*,c}^v$;
- 4 Reorder items under c and update l_c ;
- 5 **end**
- 6 Draw $(u_m, v_i) \in \mathcal{P}$ uniformly;
- 7 Draw a category c from $p(c|u_m, v_i), (1 \leq c \leq k)$;
- 8 $\rho_c ++$;
- 9 Draw a rank r from $p(r) \propto \exp(-r/\lambda), (1 \leq c \leq k)$;
- 10 $v_j \leftarrow \begin{cases} index(c, r) & \text{if } sgn(y_{m,c}^u) = 1; \\ index(c, n - r - 1) & \text{else} \end{cases}$;

4.5 本章小结

本章主要介绍了适应性采样策略, 该采样策略通过采样一个具有可比性同时又有较大概率被用户浏览的物品作为negative item. 该采样策略不仅能够降低计算复杂度同时能够避免局部极值。

5 融合内容信息的适应性BPR

在上述章节中,我们阐述了如何通过一个适应性采样策略加快BPR的学习,同时通过仅考虑隐式反馈学习了the latent factors of entities。不过,在现实世界的推荐系统中,很可能没有足够的协同信息,比如,新的物品可能会在任何时间被加入到推荐系统中。因此,我们提出一个更为全面的个性化推荐方法: Content-aware and Adaptive Bayesian Personalized Ranking,它基于上面所提出的适应性采样策略,同时将隐式反馈与内容信息融合入一个统一的推荐框架中。

5.1 Learning Content-aware Mappings

我们首先正提出一个对于学习content-aware mappings的一个非监督解决方案。用矩阵 $A^e = [a_1^e, a_2^e, a_3^e, \dots]$ 来表示content features of entities.然后我们提出对于学习content-aware mappings的目标函数:

$$L_{content} = \|A^e W^e - Y^e\|_F^2 \quad (29)$$

这里的 $W^e \in \mathbb{R}^{d^e \times k}$ 表示映射矩阵(mapping matrix), k 表示latent vectors的维度。

5.2 Parameter Inference of CA-BPR

通常来讲,由于缺乏监督信息(supervised information),在公式所表述的优化问题并无确定解法。不过,根据子空间的研究,我们可以从隐式反馈中学习一个latent matrix \widetilde{Y}^e ,并用 \widetilde{Y}^e 近似代替 Y^e 。因此,将 \widetilde{Y}^e 代替 Y^e 代入公式,那么目标函数变为:

$$L_{content} = \|A^e W^e - \widetilde{Y}^e\|_F^2 \quad (30)$$

使用 \widetilde{Y}^e 近似代替 Y^e 不仅能够优化目标函数,同时还能够一起学习包含协同信息与内容信息的 W^e 。因此,算法总体的目标函数如下:

$$\begin{aligned} \arg \min_{\Theta, W} L_{feedback} + L_{content} = & - \sum_{(m,i,j) \in D_s} \ln f(r_{mij}) + \lambda \|\theta\|^2 \\ & + \|A^e W^e - Y^e\|_F^2 + \frac{1}{2} \sum_{e \in \{u,v\}} \lambda^e \|W^e\|_F^2 \end{aligned} \quad (31)$$

这里的 $r_{mij} = r_{mi} - r_{mj}$ 。为了学习在公式中的参数 Y^u, Y^v, W^u, W^v ,在每轮迭代中,当我们更新latent factor matrix Y^e ,将矩阵 W^e 认为是一个常量(constant),并将 $L_{content}$ 视作一个正则化项(regularizer)。那么,对于一个任意latent parameter θ 的梯度如下:

$$\begin{aligned} \frac{\partial L}{\partial \theta} = & \sum_{(m,i,j) \in D_s} (f(r_{mij}) - 1) \frac{\partial (r_{mij})}{\partial \theta} \\ & + \frac{\partial \sum_{e \in \{u,v\}} \lambda^e (\|A^e W^e - Y^e\|_F^2)}{\partial \theta} + \lambda \theta \end{aligned} \quad (32)$$

对于参数 θ 的更新公式为: $\theta = \theta - \gamma \frac{\partial L}{\partial \theta}$,这里的 γ 为学习率(learning rate)。另一方面,对于一个latent

factor matrix Y^e , 将 Y^e 视为伪标签(pseudo labels), 并视 $L_{feedback}$ 为常量。因此对目标求偏导:

$$\frac{\partial L}{\partial W^e} = (A^e)^T (A^e W^e - Y^e) + \lambda^e W^e \quad (33)$$

令 $\frac{\partial L}{\partial W^e} = 0$, 那么对于 W^e 的更新公式则演变为:

$$W^e = \left((A^e)^T A^e + \lambda^e \mathbb{E} \right) A^e Y^e \quad (34)$$

这里的 $\mathbb{E} \in \mathbb{R}^{k \times k}$ 表示一个单位矩阵。

总而言之, 对于CA-BPR的参数学习如算法2所示.

算法 3: Learning paramters for BPR

输入:

The observed user-item pair set S ;
 The feature matrix of items F ;
 The content features entities $A := \{A^u, A^v\}$;

输出:

$\Theta := \{Y^u, Y^v\}$;
 $W := \{W^u, W^v\}$;

- 1 initialize the model parameter Θ and W with uniform $(-\sqrt{6}/k, \sqrt{6}/k)$;
 - 2 standarized Θ ;
 - 3 Initialize the popularity of categories ρ randomly;
 - 4 **repeat**
 - 5 Draw a triple (m, i, j) with 算法2;
 - 6 **for** each latent vector $\theta \in \Theta$ **do**
 - 7 $\theta \leftarrow \theta - \eta \frac{\partial L}{\partial \theta}$
 - 8 **end**
 - 9 **for** each $W^e \in W$ **do**
 - 10 Update W^e with the rule defined in Eq.34;
 - 11 **end**
 - 12 **until** convergence;
-

5.3 本章小结

本章通过学习了一个mapping矩阵利用了内容信息, 同时将本文所研究的适应性采样策略融合BPR的推荐框架中, 提出了CA-BPR推荐算法。

6 实验论证

6.1 数据集

本实验采用了MovieLens⁸100k的数据集,并随机分割了数据集的80%作为训练数据,其余20%作为测试数据。

MovieLens包含了943个用户对于1682个电影的100,000个评分数据。每个用户至少对20个电影评过。在实验中,用户的职业信息(occupational description)被用作用户的内容信息(content information),电影标题中的关键词被用作电影的内容信息。与^[14]中的处理过程相同,我们并不直接使用用户的等级评分数据,而将其转化为隐式反馈数据(对电影评过分为positive,未评过分为negative)来使用,以此推测是否用户是否会有对电影进行评分的行为。因此,对于一个特定的用户而言,我们的任务就是为其预测的一个有着潜在评分可能电影的排序列表。

如图7所示,MovieLens中的用户所评分过电影数目显然呈长尾分布,有422个近一半的用户所评分过的电影个数在区间[20,56]中。

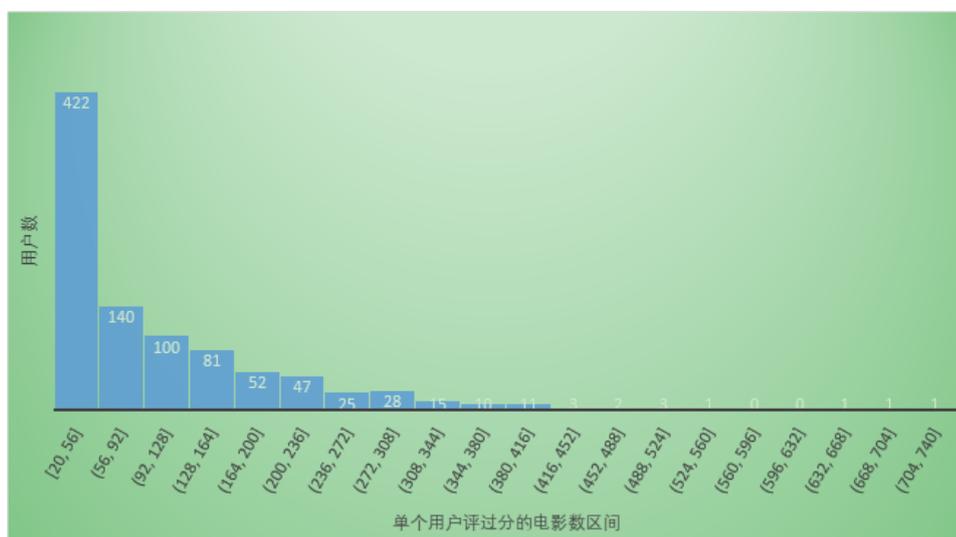


图 7: 用户对电影评分个数区间的长尾分布

6.2 评测标准

MAP: 先看AP(Average Precision), AP即为平均准确率。对于AP可以用这种方式理解:假使当我们使用google搜索某个关键词,返回了10个结果。当然最好的情况是这10个结果都是我们想要的相关信息。但是假如只有部分是相关的,比如5个,那么这5个结果如果被显示的比较靠前也是一个相对不错的结果。但是如果这个5个相关信息从第6个返回结果才开始出现,那么这种情况便是比较差的。这便是AP所反映的指标,与recall的概念有些类似,不过是“顺序敏感的recall”。

⁸<http://grouplens.org/datasets/movielens/>

对于 u 的平均准确率定义为:

$$AP_u = \frac{1}{|\mathcal{I}_u^{te}|} \sum_{i \in \mathcal{I}_u^{te}} \frac{\sum_{j \in \mathcal{I}_u^{te}} \delta(p_{uj} \prec p_{ui}) + 1}{p_{ui}}$$

在这里 p_{ui} 表示推荐列表中物品 i 的排序位置。 $p_{uj} \prec p_{ui}$ 表示在对用户 u 的排序列表中物品 j 的排序位置在物品 i 的前面。

对于MAP(Mean Average Precision)就很容易知道即为所有用户的AP的均值而已。那么则有:

$$MAP = \frac{\sum_{u \in \mathcal{U}^{te}} AP_u}{|\mathcal{U}^{te}|}$$

NDCG: 先从CG(Cummulative Gain)说起, CG即将每个推荐结果相关性的分值累加后作为整个推荐列表的得分。

$$CG_p = \sum_{i=1}^p rel_i$$

在 rel_i 表示处于位置 i 的推荐结果的相关性, p 表示所要考察的推荐列表的大小

CG的一个缺点是没有考虑结果处于不同位置对结果的影响, 例如我们总是希望相关性高的结果应排在前面, 相关性低的结果排在靠前的位置会严重影响用户体验, 所以在CG的基础上引入位置影响因素, 即DCG(Discounted Cummulative Gain):

$$DCG_p = \sum_{i=1}^p \frac{2^{rel_i} - 1}{\log_2(i + 1)}$$

DCG仍然有其局限之处, 即不同的推荐列表之间, 很难进行横向的评估。而我们评估一个推荐系统, 不可能仅使用一个用户的推荐列表及相应结果进行评估, 而是对整个测试集中的用户及其推荐列表结果进行评估。那么不同用户的推荐列表的评估分数就需要进行归一化, 也即NDCG(Normalized Discounted Cummulative Gain)。

IDCG(Ideal DCG)为推荐系统某一用户返回的最好结果, 即假设返回结果按照相关性排序, 最相关的结果放在最前面, 此序列的DCG为IDCG。因此DCG的值介于 $(0, IDCG]$, 故NDCG的值介于 $(0, 1]$ 。

对于用户 u 的NDCG@k定义为:

$$NDCG_u@k = \frac{DCG_u@k}{IDCG_u}$$

那么, 则有:

$$NDCG@k = \frac{\sum_{u \in \mathcal{U}^{te}} NDCG_u@k}{|\mathcal{U}^{te}|}$$

在具体操作中, 可以事先确定推荐目标和推荐结果的相关性分级, 例如可以使用0, 1分别表示相关或不相关, 比如此处我们用 $ref_i = \delta(i \in \mathcal{I}_u^{te})$, 在这里如果 x 为true, 则 $\delta(x) = 1$, 否则 $\delta(x) = 0$ 。或是这是0.5分别表示严重不相关到非常相关, 也即相当于确定了 rel 值的范围。之后对于每一个推荐目标的返回结果给定 rel 值, 然后使用DCG的计算公式计算出返回结果的DCG值。使用根据排序

后的 rel 值序列计算IDCG值，即可计算NDCG。

6.3 实验过程与分析

我们对BPR-MF与CA-BPR分别就MAP与NDCG评测指标进行了比较。BPR-MF^[39]应用了矩阵分解的BPR算法框架，同时采用均匀采样策略选取训练采样。CA-BPR^[15]在利用了隐式反馈数据的基础上同时融入了内容信息，并采取非均匀的适应性采样策略。

表2显示了实验方法的不同之处。

表 2: BPR-MF与CA-BPR方法特征比较

| Method | Content | Sampling |
|--------|---------|-------------|
| BPR-MF | no | uniform |
| CA-BPR | yes | non-uniform |

表3显示了BPR-MF与CA-BPR的MAP与NDCG实验结果。

表 3: 不同维度k下算法MAP与NDCG实验结果

| BPR-MF | k=10 | k=20 | k=30 | k=40 | k=50 |
|---------|--------|--------|--------|--------|--------|
| MAP | 0.0879 | 0.0877 | 0.1043 | 0.0888 | 0.1074 |
| NDCG@3 | 0.3051 | 0.3545 | 0.3398 | 0.2491 | 0.3790 |
| NDCG@5 | 0.3616 | 0.4296 | 0.3708 | 0.2984 | 0.4153 |
| NDCG@10 | 0.4120 | 0.4632 | 0.4010 | 0.3163 | 0.4458 |
| NDCG@20 | 0.4121 | 0.4575 | 0.4164 | 0.3415 | 0.4323 |

| CA-BPR | k=10 | k=20 | k=30 | k=40 | k=50 |
|---------|--------|--------|--------|--------|--------|
| MAP | 0.1074 | 0.1072 | 0.1274 | 0.1016 | 0.1229 |
| NDCG@3 | 0.3790 | 0.4336 | 0.4152 | 0.3044 | 0.4631 |
| NDCG@5 | 0.4153 | 0.4752 | 0.4531 | 0.3646 | 0.5074 |
| NDCG@10 | 0.4458 | 0.5101 | 0.4900 | 0.3865 | 0.5447 |
| NDCG@20 | 0.4323 | 0.4946 | 0.5088 | 0.4173 | 0.5282 |

图8显示了BPR-MF与CA-BPR在不同维度下的MAP结果对比。显然，融合了内容信息同时采用适应性采样策略的CA-BPR推荐效果比BPR-MF要好，由此说明内容信息及适应性采样策略的确是有效的。

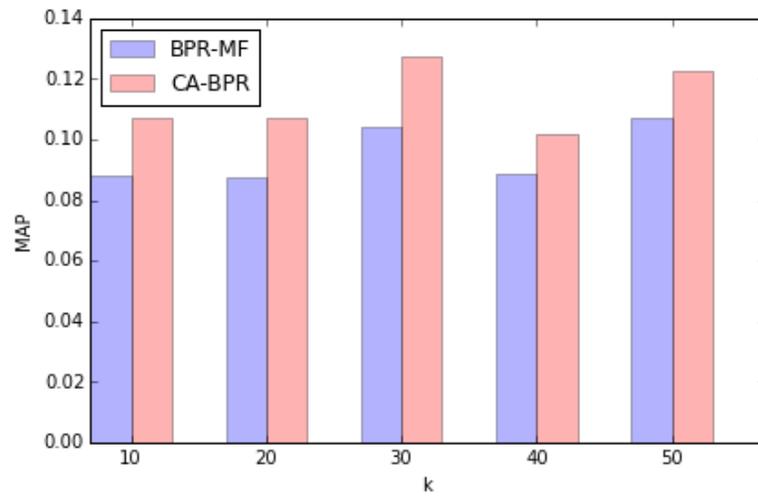


图 8: 不同维度算法MAP结果对比

7 结论与展望

7.1 本文的主要内容

本文首先回顾了采用均匀采样策略的经典BPR推荐算法，然后分析其在随机梯度学习算法中导致收敛缓慢的原因。而后在隐式反馈的基础上加入内容信息提出了非均匀的适应性采样策略，并将其融入BPR推荐框架中。实验证明本文所研究的方法的确能够提高推荐效果。

7.2 进一步的研究工作

尽管本文实验证明通过加入内容信息的确有助于提高推荐效果，但是对于加入内容信息的适应性采样策略在整个学习过程每个阶段的影响仍然有待研究。同时对于一些已有的一些融合内容信息的推荐方法，比如采用Word2Vec技术，还需进一步的研究调查在这些融合内容信息的不同推荐方法中的特点，适用性及其局限性。

【参考文献】

- [1] Ayse Merve Acilar and Ahmet Arslan. A collaborative filtering method based on artificial immune network. *Expert Syst. Appl.*, 36(4):8324–8332, 2009.
- [2] Lada A Adamic and Bernardo A Huberman. Zipf’ s law and the internet. *Glottometrics*, 3(1):143–150, 2002.
- [3] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6):734–749, 2005.
- [4] Amir Albadvi and Mohammad Shahbazi. A hybrid recommendation technique based on product category attributes. *Expert Syst. Appl.*, 36(9):11480–11488, 2009.
- [5] Marko Balabanović and Yoav Shoham. Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72, 1997.
- [6] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.
- [7] John S Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 43–52. Morgan Kaufmann Publishers Inc., 1998.
- [8] Robin Burke. Hybrid web recommender systems. In *The adaptive web*, pages 377–408. Springer, 2007.
- [9] Keunho Choi, Donghee Yoo, Gunwoo Kim, and Yongmoo Suh. A hybrid online-product recommendation system: Combining implicit rating-based collaborative filtering and sequential pattern analysis. *Electronic Commerce Research and Applications*, 11(4):309–317, 2012.
- [10] Anja Feldmann and Ward Whitt. Fitting mixtures of exponentials to long-tail distributions to analyze network performance models. In *INFOCOM’97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution., Proceedings IEEE*, volume 3, pages 1096–1104. IEEE, 1997.
- [11] Zeno Gantner, Lucas Drumond, Christoph Freudenthaler, Steffen Rendle, and Lars Schmidt-Thieme. Learning attribute-to-feature mappings for cold-start recommendations. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 176–185. IEEE, 2010.
- [12] Rainer Gemulla, Erik Nijkamp, Peter J Haas, and Yannis Sismanis. Large-scale matrix factorization with distributed stochastic gradient descent. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 69–77. ACM, 2011.
- [13] Sheetal Girase, Debajyoti Mukhopadhyay, et al. Role of matrix factorization model in collaborative filtering algorithm: A survey. *arXiv preprint arXiv:1503.07475*, 2015.

- [14] Asela Gunawardana and Christopher Meek. A unified approach to building hybrid recommender systems. In *Proceedings of the third ACM conference on Recommender systems*, pages 117–124. ACM, 2009.
- [15] Weiyu Guo, Shu Wu, Liang Wang, and Tieniu Tan. Adaptive pairwise learning for personalized ranking with content and implicit feedback. In *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, WI-IAT 2015, Singapore, December 6-9, 2015 - Volume I*, pages 369–376, 2015.
- [16] Jonathan L Herlocker, Joseph A Konstan, Al Borchers, and John Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 230–237. ACM, 1999.
- [17] Liangjie Hong, Aziz S Doumith, and Brian D Davison. Co-factorization machines: modeling user interests and predicting individual decisions in twitter. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 557–566. ACM, 2013.
- [18] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), December 15-19, 2008, Pisa, Italy*, pages 263–272, 2008.
- [19] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 263–272. Ieee, 2008.
- [20] Ahmad A. Kardan and Mahnaz Ebrahimi. A novel approach to hybrid recommendation systems based on association rules mining for content recommendation in asynchronous discussion groups. *Inf. Sci.*, 219:93–110, 2013.
- [21] Yong Soo Kim and Bong-Jin Yum. Recommender system based on click stream data using association rule mining. *Expert Systems with Applications*, 38(10):13320–13327, 2011.
- [22] Ken Lang. Newsweeder: Learning to filter netnews. In *Proceedings of the 12th international conference on machine learning*, pages 331–339, 1995.
- [23] Seok Kee Lee, Yoon Ho Cho, and Soung Hie Kim. Collaborative filtering with ordinal scale-based implicit ratings for mobile music recommendations. *Information Sciences*, 180(11):2142–2155, 2010.
- [24] Tong-Queue Lee, Young Park, and Yong-Tae Park. A similarity measure for collaborative filtering with implicit feedback. In *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence, Third International Conference on Intelligent Computing, ICIC 2007, Qingdao, China, August 21-24, 2007, Proceedings*, pages 385–397, 2007.
- [25] Xirong Li, Cees GM Snoek, and Marcel Worring. Learning social tag relevance by neighbor voting. *Multimedia, IEEE Transactions on*, 11(7):1310–1322, 2009.

- [26] Yanen Li, Jia Hu, ChengXiang Zhai, and Ye Chen. Improving one-class collaborative filtering by incorporating rich user information. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 959–968. ACM, 2010.
- [27] Greg Linden, Brent Smith, and Jeremy York. Amazon. com recommendations: Item-to-item collaborative filtering. *Internet Computing, IEEE*, 7(1):76–80, 2003.
- [28] Dong Liu, Xian-Sheng Hua, Linjun Yang, Meng Wang, and Hong-Jiang Zhang. Tag ranking. In *Proceedings of the 18th international conference on World wide web*, pages 351–360. ACM, 2009.
- [29] Hao Ma, Dengyong Zhou, Chao Liu, Michael R Lyu, and Irwin King. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 287–296. ACM, 2011.
- [30] Tao Mei, Bo Yang, Xian-Sheng Hua, Linjun Yang, Shi-Qiang Yang, and Shipeng Li. Videoreach: an online video recommendation system. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 767–768. ACM, 2007.
- [31] Raymond J Mooney and Loriene Roy. Content-based book recommending using learning for text categorization. In *Proceedings of the fifth ACM conference on Digital libraries*, pages 195–204. ACM, 2000.
- [32] Weike Pan and Li Chen. Gbpr: Group preference based bayesian personalized ranking for one-class collaborative filtering. In *IJCAI*, volume 13, pages 2691–2697, 2013.
- [33] Michael Pazzani and Daniel Billsus. Learning and revising user profiles: The identification of interesting web sites. *Machine learning*, 27(3):313–331, 1997.
- [34] Shuang Qiu, Jian Cheng, Ting Yuan, Cong Leng, and Hanqing Lu. Item group based pairwise preference learning for personalized ranking. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 1219–1222. ACM, 2014.
- [35] Reza Rafteh and Arash Bahrehmand. An adaptive approach to dealing with unstable behaviour of users in collaborative filtering systems. *Journal of Information Science*, 38(3):205–221, 2012.
- [36] Nikhil Rasiwasia, Jose Costa Pereira, Emanuele Coviello, Gabriel Doyle, Gert RG Lanckriet, Roger Levy, and Nuno Vasconcelos. A new approach to cross-modal multimedia retrieval. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 251–260. ACM, 2010.
- [37] Steffen Rendle. Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(3):57, 2012.
- [38] Steffen Rendle and Christoph Freudenthaler. Improving pairwise learning for item recommendation from implicit feedback. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 273–282. ACM, 2014.

- [39] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, pages 452–461. AUAI Press, 2009.
- [40] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *CSCW '94, Proceedings of the Conference on Computer Supported Cooperative Work, Chapel Hill, NC, USA, October 22-26, 1994*, pages 175–186, 1994.
- [41] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295. ACM, 2001.
- [42] Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, and David M Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 253–260. ACM, 2002.
- [43] Xiaoyuan Su and Taghi M Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009:4, 2009.
- [44] Marco Tiemann and Steffen Pauws. Towards ensemble learning for hybrid music recommendation. In *Proceedings of the 2007 ACM conference on Recommender systems*, pages 177–178. ACM, 2007.
- [45] Raghavendra Udupa and Mitesh Khapra. Improving the multilingual user experience of wikipedia using cross-language name search. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 492–500. Association for Computational Linguistics, 2010.
- [46] Kuansan Wang, Toby Walker, and Zijian Zheng. Pskip: estimating relevance ranking quality from web search clickthrough data. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1355–1364. ACM, 2009.
- [47] Zhi Wang, Lifeng Sun, Wenwu Zhu, Shiqiang Yang, Hongzhi Li, and Dapeng Wu. Joint social and content recommendation for user-generated videos in online social network. *Multimedia, IEEE Transactions on*, 15(3):698–709, 2013.
- [48] Xiao Yu, Xiang Ren, Yizhou Sun, Bradley Sturt, Urvashi Khandelwal, Quanquan Gu, Brandon Norrick, and Jiawei Han. Recommendation in heterogeneous information networks with implicit user feedback. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 347–350. ACM, 2013.
- [49] Wancai Zhang, Hailong Sun, Xudong Liu, and Xiaohui Guo. Temporal qos-aware web service recommendation via non-negative tensor factorization. In *Proceedings of the 23rd international conference on World wide web*, pages 585–596. ACM, 2014.

- [50] Nan Zheng and Qiudan Li. A recommender system based on tag and time information for social tagging systems. *Expert Syst. Appl.*, 38(4):4575–4587, 2011.
- [51] Hao Zhong, Weike Pan, Congfu Xu, Zhi Yin, and Zhong Ming. Adaptive pairwise preference learning for collaborative recommendation with implicit feedbacks. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 1999–2002. ACM, 2014.
- [52] 项亮. 推荐系统实践[B]. 北京:人民邮电出版社, 2012.

致谢

首先衷心地感谢潘微科老师。在本科生涯最后的一年多里，不仅是现时的学业与学术，更是对未来的发展给予了我很多指导与帮助。本次毕业设计，从选题到论文撰写，给予了我很多宝贵的意见。他渊博的学识、严谨的治学态度及认真负责的工作态度都使我受到鼓舞和熏陶。在此向潘微科老师表示崇高的敬意和衷心的感谢，他的言传身教将使我终生受益。

感谢key哥哥与在453认识的朋友们，与你们的交流大概就是我对计算机启蒙的开始。如果不是有幸与你们相识，这一路走来必是要曲折地多。

感谢Thuthesis 及其作者薛瑞尼。最终虽未使用Thuthesis模板，但是此间对其研习所得对我顺利使用L^AT_EX完成论文撰写仍然起了很大作用。

感谢一直关心我的父母与兄长。远游在外，感谢还有你们牵挂。

感谢自己熬过了那段难捱的日子。从学习画画到广播电视再到计算机科学，在如今看来似曾是做了诸多无用功，不过幸而没有因为短时的平庸迷茫而消磨掉满心的戾气。

前路漫漫，不冀求大步流星，唯盼能步步坚实。

Research on Content-Aware Collaborative Filtering

【Abstract】 Pairwise learning algorithms are a vital technique for personalized ranking with implicit feedback. They usually assume that each user is more interested in items which have been selected by the user than remaining ones. This pairwise assumption usually derives massive training pairs. To deal with such large-scale training data, the learning algorithms are usually based on stochastic gradient descent with uniformly drawn pairs. However, the uniformly sampling strategy often results in slow convergence. In this paper, we first uncover the reasons of slow convergence. Then, we associate contents of entities with characteristics of dataset to develop an adaptive item sampler for drawing informative training data. In this end, to devise a robust personalized ranking method, we accordingly embed our sampler into Bayesian Personalized Ranking (BPR) framework, and further propose a Content-aware and Adaptive Bayesian Personalized Ranking (CA-BPR) method, which can model both contents and implicit feedbacks in a unified learning process. The experimental results show that our adaptive item sampler can indeed improve recommendation performance.

【Keywords】 Recommendation System; Collaborative Filtering; Adaptive Sampling

指导教师: 潘微科