

基于用户时点可见性的无趣项挖掘及协同过滤推荐方法^{*}

石磊 李树青

(南京财经大学信息工程学院 南京 210023)

摘要:【目的】解决基于显式反馈信息的协同过滤算法无法处理数据稀疏性和用户选择偏差影响的问题。【方法】根据看见但未交互的项目表现用户的负面偏好,结合用户活跃度、项目流行度和时间因素综合衡量用户对项目的可见性。引入使用前偏好的概念,构建基于用户时点可见性的加权矩阵分解模型以识别缺失数据中用户不感兴趣的项目,并将其填充为低值。【结果】通过在 MovieLens 两个数据集的实验表明,经过基于无趣项挖掘与低值填充的数据填充算法(UIMLF)填充后,ItemCF 和 BiasSVD 的推荐精度平均提升 2~2.5 倍。【局限】仅依据“看见未交互”的项目表现用户负面偏好的经验对使用前偏好建模,可能存在经验偏差。【结论】所提方法能有效缓解数据稀疏性和用户选择偏差的影响,使推荐结果更准确。

关键词: 协同过滤 显式反馈 选择偏差 使用前偏好 无趣项

分类号: TP393

DOI: 10.11925/infotech.2096-3467.2021.0842

引用本文: 石磊, 李树青. 基于用户时点可见性的无趣项挖掘及协同过滤推荐方法[J]. 数据分析与知识发现, 2022, 6(5): 64-76.(Shi Lei, Li Shuqing. Mining Uninteresting Items with Visibility of User Time Points and Collaborative Filtering Recommendation Method[J]. Data Analysis and Knowledge Discovery, 2022, 6(5): 64-76.)

1 引言

在信息爆炸的时代,推荐系统在缓解信息过载问题上起着举足轻重的作用,已经被电子商务、在线新闻和社交媒体网站等许多在线服务广泛采用,是学术界和工业界的研究热点^[1]。

协同过滤(Collaborative Filtering, CF)是推荐系统中应用最广泛、研究最深入的方法之一。协同过滤算法基于用户历史行为数据,利用用户行为模式之间的相似性推断用户偏好,产生推荐^[2]。用户行为数据可分为两类:显式反馈数据(如评分、评论等)和隐式反馈数据(如购买、点击等)。显式反馈数据

主要用于评分预测场景,基于观察到的评分数据预测缺失评分。通过对所有项目的预测评分按大小排序,也可以进行 TopN 推荐。但已有研究指出仅考虑观察到的评分数据,忽视了数据并非随机缺失的事实,推荐效果并不理想^[3-4]。与之不同,隐式反馈数据同时考虑了观察到的数据和缺失数据,已被广泛应用于 TopN 推荐场景。隐式反馈的优势在于充分利用缺失数据隐含的负面偏好,但相较于显式反馈,隐式反馈对用户偏好的表达并不明确,且无法表达偏好的程度^[5]。因此,本文主要研究如何结合隐式反馈的优势,充分利用缺失数据改进现有基于显式反馈的协同过滤算法,使其有效地应用于 TopN 推

通讯作者(Corresponding author): 李树青(Li Shuqing), ORCID: 0000-0001-9814-5766, E-mail: leeshuqing@163.com。

^{*}本文系江苏省高等学校自然科学研究重大项目(项目编号: 19KJA510011)和江苏省研究生科研与实践创新计划项目(项目编号: KYCX20_1348)的研究成果之一。

The work is supported by Natural Science Major Foundation of the Jiangsu Higher Education Institutions of China (Grant No. 19KJA510011), Postgraduate Research & Practice Innovation Program of Jiangsu Province (Grant No. KYCX20_1348).

荐场景。

2 相关工作与分析

2.1 显式反馈方法的局限性分析

数据稀疏性和用户选择偏差是显式反馈方法面临的重要挑战。真实世界中,用户往往只对小部分项目进行评分,大多数项目都缺失评分值,模型难以准确学习用户真实偏好^[6]。更关键的是,评分数据带有用户选择偏差,用户倾向于选择可能给他们带来高满意度的项目,而忽略那些可能带来低满意度的项目^[4,7]。换言之,评分数据不是随机缺失的,而是用户自由选择的结果。这导致已有评分数据中大多数评分偏高,只有小部分为低分^[8],本文所用数据集的评分分布也符合这一情况。与之不同,有研究表明真实世界中用户往往只对小部分项目感兴趣,大多数项目都是不感兴趣的项目^[9]。即由于用户选择偏差的影响,观察到的评分数据并不是所有评分的代表性样本,其偏向于反映用户正面偏好,很多可能表明用户负面偏好的项目因评分缺失而缺乏利用。

上述问题导致了基于显式反馈的协同过滤算法在评分预测和 TopN 推荐这两个场景的表现不一致。评分预测场景表现好的算法,在 TopN 推荐场景却并不一定有效。Steck 发现这两个场景主要的区别在于所考虑的训练和测试数据^[10]。评分预测场景只与观察到的评分有关,为了评估预测评分的精度,现有研究往往仅对用户实际已评分的项目进行预测以计算误差。而 TopN 推荐场景往往需要预测所有的缺失评分并排序。仅从观察到的评分数据学习用户偏好,虽能较准确地预测实际已评分的项目,但观察到的评分数据并不是所有评分的代表性样本,不足以有效预测所有缺失评分。

综上所述,现有协同过滤算法往往仅考虑观察到的评分数据,忽视了数据稀疏性和用户选择偏差的影响,难以平衡地学习到用户的正面偏好和负面偏好,对缺失项的预测评分和排序存在偏差,从而导致 TopN 推荐效果不佳。因此,如何有效挖掘和利用缺失数据隐含的用户偏好是本文研究的关键。

2.2 已有工作分析

针对显式反馈方法对缺失数据的利用,数据填

充是一种直接且有效的手段。很多研究基于已有评分数据执行评分预测算法,将缺失项填充为该算法的预测值^[11-12]。Ma 等提出一种有效的缺失数据预测方法 (Effective Missing Data Prediction, EMDP), 结合基于用户和基于物品的协同过滤算法的预测评分填充缺失值,并优先考虑填充高可靠性的数据^[11]。Ren 等提出自动填充和自适应最大填充方法,能够自适应地从用户和项目的角度考虑邻域信息,识别并填充每个预测中关键缺失数据^[12]。上述方法虽然能够有效缓解数据稀疏性,在评分预测场景有不错的效果,但在 TopN 推荐场景中往往效果不佳,因为它们都存在高估缺失项评分的问题。

由于现有填充算法大多都基于已有评分数据训练模型^[11-12],往往没有考虑到已有评分数据带有用户选择偏差,因此大多数填充评分都偏高。Chac 等也注意到这个问题,发现模型如果从已有评分数据学习到如此高的评分,那么它也会为大部分缺失项生成高分^[13]。尽管大多数缺失项表明用户负面偏好,但上述方法仍很可能将其填充为高分。这些高估的缺失项评分将进一步加剧已有评分偏高的现象。这类填充算法虽然缓解了数据稀疏性,但没有考虑到用户选择偏差的影响,导致协同过滤算法难以学习用户对缺失项的真实偏好,TopN 推荐效果甚至更差。

不同于以上基于显式反馈的方法,基于隐式反馈的推荐方法对缺失数据的利用已经在 TopN 推荐场景取得丰富的成果。隐式反馈数据往往只有正反馈信息,缺乏负反馈,因此必须对缺失数据进行处理,主要有以下两种策略:一种是基于样本的学习,如基于采样的最小二乘法^[14] (Sampling-based Alternating Least Squares, sALS)、贝叶斯个性化排序的矩阵分解^[15] (Bayesian Personalized Ranking Optimizations for Matrix Factorization, BPR-MF), 其根据一定的抽样策略,从缺失数据中抽取负样本;另一种是基于整体数据的学习,如加权正则化矩阵分解^[16] (Weighted Regularized Matrix Factorization, WRMF)、基于元素级优化的交替最小二乘法^[17] (Element-Wise Alternating Least Squares, eALS), 其将所有缺失数据视为负样本,但给予权重以衡量其为负样本的置信度。前者不需要考虑所有缺失的数

据,效率较高,但存在丢失有价值信息的风险,鲁棒性较差;后者利用所有数据,覆盖率更高但效率较低。以上两种策略都基于一个经验,即大多数缺失项目表现用户的负面偏好。而显式反馈数据同时包含正反馈和负反馈,往往并没有有效利用缺失数据隐含的负面偏好,仅对观察到的数据进行建模,忽略了选择偏差的影响,TopN推荐效果不佳。

也有研究尝试将缺失数据隐含的负面偏好融入显式反馈方法,如PureSVD简单地使用0填充所有缺失项^[18],然后直接进行传统的奇异值分解。Steck证明了评分数据是非随机缺失的,认为缺失项携带用户负面偏好信息,为所有缺失项赋予统一的低值^[19]。上述方法实质上缓解了用户选择偏差的影响,从而有效增强了TopN推荐效果,但它们对缺失项的利用比较简单,未考虑到缺失项中也包含用户可能感兴趣的项目,并且其效果也没有在其他现有算法中得到验证。基于隐式反馈的方法对缺失数据的有效利用已经得到验证,但这些方法难以直接应用于显式反馈数据。本文探索一种通用的基于无趣项填充的解决方案,能够直接应用于现有的显式反馈方法,使它们能更好地应用于TopN推荐场景。

2.3 本文解决方案的设计思路

一些研究从缺失数据中挖掘用户负面偏好项目,并以统一的低值填充,这种方式已经取得一定的成果^[9, 18-19]。本文据此提出一种基于无趣项挖掘与低值填充的数据填充算法(Uninterested Items Mining and Low-score Filling, UIMLF)。基于用户时点可见性以更准确地识别缺失项中用户不感兴趣的项目,然后使用非统一的低值填充,缓解数据稀疏性的同时解决了用户选择偏差的影响,从而有效增强TopN推荐性能。因此,其中的关键在于如何识别缺失项中用户不感兴趣的项目。

根据Pan等的分析^[14],缺失项可能由两种原因导致:用户没见过该项目,不知道其存在,所以没有评分;用户见过该项目,知道其存在,但对其不感兴趣,所以没有评分。

其中,第二种缺失项明显表现出用户的负面偏好^[20],将其定义为无趣项。为识别缺失项中的无趣项,本文利用使用前偏好的表达方法^[9],将用户对项

目的偏好分为使用前偏好和使用后偏好。使用前偏好是用户购买和使用前对项目的印象,决定用户是否与项目交互;使用后偏好决定用户实际使用后对项目的评分。

由于在用户活跃时期流行的项目更可能被用户看见,而如果用户未进行交互,说明用户对其使用前偏好很低,即用户仅依据对其外在印象就决定不交互。本文据此提出基于用户时点可见性的加权矩阵分解算法(User Time Point Visibility Element-Wise Alternating Least Squares, UTV-eALS)对缺失项的使用前偏好建模训练,利用使用前偏好识别缺失项中的无趣项目,并结合用户评分均值以低值填充无趣项评分。最后将填充后的评分矩阵直接应用于现有的协同过滤算法。值得指出的是,使用前偏好反映用户与项目是否交互,本质上是一种隐式反馈。本文填充算法实际上结合用户可见性和隐式反馈的隐含信息识别缺失项中的无趣项并以低值填充,让显式评分数据也能有效利用缺失数据。

本文方法可以从两方面改进现有的协同过滤算法。第一,现有的协同过滤算法将所有未评分的项目作为TopN推荐的候选项,本文使用低值填充无趣项,将无趣项从推荐的候选项中排除。无趣项是使用前偏好低的项目,即使向用户推荐,用户也不会与之交互,这可以避免推荐那些可能预测为高分但用户实际不会交互的项目。第二,低值无趣项的填充不仅缓解了数据稀疏性,更重要的是充分利用了缺失数据隐含的负面偏好,缓解了选择偏差的问题。协同过滤算法能够平衡地学习用户正面偏好与负面偏好。本文的主要工作如下:

(1)分析并实验验证了用户选择偏差对TopN推荐的影响,为基于无趣项挖掘的填充方案提供理论基础;

(2)提出基于用户时点可见性的加权矩阵分解算法以识别无趣项,结合用户活跃度、项目流行度和时间因素综合衡量缺失项为负例的置信度;

(3)在MovieLens两个真实世界数据集上验证了本文填充方案的有效性,并通过实验表明充分利用缺失数据的显式反馈方法相较于目前流行的基于隐式反馈的推荐方法,能够做出更有效的推荐。

3 UIMLF 算法说明与实现

3.1 使用前偏好分析

使用前偏好一般基于用户对项目外部特征的偏好,即使用户与项目没有交互,也可以直接获取这些特征信息,如电影的类型、主演等。使用后偏好取决于用户对项目内在特征的偏好,如电影故事情节。使用前偏好决定用户是否与项目进行交互,Marlin 等证明了用户倾向于对其使用前偏好高的项目进行评分,并且他们的调查显示,在带有用户选择偏差的 MovieLens 和 Netflix 等数据集中,评级偏向高分;而在特别收集的无选择偏差的 Yahoo 数据集中,大部分评级为低分^[4]。Steck 的研究也表明,与高分相比,低评分从观察到的评分数据中缺失的可能性更大^[19]。因此,可以简单地推断使用前偏好低的项目,其使用后偏好一般也偏低,即评分偏低。而对于使用前偏好高的项目,其评分多数偏高,这与本文所用数据集的评分分布也相符。

对于用户已评分的项目,可以简单地推断其使用前偏好较高,因为用户开始一定对其感兴趣才会与之交互并且予以评价。挑战在于如何准确推断缺失项的使用前偏好。这是一个单类问题^[14],即已有评分数据都是正样本,缺乏负样本。基于之前的分析,为了保持模型的鲁棒性,本文使用基于整体数据的学习。WRMF 把所有缺失数据作为负样本并分配较小的统一权重^[16],认为缺失项有相同的置信度为负样本,但这与实际情况不符。一般来说,流行的项目更容易被用户看见。He 等基于物品流行度对缺失项目进行非均匀加权^[17],热门项目更容易被用户看见,而用户未交互,其为负样本的置信度就越高,应给予更高的权重,但未考虑到用户之间的差异和时间因素对流行度的影响。

3.2 UTV-eALS 模型

目前基于整体数据学习的方法的权重设计大多基于一个经验,“看见未交互”的项目表现出用户的负面偏好^[20]。越可能被用户看见的项目而用户未交互,其为负例的置信度就越高,给予其更高的权重。关键在于如何准确推断用户是否看见该项目。目前流行的做法是使用项目流行度推断用户对项目的可见性,可信度不高。事实上,不同活跃度的用户对项

目具有不同的可见性^[21]。Pan 等分别从用户和项目的角度提出两种加权方案,认为越活跃的用户,其缺失项目更可能是负面的,并且其研究结果显示基于用户活跃度的加权方式相较于项目流行度效果更好^[14]。然而,上述研究都没有结合项目流行度和用户活跃度综合衡量用户对项目的可见性,并且没有考虑时间因素。值得注意的是,项目流行度是随时间变化的,网站对于流行项目的展示也具有实时性^[21]。使用全局的项目流行度粒度较粗,在用户活跃时期流行的项目更可能被用户看见。

基于以上分析,本文线性结合用户活跃时期的项目流行度和用户活跃度衡量项目对于用户的时点可见性,对缺失项进行非均匀加权。同时,为了缓解基于整体数据学习带来的效率问题,使用 eALS 优化目标算法^[17],eALS 为处理加权矩阵分解提供了一种新的思路,在元素级别优化参数,避免了传统 ALS 矩阵分解时需要矩阵求逆,提高了效率。

首先使用原评分矩阵 $\mathbf{R} = (r_{ui})_{m \times n}$ 构建使用前偏好矩阵 $\mathbf{P} = (p_{ui})_{m \times n}$ 。其中, m 为用户数量, n 为项目数量, r_{ui} 为用户 u 对物品 i 的评分。根据 3.1 节分析,已评分项都为正样本,使用前偏好设为 1。所有缺失项看作负样本,使用前偏好设为 0,如公式(1)所示。

$$p_{ui} = \begin{cases} 0, & r_{ui} = \text{null} \\ 1, & r_{ui} \neq \text{null} \end{cases} \quad (1)$$

由于缺失项并非都是负样本,本文构建权重矩阵 $\mathbf{W} = (w_{ui})_{m \times n}$ 。对于已评分项,设置其权重 $w_{ui} = 1$;对于缺失项,其权重代表其为负样本的置信度, $w_{ui} \in [0, 1)$,权重越高,其为负样本的置信度就越高。

定义用户 u 的活跃度 α_u 为该用户评分数量,即 $\alpha_u = \sum_{i=1}^n p_{ui}$ 。用户 u 对项目 i 的评分时间为 t_{ui} ,定义用户 u 的活跃时期为 $[t_{min}, t_{max}]$ 。 t_{min} 为用户 u 最早评分时间, t_{max} 为最晚评分时间。用户 u 活跃时期项目的流行度 β_i 为时间段 $[t_{min}, t_{max}]$ 项目 i 的评分数量,即 $\beta_i = \sum_{u=1}^m p_{ui}, t_{ui} \in [t_{min}, t_{max}]$ 。使用 \log 函数对 α_u 和 β_i 进行平滑处理后使用最大值归一化,如公式(2)和公式(3)所示。进行平滑处理可以平缓少数极其活跃的用户或流行项目带来的影响。

$$\hat{\alpha}_u = \frac{\log(\alpha_u)}{\max(\log(\alpha))} \quad (2)$$

$$\hat{\beta}_i = \frac{\log(\beta_i)}{\max(\log(\beta))} \quad (3)$$

对 $\hat{\alpha}_u$ 和 $\hat{\beta}_i$ 进行线性加权, 权重系数 $\varepsilon \in [0, 1]$ 。最终权重如公式(4)所示。

$$w_{ui} = \begin{cases} 1 & p_{ui} = 1 \\ \varepsilon\hat{\alpha}_u + (1 - \varepsilon)\hat{\beta}_i & p_{ui} = 0 \end{cases} \quad (4)$$

使用公式(4)构建权重矩阵 W 。利用使用前偏好矩阵 P 和权重矩阵 W 构建加权矩阵分解, 将矩阵 P 分解为两个低秩矩阵 X 和 Y , $X \in \mathbf{R}^{k \times m}$ 和 $Y \in \mathbf{R}^{k \times n}$ 分别表示用户和物品的隐因子矩阵, k 为特征数量。采用与加权矩阵分解^[14]相同的损失函数, 目标函数 J 如公式(5)所示。

$$J = \sum_{u=1}^m \sum_{i=1}^n w_{ui} (p_{ui} - \hat{r}_{ui})^2 + \lambda \left(\sum_{u=1}^m \|x_u\|^2 + \sum_{i=1}^n \|y_i\|^2 \right) \quad (5)$$

其中, x_u 和 y_i 分别为矩阵 X 的第 u 行和矩阵 Y 的第 i 行, 表示用户 u 和物品 i 的隐因子向量; $\hat{r}_{ui} = x_u y_i^T$; λ 为正则化参数。为了最小化损失函数 J , 首先随机初始化 X 和 Y , 然后使用 eALS 优化目标函数 J ^[17]。

采用 eALS 在元素级别优化参数的方式, 对公式(5)求 x_{uf} 的偏导数, 如公式(6)所示。

$$\frac{\partial J}{\partial x_{uf}} = -2 \sum_{i=1}^n (p_{ui} - \hat{r}_{ui}^f) w_{ui} y_{if} + 2x_{uf} \sum_{i=1}^n w_{ui} y_{if}^2 + 2\lambda x_{uf} \quad (6)$$

其中, $\hat{r}_{ui}^f = \hat{r}_{ui} - x_{uf} y_{if}$ 。令 $\frac{\partial J}{\partial x_{uf}} = 0$, 可得 x_{uf} 的解, 如公式(7)所示。同理, 可得 y_{if} 的解, 如公式(8)所示。

$$x_{uf} = \frac{\sum_{i=1}^n (p_{ui} - \hat{r}_{ui}^f) w_{ui} y_{if}}{\sum_{i=1}^n w_{ui} y_{if}^2 + \lambda} \quad (7)$$

$$y_{if} = \frac{\sum_{u=1}^m (p_{ui} - \hat{r}_{ui}^f) w_{ui} x_{uf}}{\sum_{u=1}^m w_{ui} x_{uf}^2 + \lambda} \quad (8)$$

迭代执行公式(7)和公式(8)直至模型收敛到局部最优。最后通过计算 $\hat{P} = XY^T$ 近似矩阵 P 。矩阵 \hat{P} 中每个元素 \hat{p}_{ui} 表示用户 u 对物品 i 的使用前偏好。 $\hat{p}_{ui} \in [0, 1]$, 越接近 1, 使用前偏好越高。

3.3 无趣项识别与填充

计算出缺失项的使用前偏好后, 就可以据此识别用户不感兴趣的项目, 称之为无趣项。本文利用

使用前偏好的百分比数值筛选无趣项。对每个用户, 将其未评分项中使用前偏好最低的 θ 个识别为该用户的无趣项。参数 θ 控制被识别为无趣项的个数。如果 θ 设置太低, 填充数量较少, 可能无法充分利用无趣项的好处; 如果 θ 设置太高, 一些用户感兴趣的项目可能也被识别为无趣项。本文将调整参数 θ 以获得最高的推荐性能。

当缺失项目被识别为无趣项时, 为其填充一个低值。因为用户不会与一个使用前偏好低的项目交互, 填充为低值可以防止其被推荐并缓解已有评分因用户选择偏差而缺乏负样本的问题。具体如何填充低值, 一个简单的方法是使用 0 填充^[9,18], 但 0 甚至超出已有评分的最低分的范围, 可能夸大了用户的负面偏好。由于不感兴趣的项目评分通常低于已有评分均值^[19], 可以通过衰减用户评分均值的方式填充无趣项。对于无趣项, 填充值 F_{ui} 计算如公式(9)所示。

$$F_{ui} = \bar{r}_u \times \sigma \quad (9)$$

其中, \bar{r}_u 表示用户 u 平均评分; $\sigma \in [0, 1]$ 用于控制无趣项与已评分项相比不感兴趣的程度, 通过调整参数 σ 可以获得最高的推荐性能。

值得注意的是, 本文的填充方法可以应用于任何协同过滤算法, 因为只是用填充后矩阵替换原始评分矩阵。换言之, 本文方法正交于所有协同过滤算法, 可以直接应用。

4 实验结果与分析

4.1 数据集和评价方法

实验使用的两个真实世界数据集 MovieLens 100k 和 MovieLens latest 是 GroupLens 项目组从 MovieLens 网站收集并公开的电影评分数据集, 其统计数据如表 1 所示。

表 1 数据集统计数据

Table 1 Description of Dataset

统计数据	MovieLens 100k	MovieLens latest
评分总数	100 000	100 836
用户数量	943	610
项目数量	1 682	9 742
稀疏度	93.7%	98.3%
评分时间跨度	1997.09-1998.04	1996.03-2018.09

对于 TopN 推荐, N 为对每个用户推荐的项目个数, 分别取值为 5、10、20。对于每个数据集, 随机选择 80% 的评分作为训练数据, 剩下 20% 作为测试数据, 所有实验结果使用 5 折交叉验证。在测试数据的评分中, 仅考虑评分大于 4 分的项目作为相关项, 因为成功推荐用户高评分的项目更具有意义^[18]。本文采用三个常用的指标衡量推荐的准确性: 精确率 (Precision@N, $P@N$)、召回率 (Recall@N, $R@N$)、归一化折扣累计收益 (Normalized Discounted Cumulative Gain@N, $NDCG@N$)。各指标计算如公式(10)–公式(12)所示。

$$P = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |R(u)|} \quad (10)$$

$$R = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |T(u)|} \quad (11)$$

$$NDCG@N = \frac{DCG@N}{IDCG@N}; DCG@N = \sum_{i=1}^N \frac{2^{rel_i} - 1}{\log_2(i + 1)} \quad (12)$$

其中, $R(u)$ 为向用户 u 推荐的一组项目; $T(u)$ 为测试集中用户 u 的相关项; $DCG@N$ 为折损信息增益, 使 $R(u)$ 中排名靠前的项目增益更高, 对排名靠后的项目进行折损; rel_i 表示 $R(u)$ 中第 i 个项目是否在 $T(u)$ 中, 若在, 则 $rel_i=1$, 否则 $rel_i=0$; $IDCG@N$ 为理想情况下的 $DCG@N$ 值。

4.2 验证用户选择偏差及其影响

首先, 对已有评分数据带有用户选择偏差的问题, 本文所使用的两个真实世界数据集清楚地反映了这一点, 评分分布如表 2 所示。

表 2 评分分布

Table 2 Rating Distribution

评分	MovieLens 100k	MovieLens latest
低评分(1和2)	17.48%	13.41%
高评分(3,4和5)	82.52%	86.59%

根据表 2 可以观察到, 已有评分数据中高评分的数量远多于低评分。为了验证其对协同过滤算法 TopN 推荐的影响, 设置了一组简单的对比填充实验。首先, 对于 MovieLens 100k 和 MovieLens latest 两个数据集, 分别为每个用户随机抽取 100 和 150 个未评分的项目。第一组使用 EMDP^[11] 算法填充这些

缺失项, 使用与文献[11]相同的参数设置。第二组使用统一低值 0 分填充这些缺失项 (Random Zero Filling, RZF), 这相当于在缺失项中随机选择无趣项, 是 UIMLF 算法最简单的一种实现。实验使用两个经典且使用广泛的协同过滤算法 ItemCF^[2] 和 BiasSVD^[22] 对比两种填充算法填充前后对 TopN 推荐的影响。实验结果如表 3 和表 4 所示。

表 3 MovieLens 100k 数据集推荐性能

Table 3 Recommendation Accuracy in MovieLens 100k

评价指标	ItemCF	Bias SVD	EMDP+ ItemCF	EMDP+ Bias SVD	RZF+ ItemCF	RZF+ Bias SVD
$P@5$	0.061 2	0.064 5	0.053 3	0.061 4	0.142 5	0.140 2
$P@10$	0.057 4	0.057 3	0.052 3	0.053 4	0.114 3	0.109 0
$P@20$	0.048 6	0.045 9	0.045 3	0.042 6	0.086 3	0.082 0
$R@5$	0.046 2	0.056 6	0.048 5	0.053 0	0.159 0	0.166 2
$R@10$	0.093 8	0.093 1	0.094 5	0.091 9	0.243 6	0.249 6
$R@20$	0.172 4	0.159 4	0.172 7	0.152 9	0.354 3	0.363 9
$NDCG@5$	0.071 9	0.077 4	0.063 0	0.076 0	0.193 8	0.189 3
$NDCG@10$	0.083 8	0.085 1	0.077 9	0.084 6	0.212 6	0.206 8
$NDCG@20$	0.108 3	0.104 8	0.103 1	0.101 9	0.246 2	0.242 0

表 4 MovieLens latest 数据集推荐性能

Table 4 Recommendation Accuracy in MovieLens latest

评价指标	ItemCF	Bias SVD	EMDP+ ItemCF	EMDP+ BiasSVD	RZF+ ItemCF	RZF+ BiasSVD
$P@5$	0.042 9	0.046 2	0.025 9	0.028 5	0.095 2	0.105 9
$P@10$	0.034 6	0.040 1	0.025 2	0.025 0	0.070 9	0.083 8
$P@20$	0.027 6	0.032 6	0.020 9	0.022 2	0.054 0	0.063 2
$R@5$	0.029 1	0.033 5	0.022 1	0.021 2	0.080 5	0.093 2
$R@10$	0.043 5	0.053 2	0.039 6	0.034 2	0.121 8	0.145 9
$R@20$	0.069 5	0.088 7	0.060 7	0.059 3	0.169 4	0.208 1
$NDCG@5$	0.051 4	0.054 6	0.035 5	0.034 4	0.117 7	0.132 1
$NDCG@10$	0.052 1	0.056 9	0.040 2	0.036 1	0.119 6	0.139 9
$NDCG@20$	0.058 5	0.065 8	0.047 1	0.043 5	0.132 8	0.157 1

根据表 3 和表 4 结果可知, 在两个数据集中, RZF 算法只是简单地用 0 随机填充一些缺失项, 缓解了已有评分数据由于用户选择偏差缺乏负样本的问题, 即能提升 TopN 推荐性能。而 EMDP 算法根据已有评分数据预测缺失项评分并填充, 效果不明显甚至更差。EMDP 算法填充的评分分布如图 1 所示。

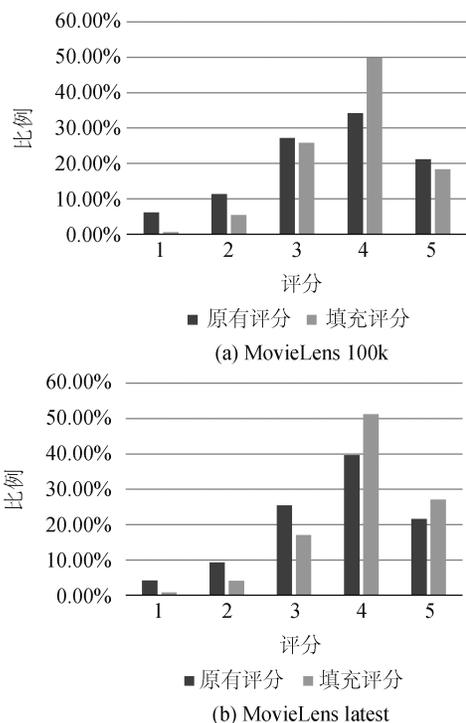


图1 EMDP填充的评分分布

Fig.1 Score Distribution of EMDP Filling

根据图1可以明显地观察到,EMDP算法为大多数缺失项填充高分。这与第2节中的分析相符,基于已有评分的填充算法往往受到用户选择偏差的影响,将大多数缺失项填充为高分。然而,这并不符合真实的评分分布,反而加剧了用户选择偏差的影响,这也可以解释它虽然缓解数据稀疏问题却反而效果不佳的原因。通过这组对比实验,充分证明了用户选择偏差对TopN推荐的影响。RZF算法作为UIMLF算法最简单的一种实现,也验证了UIMLF算法的有效性。

4.3 使用前偏好建模实验结果与分析

(1) UTV-eALS算法实验结果

使用前偏好实质是一种隐式反馈,UTV-eALS可视为一种最新的隐式矩阵分解算法。本文使用WRMF^[16]和eALS^[17]两种算法作为对比,与UTV-eALS相同,它们都是基于全数据加权的算法,以解决隐式反馈存在的单类问题。

WRMF是经典的单类协同过滤方法,将所有缺失项视为负样本,给予它们较小的统一权重。

eALS是先进的隐式矩阵分解方法,将所有缺失

项视为负样本,根据项目流行度对它们进行非均匀加权。

使用前偏好决定用户是否与项目交互,可以通过各算法计算得到的使用前偏好进行TopN推荐以衡量其有效性,推荐性能越好,说明使用前偏好计算越准确。需要说明的是,由于是隐式反馈,与4.1节设置不同,测试集所有评分都设为相关项,而不是仅评分大于4分的作为相关项。由于都为隐式矩阵分解模型,统一设置特征数量 $k=20$,正则项系数 $\lambda=0.01$ 。按照文献[16-17]的方法对WRMF和eALS参数进行调整。对于UTV-eALS算法,参数 ϵ 为用户活跃度和项目流行度的权重系数,其取值范围为0~1,以0.1为增量增加,观察其在 $P@5$ 时的推荐性能。实验结果如图2所示。

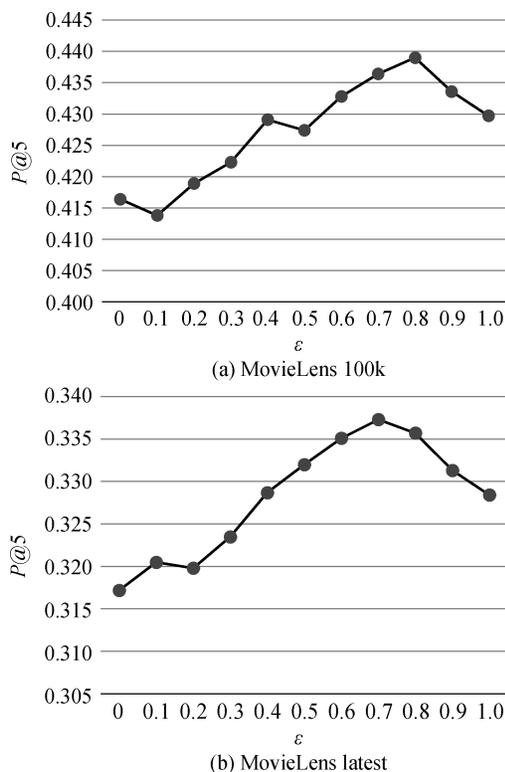


图2 参数 ϵ 的影响

Fig.2 Influence of Parameters ϵ

由图2可以看出,随着权重系数 ϵ 的增加,推荐性能总体上呈现上升趋势,分别在 $\epsilon=0.8$ 和 $\epsilon=0.7$ 左右达到最佳,然后逐渐下降。这说明对缺失数据进行加权时,相较于使用单一的用户活跃度和项目流

行度进行加权,使用适当的权重系数 ε 组合能取得更佳的性能。而且用户活跃度发挥更重要的作用,其权重比重较大时,推荐效果更好。另外,使用单一的用户活跃度相较于单一的项目流行度效果更好,这与文献[14]的结论相符。

综上,在 MovieLens 100k 数据集中,设置 $\varepsilon=0.8$; 在 MovieLens latest 数据集中,设置 $\varepsilon=0.7$ 。实验结果如表 5 和表 6 所示。

表 5 MovieLens 100k 数据集使用前偏好建模实验结果
Table 5 Experimental Results of Pre-use Preference Modeling in MovieLens 100k

评价指标	WRMF	eALS	UTV-eALS
$P@5$	0.401 7	0.418 4	0.439 0
$P@10$	0.342 1	0.344 8	0.358 7
$P@20$	0.272 0	0.268 8	0.287 1
$R@5$	0.147 1	0.149 4	0.158 2
$R@10$	0.235 2	0.231 2	0.240 4
$R@20$	0.350 2	0.340 3	0.358 0
$NDCG@5$	0.428 2	0.446 8	0.469 7
$NDCG@10$	0.419 0	0.416 3	0.434 5
$NDCG@20$	0.417 9	0.408 3	0.431 6

表 6 MovieLens latest 数据集使用前偏好建模实验结果
Table 6 Experimental Results of Pre-use Preference Modeling in MovieLens latest

评价指标	WRMF	eALS	UTV-eALS
$P@5$	0.315 9	0.319 5	0.337 3
$P@10$	0.272 0	0.270 4	0.281 4
$P@20$	0.220 6	0.218 4	0.226 4
$R@5$	0.090 3	0.093 2	0.097 0
$R@10$	0.152 1	0.151 8	0.154 4
$R@20$	0.230 2	0.232 9	0.235 2
$NDCG@5$	0.333 9	0.339 2	0.357 5
$NDCG@10$	0.314 3	0.312 9	0.328 7
$NDCG@20$	0.309 8	0.309 3	0.319 4

根据表 5 和表 6 实验结果,UTV-eALS 算法在两个数据集上都能获得最高的推荐性能。这说明结合用户活跃度与用户活跃时期的项目流行度能更准确识别“看见未交互”的项目。相较于现有加权方案,实现了对缺失项更细粒度的加权,从而更准确地推断用户对缺失项的使用前偏好,这对后续工作识别

缺失项中的无趣项非常重要。

(2) 使用前偏好分布分析

使用 UTV-eALS 算法计算出缺失项的使用前偏好后,可以根据使用前偏好的分布分析用户对缺失项的偏好倾向。使用前偏好的范围是 $[0,1]$,缺失项的使用前偏好分布如表 7 所示。

表 7 使用前偏好分布
Table 7 Pre-use Preference Distribution

使用前偏好	MovieLens 100k	MovieLens latest
$[0,0.2)$	92.64%	97.24%
$[0.2,0.4)$	5.40%	2.03%
$[0.4,0.6)$	1.48%	0.52%
$[0.6,0.8)$	0.38%	0.15%
$[0.8,1]$	0.10%	0.06%

根据表 7,用户对绝大多数缺失项使用前偏好都很低,即用户对大多数缺失项是不感兴趣的。这与第 2 节中的描述相符。同时,也可以从另一方面验证 EMDP 等基于已有评分的填充算法为大多数缺失项填充高分反而效果更差,因为这违背了用户对大多数缺失项是不感兴趣的事实,无法反映用户真实偏好。从而更有力地证明了 UIMLF 算法的有效性,因为它还原了用户真实偏好,相较于已有评分带有用户选择偏差,填充后的评分矩阵更能反映用户真实评分分布,有助于协同过滤算法更准确地进行推荐。

4.4 UIMLF 算法实验结果与分析

(1) 参数影响

使用 UTV-eALS 算法计算出缺失项的使用前偏好后,对每个用户,将其使用前偏好最低的 θ 个识别为该用户的无趣项, θ 为百分比数值。然后通过衰减用户评分均值对无趣项进行填充,参数 σ 控制衰减的程度。因此需要对 θ 和 σ 进行调参以达到最佳推荐性能。实验使用 ItemCF^[2] 和 BiasSVD^[22] 观察填充后对 TopN 推荐的影响。由于这两个参数在两个数据集上有相同的变化趋势,仅展示在 MovieLens 100k 数据集上的结果。

首先,固定填充值为 0。参数 θ 控制具体填充数量,其取值范围从 0%~100%,以 10% 为增量增加,观察其在 $N=5$ 时的推荐性能。当 θ 为 100% 时,所有缺失项都被认为是无趣项,由于本文填充策略会在推

荐候选项中排除无趣项,此时不会推荐任何项目。同时,本文定义一个错误率 $error$ 表示取参数 θ 时将测试集中的相关项识别为无趣项的概率,如公式(13)所示。

$$error = \frac{\sum_{u \in U} |B(u) \cap T(u)|}{\sum_{u \in U} |T(u)|} \quad (13)$$

其中, $B(u)$ 表示用户 u 的无趣项; $T(u)$ 表示测试集中用户 u 的相关项。实验结果如图3所示。

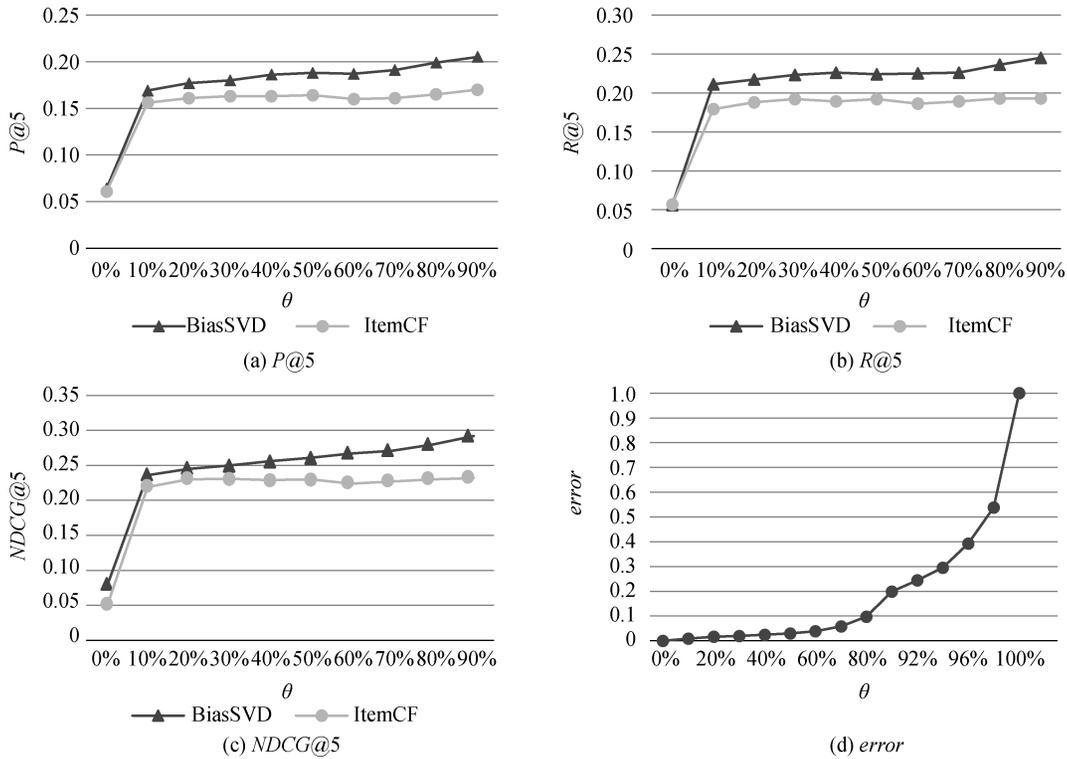


图3 参数 θ 的影响

Fig.3 Influence of Parameters θ

根据图3可知, θ 大于90%时,错误率会上升得非常快。即 θ 大于90%时,很容易将用户感兴趣的项目也识别为无趣项。 θ 在10%~90%时,精确率、召回率和归一化折扣累计收益都呈现出相似的变化趋势,总体都随着 θ 增加而平稳提高。这很好理解, θ 达到10%后变化比较平缓,此时已经填充了足够的负样本,缓解了由用户选择偏差带来的缺乏负样本的问题。随着 θ 增加,会进一步把无趣项从推荐中排除,有助于提高性能。综合考虑上述问题,设置 θ 为90%。

固定 θ 为90%。参数 $\sigma \in [0, 1]$ 用于控制无趣项与已评分项相比不感兴趣的程度。在其取值范围内,以0.1为增量增加,观察其在 $N=5$ 时的推荐性能。实验结果如图4所示。

根据图4, ItemCF 算法下,参数 σ 取0.6最佳;

BiasSVD 算法下,参数 σ 取0.3最佳。参数 σ 越大,为无趣项填充的值越大,当参数 σ 取1时,将为无趣项填充用户评分均值。当 σ 接近1时性能明显下降,这是由于此时填充值较大,已经无法有效表达用户负面偏好。但此时 θ 为90%,已经将大多数无趣项排除在推荐之外,对比表3,性能依旧优于未填充的协同过滤算法。这也证明了将无趣项排除推荐列表策略的有效性。本文所用数据集的用户全局评分均值为3.5左右,为了观察当填充值更高时对推荐结果的影响,单独列出填充值为4和5时 $P@5$ 的结果,如表8所示。

根据表8,当填充值进一步增大,推荐效果迅速下降。因为此时的填充值已经倾向于表达用户的正面偏好,填充结果已不符合用户真实的评分分布^[19],反而加剧了选择偏差的影响。本文方法对参数 σ 并

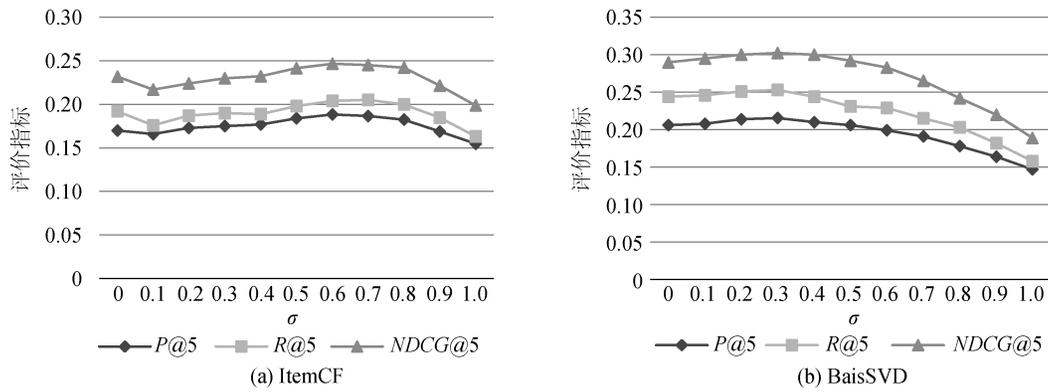


图 4 参数 σ 的影响

Fig.4 Influence of Parameters σ

表 8 高值填充结果

Table 8 Results of High Value Filling

填充评分	$P@5$	
	ItemCF	BiasSVD
4	0.092 6	0.086 8
5	0.015 2	0.023 6

不敏感,只要填充适当的低值,能够表达用户负面偏好,推荐性能变化比较平稳。同时,不同的算法对低值的具体需求并不相同。所以,使用 0 等统一的固定值^[9, 18]填充的方式很难适用于现有算法。本文通过衰减用户评分均值而填充的低分更有效。

综上所述,设置 θ 为 90%,在 ItemCF 和 BiasSVD 算法下, σ 分别设置为 0.6 和 0.3。

(2) 填充实验结果

为了验证 UIMLF 算法填充的有效性,使用

ItemCF 和 BiasSVD 对比 UIMLF 填充前后对 TopN 推荐的影响。同时,本文选取几种经典推荐算法进行比较,各算法简要说明如下。

①ItemKNN^[16]: ItemCF 面向 TopN 推荐最流行的变种,也是目前流行的基线方法。与 ItemCF 通过预测所有缺失项评分再排序进行推荐不同,它计算用户对已评分项目的相似项目的偏好而非评分进行排序。本文使用与文献[16]相同的设置。

②PureSVD^[18]:使用 0 填充所有缺失项,然后进行传统的奇异值分解。

③UTV-eALS:本文提出的基于用户时点可见性的加权矩阵分解算法,用于计算缺失项使用前偏好。由于展现了优于 WRMF 和 eALS 的性能,作为隐式矩阵分解算法对比以上基于显式评分的算法。

实验结果如表 9 和表 10 所示。

表 9 MovieLens 100k 数据集数据填充实验结果

Table 9 Experimental Results of Data Filling in MovieLens 100k

评价指标	ItemCF	BiasSVD	UIMLF+ItemCF	UIMLF+BiasSVD	ItemKNN	PureSVD	UTV-eALS
$P@5$	0.061 2	0.064 5	0.188 3	0.213 4	0.142 3	0.186 5	0.178 6
$P@10$	0.057 4	0.057 3	0.145 6	0.161 6	0.118 3	0.144 0	0.139 2
$P@20$	0.048 6	0.045 9	0.108 4	0.114 5	0.089 1	0.104 8	0.103 1
$R@5$	0.046 2	0.056 6	0.204 0	0.248 5	0.166 9	0.215 5	0.209 1
$R@10$	0.093 8	0.093 1	0.316 4	0.345 4	0.268 6	0.315 6	0.310 4
$R@20$	0.172 4	0.159 4	0.460 1	0.480 2	0.387 8	0.440 7	0.441 5
$NDCG@5$	0.071 9	0.077 4	0.246 7	0.303 7	0.193 4	0.258 4	0.247 2
$NDCG@10$	0.083 8	0.085 1	0.267 8	0.319 5	0.219 9	0.276 7	0.270 0
$NDCG@20$	0.108 3	0.104 8	0.311 9	0.358 1	0.256 8	0.314 6	0.310 8

表 10 MovieLens latest 数据集数据填充实验结果

Table 10 Experimental Results of Data Filling in MovieLens latest

评价指标	ItemCF	BiasSVD	UIMLF+ ItemCF	UIMLF+ BiasSVD	ItemKNN	PureSVD	UTV-eALS
<i>P@5</i>	0.042 9	0.046 2	0.152 2	0.168 5	0.116 7	0.150 3	0.144 8
<i>P@10</i>	0.034 6	0.040 1	0.115 9	0.125 1	0.091 9	0.117 4	0.114 4
<i>P@20</i>	0.027 6	0.032 6	0.091 6	0.094 2	0.071 1	0.088 4	0.087 6
<i>R@5</i>	0.029 1	0.033 5	0.113 6	0.131 7	0.105 7	0.122 1	0.118 9
<i>R@10</i>	0.043 5	0.053 2	0.171 4	0.191 5	0.153 8	0.182 4	0.179 7
<i>R@20</i>	0.069 5	0.088 7	0.262 6	0.282 1	0.228 6	0.255 4	0.268 0
<i>NDCG@5</i>	0.051 4	0.054 6	0.183 9	0.204 5	0.150 8	0.189 5	0.182 0
<i>NDCG@10</i>	0.052 1	0.056 9	0.183 0	0.202 8	0.148 2	0.191 3	0.183 3
<i>NDCG@20</i>	0.058 5	0.065 8	0.206 2	0.224 8	0.168 0	0.207 6	0.205 4

仅考虑已有评分数据, ItemCF 和 BiasSVD 的表现并不理想。经过 UIMLF 算法填充后, ItemCF 和 BiasSVD 的推荐精度显著提升, 平均提升了 2~2.5 倍。这充分表明了有效利用缺失数据隐含的用户负面偏好对推荐的积极作用。ItemKNN 为用户推荐与其已评分项目相似度高的项目, 通过计算非规范化的偏好来排序, 避免了使用预测评分排序的偏差, 表现出比 ItemCF 更好的性能, 是目前流行的基线方法。但其仅考虑了已评分数据, 而 UIMLF 填充后的 ItemCF 充分利用了缺失数据, 缓解了用户选择偏差, 推荐精度较 ItemKNN 更高。

对比 PureSVD 和表 3、表 4 中的 RZF, UIMLF 也显示出更高的推荐性能。因为 PureSVD 只是简单地将所有缺失项都作为无趣项, RZF 从缺失项随机选择无趣项, 它们都没有真正识别缺失项中的无趣项, 这表明 UIMLF 准确识别无趣项的有效性。

作为基于隐式反馈的推荐算法之一, UTV-eALS 由于同时考虑了观察到的数据和缺失数据, 推荐精度显著优于 ItemCF 和 BiasSVD。但由于隐式反馈对用户偏好表达并不明确且无法表达用户偏好程度, 其在推荐高分项的性能同样不如 UIMLF 填充后的两个经典协同过滤算法。这表明相较于隐式反馈方法推荐项目仅考虑到使用前偏好, 推荐的项目用户虽然会交互, 但可能实际交互后并不喜欢, 即使用后偏好不高。而使用 UIMLF 填充后显式反馈方法同时考虑到使用前偏好和使用后偏好, 能够更有效地推荐用户既会交互且实际使用后会评高分的项目, 这在实际应用中更具有意义。

5 结 语

本文针对基于显式反馈的协同过滤算法仅考虑已有的评分数据难以有效进行 TopN 推荐的问题, 分析并论证了数据稀疏性和用户选择偏差对推荐的影响。并针对这个问题, 提出通用的数据填充算法 UIMLF, 充分利用缺失项隐含的负面偏好。为挖掘缺失项中的无趣项, 提出使用前偏好的概念。根据“看见未交互”的项目表现用户负面偏好的经验, 结合用户活跃度、项目流行度和时间因素衡量用户时点可见性, 提出 UTV-eALS 算法对缺失项的使用前偏好建模。将使用前偏好低的项目作为无趣项并以低值填充。实验证明, UIMLF 算法能够在仅利用评分数据而不使用任何辅助信息的情况下, 有效提升协同过滤算法的推荐性能。然而, UTV-eALS 算法仅依据“看见未交互”的项目表现负面偏好这一经验对使用前偏好建模, 可能存在经验偏差。使用前偏好往往取决于项目的外在特征, 下一步将考虑融入丰富的辅助信息实现对使用前偏好更精确的建模。最后, 对无趣项的低值填充采用的是较简单的方法, 不够个性化。后续工作将深入研究使用前偏好和使用后偏好之间的联系, 实现更精确的填充。

参考文献:

- [1] Lu J, Wu D S, Mao M S, et al. Recommender System Application Developments: A Survey[J]. Decision Support Systems, 2015, 74: 12-32.
- [2] Sarwar B, Karypis G, Konstan J, et al. Item-Based Collaborative Filtering Recommendation Algorithms [C]//Proceedings of the

- 10th International Conference on World Wide Web. 2001: 285-295.
- [3] Marlin B M, Zemel R S. Collaborative Prediction and Ranking with Non-random Missing Data[C]//Proceedings of the 3rd ACM Conference on Recommender Systems. 2009: 5-12.
- [4] Marlin B M, Zemel R S, Roweis S, et al. Collaborative Filtering and the Missing at Random Assumption [C]//Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence. 2007: 267-275.
- [5] Jawaheer G, Szomszor M, Kostkova P. Comparison of Implicit and Explicit Feedback from an Online Music Recommendation Service[C]//Proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems. 2010: 47-51.
- [6] da Silva J F G, de Moura Junior N N, Caloba L P. Effects of Data Sparsity on Recommender Systems Based on Collaborative Filtering[C]//Proceedings of 2018 International Joint Conference on Neural Networks. 2018: 1-8.
- [7] Chen J W, Dong H D, Wang X, et al. Bias and Debias in Recommender System: A Survey and Future Directions[OL]. arXiv Preprint, arXiv: 2010.03240.
- [8] Dalvi N, Kumar R, Pang B. Para ‘Normal’ Activity: On the Distribution of Average Ratings[C]//Proceedings of the 7th International AAAI Conference on Weblogs and Social Media. 2013: 110-119.
- [9] Hwang W S, Parc J, Kim S W, et al. “Told You I Didn’t Like It”: Exploiting Uninteresting Items for Effective Collaborative Filtering[C]//Proceedings of 2016 IEEE 32nd International Conference on Data Engineering. 2016: 349-360.
- [10] Steck H. Evaluation of Recommendations: Rating-Prediction and Ranking[C]//Proceedings of the 7th ACM Conference on Recommender Systems. 2013: 213-220.
- [11] Ma H, King I, Lyu M R. Effective Missing Data Prediction for Collaborative Filtering[C]//Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. 2007: 39-46.
- [12] Ren Y L, Li G, Zhang J, et al. The Efficient Imputation Method for Neighborhood-Based Collaborative Filtering[C]//Proceedings of the 21st ACM International Conference on Information and Knowledge Management. 2012: 684-693.
- [13] Chae D K, Kang J S, Kim S W, et al. Rating Augmentation with Generative Adversarial Networks Towards Accurate Collaborative Filtering[C]//Proceedings of the World Wide Web Conference. 2019: 2616-2622.
- [14] Pan R, Zhou Y H, Cao B, et al. One-Class Collaborative Filtering [C]//Proceedings of 2008 8th IEEE International Conference on Data Mining. 2008: 502-511.
- [15] Rendle S, Freudenthaler C, Gantner Z, et al. BPR: Bayesian Personalized Ranking from Implicit Feedback[C]//Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence. 2009: 452-461.
- [16] Hu Y F, Koren Y, Volinsky C. Collaborative Filtering for Implicit Feedback Datasets[C]//Proceedings of 2008 8th IEEE International Conference on Data Mining. 2008: 263-272.
- [17] He X N, Zhang H W, Kan M Y, et al. Fast Matrix Factorization for Online Recommendation with Implicit Feedback[C]//Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2016: 549-558.
- [18] Cremonesi P, Koren Y, Turrin R. Performance of Recommender Algorithms on Top-N Recommendation Tasks[C]//Proceedings of the 4th ACM Conference on Recommender Systems. 2010: 39-46.
- [19] Steck H. Training and Testing of Recommender Systems on Data Missing not at Random[C]//Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2010: 713-722.
- [20] Liang D W, Charlin L, McInerney J, et al. Modeling User Exposure in Recommendation[C]//Proceedings of the 25th International Conference on World Wide Web. 2016: 549-558.
- [21] He X N, Gao M, Kan M Y, et al. Predicting the Popularity of Web 2.0 Items Based on User Comments [C]//Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval. 2014: 233-242.
- [22] Koren Y, Bell R, Volinsky C. Matrix Factorization Techniques for Recommender Systems[J]. Computer, 2009, 42(8): 30-37.

作者贡献声明:

石磊:进行实验,清洗、分析数据,起草论文;
李树青:提出研究思路,设计研究方案,论文最终版本修订。

利益冲突声明:

所有作者声明不存在利益冲突关系。

支撑数据:

支撑数据[1-2]为 MovieLens 数据集;支撑数据[3-4]由作者自存储, E-mail: 1176466796@qq.com。

[1] ml-100k. MovieLens 100k 数据集 . <https://files.grouplens.org/datasets/movielens/ml-100k.zip>.

[2] ml-latest-small. MovieLens latest 数据集 . <https://files.grouplens.org/datasets/movielens/ml-latest-small.zip>.

[3] 石磊 . pre-use-preference.txt. 使用前偏好矩阵。

[4] 石磊 . filling-rating.csv. 填充评分。

收稿日期:2021-08-12

收修改稿日期:2022-02-14

Mining Uninteresting Items with Visibility of User Time Points and Collaborative Filtering Recommendation Method

Shi Lei Li Shuqing

(College of Information Engineering, Nanjing University of Finance & Economics, Nanjing 210023, China)

Abstract: [Objective] This paper proposes a new method to improve the collaborative filtering algorithm based on explicit feedbacks, aiming to address data sparsity and user selection bias issues. [Methods] First, we retrieved the negative preferences of users who have seen the items but did not interact with them. Then, we measured the visibility of items along with user activity, item popularity and time factors. Third, we introduced the concept of pre-use preferences to construct a weighted matrix factorization model based on user time point visibility. Finally, we identified items that users were not interested in, and marked them with low values. [Results] We examined our model with the MovieLens datasets, and found the recommendation accuracy of ItemCF and BiasSVD increased by an average of 2 to 2.5 times. [Limitations] There may be empirical bias in modeling pre-use preferences based on the users' negative preferences from the "seen-but-not-interacted items". [Conclusions] The proposed model could effectively reduce the impacts of data sparsity and user selection bias, and make accurate recommendation results.

Keywords: Collaborative Filtering Explicit Feedback Selection Bias Pre-use Preference Uninteresting Items

研究为STEM组织如何更有效地收集人口统计信息提供建议

近期,发表在 *Science* 期刊上的一项研究为科学、技术、工程和数学 (Science, Technology, Engineering and Mathematics, STEM) 领域的专业组织如何更有效地收集有关其领域代表性不足群体的数据提供了方法。有了更可靠的信息,STEM组织可以更好地针对招募和留住多样化的成员而努力。

研究人员共调查了美国 164 个 STEM 组织,这些组织大多来自隶属于美国科学促进会的某一协会。研究人员询问了这些组织所收集的成员和会议参与者的人口统计信息的情况,以及他们如何使用这些信息。73 个组织对调查做出回应,覆盖了从生命科学、物理科学到数学、技术等各个领域的超过 700 000 名成员。虽然大多数组织(80%)都收集了一些人口统计数据,但他们所收集的内容各不相同,例如,许多组织会遵循联邦机构使用的分类方法,为“种族和民族”这一统计项提供多种选择,但也存在将几个不同的群体归为一类(例如“亚裔美国人和太平洋岛民”)的情况。

此外,研究人员还提出了几种指导调查设计的模型。STEM组织可能需要将自己的调查数据与全国收集的数据进行基准比较,例如来自美国国家科学基金会的博士学位获得者调查、美国人口普查局的社区调查和疾病控制与预防中心的全国健康访谈调查。但是,这些调查在收集的信息类别上也有所不同。例如,美国国家科学基金会的调查使用单一的“亚洲人”类别标识身份,而人口普查局的调查和全国健康访谈调查则标识了 7 种不同的亚洲身份。

大多数受访者(87.5%)表示,他们收集人口统计数据是出于某些特定目的,最常见的是用于监控和资源规划。研究人员希望这项研究能够引发 STEM 社区的广泛讨论、进一步的研究和行动。

(编译自:<https://www.sciencedaily.com/releases/2022/03/220331151457.htm>)

(本刊讯)