

融合时间因素和用户评分特性的协同过滤算法

曾安 高成思 徐小强

(广东工业大学计算机学院 广州 510006)

摘要 针对传统协同过滤技术在现实应用中遇到的数据稀疏性问题和局限性,充分挖掘用户评分特性,提出融合时间因素和用户评分特性的协同过滤算法(CF-TP)。引入用户偏好模型,将用户-项目评分矩阵转化为用户-项目偏好得分矩阵,以降低用户评分习惯差异带来的影响。在预测用户对项目的偏好得分时,充分考虑用户之间的非对称影响度,根据用户兴趣随时间的变化引入时间权重函数,以提高 top-N 推荐的准确率。基于 HetRec2011 和 MovieLens1M 数据集的实验结果表明,相对于目前比较流行的算法,所提算法在推荐结果的准确率、召回率、F1 值上均有较大的提升,有效提高了推荐系统的推荐质量。

关键词 推荐系统,协同过滤,时间因素,用户偏好,非对称性

中图分类号 TP311 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.09.046

Collaborative Filtering Algorithm Incorporating Time Factor and User Preference Properties

ZENG An GAO Cheng-si XU Xiao-qiang

(School of Computer, Guangdong University of Technology, Guangzhou 510006, China)

Abstract To alleviate the impact of data sparsity and overcome the limits of traditional collaborative filtering algorithm, a novel collaborative filtering algorithm(CF-TP) by incorporating time factor and user preference properties was proposed. A user-item rating matrix is firstly converted to a user-item preference matrix by introducing a preference model, and this helps to reduce the impact of different users' rating habits. Then the asymmetric impact between users is taken into consideration when computing the predicted scores. What's more important, to further improve the accuracy of top-N recommendation, the time weight function considering users' dynamic interest changed with time is designed. The experiment results on HetRec2011 and MovieLens1M data set suggest that the proposed algorithm is superior to other advanced approaches in precision, recall and F1.

Keywords Recommender system, Collaborative filtering, Time factor, User preference, Asymmetric

1 引言

在当今互联网信息过载的时代,面对海量的信息,用户经常需要消耗大量的时间和精力来选择一个物品^[1]。为了解决信息过载问题,目前主要有两大主流研究方向:搜索引擎和推荐系统。搜索引擎解决了目标明确的信息检索,但很多时候用户并不清楚自己的需求或者无法准确描述自己的需求,此时搜索引擎无法很好地解决用户需求问题,而推荐系统可以利用数据挖掘等技术,为用户提供个性化信息推荐^[2]。

目前,主流的推荐系统主要分为4类:基于内容的推荐、协同过滤推荐、基于知识的推荐和组合推荐^[3]。其中,协同过滤算法无需用户或项目的属性信息,只根据相似用户或项目

的评分特性即可产生推荐,并可发现用户的潜在信息需求^[4],在不同的应用中具有较强的适应性,从而得到广泛的应用。

当前,很多研究注重于将传统的协同过滤算法与其他模型相结合,形成优势互补,从而提高推荐质量。黄创光等人^[5]提出了一种不确定近邻的动态度量方法,在不确定的应用场景中结合基于用户和项目的推荐结果,通过不确定近邻因子及调和参数来计算基于用户和项目的预测评分,从而生成推荐;该算法不仅一定程度地提升了准确率,还有效平衡了基于用户和项目推荐结果所带来的不稳定影响,缓解了用户评分数据的稀疏性问题。Parivash P 等人^[6]在计算用户相似度时,考虑到不同用户评分物品的数量不同,将传统计算方法得到的对称相似度转化为非对称相似度,并加权考虑用户对每

到稿日期:2016-08-27 返修日期:2016-11-28 本文受国家自然科学基金项目(61300107),广东省自然科学基金项目(S2012010010212),广州市科技计划项目(201504301341059),广东省科技计划项目(2014B090901053)资助。

曾安(1978-),女,博士,教授,主要研究方向为智能信息处理、数据挖掘,E-mail:zengan201@126.com;高成思(1991-),男,硕士生,主要研究方向为数据挖掘、推荐系统,E-mail:chancygao@163.com(通信作者);徐小强(1987-),男,硕士生,主要研究方向为数据挖掘、推荐系统,E-mail:xxqcheers0614@163.com。

个评分的评分频率,提高了推荐的准确度。Jongwuk L 等人^[7]通过考虑用户的评分分布情况,建立用户偏好模型,将用户对项目的评分转化为用户对项目的偏好得分,从而将传统的用户-项目评分矩阵转化为用户-项目偏好得分矩阵,然后运用传统的推荐算法生成推荐;该方法在推荐准确率和召回率上均有一定的提升。

但是,大多数协同过滤算法只利用了用户对项目的历史评分来生成推荐^[8]。在这种情况下,用户的兴趣偏好被认为是静态的,即无论用户历史评分的产生时间有何不同,其在推荐中所起的作用是相等的。而现实生活中,用户的兴趣或信息需求往往随着时间的推移发生着剧烈或缓慢的变化,不同的历史评分对描述用户当前兴趣的能力也有着较大的差异,进而产生用户兴趣概念漂移问题^[9]。与其他上下文信息(如地点、天气、心情等)不同,时间信息已不需要额外的用户或设备资源,标准的推荐系统中已经包含了用户对项目的评分时间。因此,目前越来越多的学者开始关注用户兴趣概念漂移问题,在推荐系统中考虑时间因素。Xia C 等人^[10]介绍了时间衰减概念,并提出了一种融合时间衰减函数的计算物品相似度的方法,一定程度上提升了基于项目的协同过滤推荐的精确度。Bakir K 等人^[11]在传统协同过滤算法的基础上,基于群体用户偏好随时间的改变,在预测评分时添加时间权重函数来提高协同推荐的精确度。Zhang Y C 等人^[12]将用户的历史评分划分为几个时期,分析用户在各个时期的兴趣分布,并设置时间窗来找到用户的最近兴趣。这些算法在融合时间因素后均在一定程度上提升了推荐精确度,但它们并未对用户-项目评分矩阵进行深入挖掘,没有充分考虑用户的评分特性,推荐精确仍有进一步的提升空间。

因此,本文针对数据稀疏性问题,充分挖掘评分矩阵信息,使用用户偏好模型,考虑用户之间的非对称影响度,并构建了一个时间权重函数,提出了融合时间因素和用户评分特性的协同过滤算法(Collaborative Filtering Algorithm Incorporating Time Factor and User Preference Properties, CF-TP)。在 HetRec2011 和 MovieLens 数据集上的实验结果表明,本文提出的算法在推荐结果的准确率、召回率和 F1 值上均有较大的提升。

2 协同过滤算法描述

通常情况下,推荐系统包括 m 个用户集合 $U = \{u_1, u_2, \dots, u_m\}$, n 个项目集合 $I = \{i_1, i_2, \dots, i_n\}$ 。下面以基于用户的协同过滤为例,详细介绍其推荐流程。

步骤 1 构建评分矩阵。将用户集合 U 对项目集合 I 的评分数据转化为用户-项目评分矩阵 $R(m, n)$, 其中 r_{ui} 表示用户 u 对物品 i 的评分。评分值与内容有关,可以是若干等级的评分,如 0~5, 也可以是 0 或 1, 表示用户是否有某种行为,如浏览、加入购物车、购买等。

步骤 2 根据评分矩阵计算用户之间的相似度,找到近邻用户。常见的相似度计算方法有余弦相似度、Pearson 相关系数、修正的余弦相似度等。

其中,采用 Pearson 相关系数计算用户 u 和用户 v 的相似度 $sim(u, v)$ 的公式如式(1)所示^[11]:

$$sim(u, v) = \frac{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u)^2} \sqrt{\sum_{i \in I_{uv}} (r_{vi} - \bar{r}_v)^2}} \quad (1)$$

其中, I_{uv} 表示用户 u 和 v 共同评分的项目集合, r_{ui} 和 r_{vi} 分别表示用户 u 和 v 对项目 i 的评分, \bar{r}_u 和 \bar{r}_v 分别表示用户 u 和 v 对项目的平均评分。

步骤 3 为目标用户生成推荐。

(1) 获取近邻用户 v 有评分、但目标用户 u 没有评分的项目作为候选项目,计算目标用户对候选项目 i 的预测评分 p_{ui} , 如式(2)所示^[11]:

$$p_{ui} = \bar{r}_u + \frac{\sum_{v \in U_n} sim(u, v) \times (r_{vi} - \bar{r}_v)}{\sum_{v \in U_n} sim(u, v)} \quad (2)$$

其中, \bar{r}_u 和 \bar{r}_v 分别表示目标用户 u 和近邻用户 v 的平均分, U_n 表示目标用户的近邻用户集。

(2) Top- N 形式的推荐项目集。获取目标用户 u 对候选项目预测评分最高的前 N 个项目作为推荐项目集,并将其推荐给目标用户 u 。

观察分析发现,在进行 Top- N 推荐时,没有必要预测准确的评分值,关键是区别出目标用户 u 对候选项目的感兴趣程度,然后对候选项目进行排序。因此,在计算预测评分时,可忽略式(2)中用来评分标准化的分母,如式(3)所示^[13]:

$$p_{ui} = \bar{r}_u + \sum_{v \in U_n} sim(u, v) \times (r_{vi} - \bar{r}_v) \quad (3)$$

此时, p_{ui} 并不代表用户对候选项目的预测评分,而代表用户对候选项目的偏好程度。

3 融合时间因素和用户评分特性的协同过滤算法

融合时间因素和用户评分特性的协同过滤算法的基本思路是:首先,根据不同用户的评分习惯,使用用户偏好模型,将用户对项目的显性评分转化为用户对项目的隐性偏好得分,进而将用户-项目评分矩阵转化为用户-项目偏好得分矩阵,以降低用户评分习惯所带来的影响;其次,考虑不同用户之间的特异性,在预测目标用户对候选物品的偏好得分时,引入用户之间的非对称影响度,以降低数据稀疏性影响;然后,考虑用户兴趣随时间的动态变化,引入时间权重函数,减少用户评分时间差异性带来的影响;最后,将偏好得分最高的前 N 个项目推荐给目标用户。

为了能更清晰地解释文中后续提到的用户偏好模型、用户之间的非对称影响度,以表 1 为例进行说明。表 1 中共有 4 个用户和 8 个项目评分,且都采用 5 分制,即评分值为 1~5, 0 表示用户未评分的项目。

表 1 用户-项目评分矩阵

	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8
u_1	1	1	2	3	4	5	5	5
u_2	1	2	3	3	4	4	4	5
u_3	2	0	0	3	0	0	0	0
u_4	1	2	0	3	4	3	4	0

3.1 用户偏好模型

在现实生活中,用户的评分习惯有较大的差异^[7,14]。例如,用户 u 习惯给很多项目评分 5,用户 v 习惯给很多项目评分 1,当他们同时对一个物品 i 评分 5 时,由于用户 u 习惯高分,从而很难判断用户 u 对物品 i 的偏好程度。相比之下,由于用户 v 很少给项目评分 5,因此可以认为用户 v 比用户 u 更加喜欢项目 i 。

为了降低用户评分习惯带来的影响,提出了一种用户偏好模型,将用户对项目的显性评分转化为用户对项目的隐性偏好得分。设定 $\{C_1, \dots, C_k\}$ 表示数据集中的用户评分类别,并且 $C_i > C_j$ 表示类别 C_i 中的评分大于类别 C_j 中的评分。如评分范围为 $\{1, 2, 3, 4, 5\}$,则评分类别为 $\{C_1, C_2, C_3, C_4, C_5\}, C_2 > C_1$ 。

定义 1 给定用户 u 及用户 u 的评分集合 R_u ,则用户 u 对评分类别 C_i 所表现的偏好得分通过统计用户评分小于或等于类别 C_i 的评分数量并加权计算,如式(4)所示^[7]:

$$pref(C_i) = \alpha \cdot \sum_{C_j \in \{C_1, \dots, C_k\}} \frac{pref_{>}(C_i, C_j)}{|R_u|} + \beta \cdot \frac{pref_{=}(C_i)}{|R_u|} \quad (4)$$

其中, $|R_u|$ 表示用户 u 的评分数量, $pref_{>}(C_i, C_j)$ 表示用户评分在类别 C_j (类别 C_i 大于类别 C_j) 中的评分数量。 $pref_{=}(C_i)$ 表示用户评分在类别 C_i 中的评分数量。根据文献^[7, 14]的分析,设定参数 $\alpha=1, \beta=0.5$ 。

以表 1 中用户 u_1, u_2 的评分为例,根据式(4)计算用户 u_1, u_2 对各评分类别的偏好得分,如表 2 所列。表 3 列出了用户 u_1, u_2 评分转化后的用户-项目偏好得分矩阵。在表 1 中,用户 u_1, u_2 对项目 I_8 的评分均为 5,且用户 u_1, u_2 的平均评分值相同,若采用传统平均值中心化的方法,则 u_1, u_2 对项目 I_8 的偏好是一样的。但引入用户偏好模型将用户评分转化为偏好得分后,如表 3 所列,用户 u_1 对项目 I_8 的偏好得分为 0.81, u_2 对项目 I_8 的偏好得分为 0.94 分,因此,可以认为用户 u_2 比 u_1 更喜欢项目 I_8 ,从而说明引入用户偏好模型可以减少用户评分习惯差异所带来的影响。

表 2 用户 u_1, u_2 对各评分类别的偏好得分

	C_1	C_2	C_3	C_4	C_5
u_1	0.13	0.31	0.44	0.56	0.81
u_2	0.06	0.19	0.38	0.69	0.94

表 3 u_1, u_2 用户-项目偏好得分矩阵

	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8
u_1	0.13	0.13	0.31	0.44	0.56	0.81	0.81	0.81
u_2	0.06	0.19	0.38	0.38	0.69	0.69	0.69	0.94

引入用户偏好模型后,在计算用户之间的相似度时,先将用户-项目评分矩阵根据式(4)转化为用户-项目偏好得分矩阵,然后根据 Pearson 相关系数计算公式(1)来计算用户之间的相似度 $sim(u, v)$ 。在进行 Top- N 推荐评分预测时,将式(3)中的 $r_{vi} - \bar{r}_v$ 替换为 $pref(r_{vi})$,则改变后的预测偏好得分的方法如式(5)所示^[10]:

$$p_u = \sum_{v \in U_n} sim(u, v) \times pref(r_{vi}) \quad (5)$$

其中, $pref(r_{vi})$ 表示近邻用户 v 对项目 i 的偏好得分; $sim(u, v)$ 表示用户 u, v 的相似度; U_n 表示目标用户 u 的近邻用户集。

3.2 用户间的非对称影响度

考虑到不同用户的特异性,即用户之间的相互影响程度不同^[6],如:生活中独立性较强的人和依赖性较强的人在一起时,他们的相互影响程度是不同的,为了体现出不同用户之间相互影响程度的差异性,算法引入了用户间的非对称影响度。

定义 2 给定用户 u 和用户 v ,以及用户 u 和 v 的评分项目集合 I_u 和 I_v ,则用户 v 对用户 u 的非对称影响度如式(6)所示^[6]:

$$impact(u, v) = 1 - \exp\left(-\frac{|I_u \cap I_v|}{|I_u|}\right) \quad (6)$$

以表 1 中用户 u_3 和用户 u_4 为例,根据式(6)计算可得,用户 u_4 对 u_3 的影响度 $impact(u_3, u_4) = 1 - \exp(-1) = 0.632$,用户 u_3 对 u_4 的影响度 $impact(u_4, u_3) = 1 - \exp(-1/3) = 0.283$,由于用户 u_3 的评分项目少,用户 u_4 的评分项目多,则 $impact(u_3, u_4) > impact(u_4, u_3)$ 。这说明用户评分的项目越多,他对其他用户的影响就越大,其他用户对他的影响越小。

3.3 时间权重函数

由于用户的兴趣往往随时间的推移发生变化,用户最近的评分更能体现出其当前兴趣,因此一个用户在不同时间的评分对推荐系统产生的作用不是相等的,用户最近的评分相对于以前的评分应被赋予更高的权重^[9-11,15]。根据以上分析,提出了时间权重函数,如式(7)所示^[9]:

$$W(t(r_{u,i})) = \exp[-(1/\lambda) \cdot (\tau_0 - t(r_{u,i}))] \quad (7)$$

其中, $t(r_{u,i})$ 表示用户 u 对项目 i 的评分时间, τ_0 表示训练集中用户评分的最大时间, λ 表示时间权重参数。从式(7)可以看出,评分时间权重函数的取值范围为 $0 \sim 1$,用户评分时间距当前时间越远,评分时间权重越小,且参数越小,时间权重随时间衰减得越快。

3.4 预测偏好得分

为了充分考虑用户评分特性和时间因素的影响,根据上文分析,在进行预测偏好得分和 Top- N 推荐生成时,在式(5)的基础上融合了用户间非对称影响度和时间权重函数,最终预测偏好得分公式如式(8)所示:

$$p_{ui} = \sum_{v \in U_n} sim(u, v) \times pref(r_{vi}) \times impact(u, v) \times w(t(r_{v,i})) \quad (8)$$

其中, $sim(u, v), pref(r_{vi}), impact(u, v), w(t(r_{v,i}))$ 分别根据式(1)、式(4)、式(6)、式(7)计算得到。

3.5 算法详细步骤

融合时间因素和用户评分特性的协同过滤算法的具体步骤如下。

输入: 目标用户 u , 用户-项目评分矩阵 $R(m, n)$, 用户近邻个数 N_{neig} , top- N 推荐的项目个数 N_{top}
 输出: 推荐给目标用户 u 的 N_{top} 个项目集 I_{top}

1. 根据式(4),基于用户-项目评分矩阵 $R(m, n)$ 计算每个用户对各评分类别 $\{C_1, C_2, C_3, C_4, C_5\}$ 的偏好得分,将用户-项目评分矩阵转化为用户-项目偏好得分矩阵 $Pref(m, n)$;
2. 基于步骤 1 计算得到的 $Pref(m, n)$,根据式(1)计算出用户 u 与其他用户 v 的相似度 $sim(u, v)$;
3. 根据步骤 2 计算的用户相似度,选取与目标用户 u 最相似的前 N_{neig} 个用户作为近邻用户集 U_{neig} ;
4. 根据式(6)计算 U_{neig} 中近邻用户 v 对用户 u 的影响度 $impact(u, v)$;
5. 获取 U_{neig} 中近邻用户 v 有评分但目标用户 u 没有评分的所有项目作为候选项目集 I_{neig} ;
6. 根据式(7),计算近邻用户 v 对候选项目集中候选项目 i 的评分时间权重 $w(t(r_{v,i}))$;
7. 基于步骤 1 计算得到的 $Pref(m, n)$,根据式(8)和步骤 2、步骤 4、步骤 6 计算的结果,计算出目标用户 u 对 I_{neig} 中所有候选项目的预测偏好得分;
8. 从 I_{neig} 中取偏好得分最高的前 N_{top} 个项目,生成 top-N 推荐列表,并将其推荐给目标用户 u .

4 评估方法

4.1 数据集

本文在进行实验时采用了 HetRec2011¹⁾ 和 MovieLens1M²⁾ 两个真实数据集。MovieLens 是美国明尼苏达大学的 Group Lens 研究小组提供的数据集,该数据集广泛用于验证推荐算法的推荐性能。HetRec2011 数据集是在 MovieLens10M 数据集上延伸出来的一个版本,它包含了 2113 个用户对 10197 部电影的 855598 个评分,评分范围为 1~5,评分时间从 1997-09-18 到 2009-01-05。MovieLens1M 数据集包含了 6040 个用户对 3652 部电影的 1000209 个评分,评分范围为 1~5,评分时间从 2000-04-26 到 2003-03-01。

为了验证融入用户评分时间特性对推荐算法性能的提升,参考文献[9]将两个原始数据集分别做以下调整:对于 HetRec2011 数据集,首先将所有评分按照评分时间递增排序,抽取从 2008-01-05 到 2009-01-05 近一年的评分数据作为子数据集,然后取前 80% 的数据作为训练集,后 20% 的数据作为测试集;对于 MovieLens1M 数据集,首先将所有评分按照评分时间递增排序,通过分析发现,该数据集评分范围为 2000-04-06 到 2003-03-01,大概是 3 年的评分周期,但 94.1% 的评分集中在第一年(2000-04-26 到 2001-04-26),共 941079 条评分记录,因此抽取 2000-04-26 到 2001-04-26 一年的评分数据作为子数据集,然后取前 80% 的数据作为训练集,后 20% 的数据作为测试集。

进一步对抽取的数据集进行分析发现,测试集中许多参与评分的测试用户在训练集中完全没有参与评分,测试集中许多被评分的测试项目在训练集中完全没有评分,协同过滤算法不能为这些新用户产生推荐,或者给用户推荐这些新项目,这就是典型的用户冷启动和项目冷启动问题^[16]。为了减少冷启动问题的影响,更好地验证本文提出的 CF-TP 算法的性能,从测试集中移除了这些新用户和新项目。调整后的最

终数据集统计信息如表 4 所列。

表 4 最终数据集统计信息

数据集	HetRec2011		MovieLens1M	
	训练集	测试集	训练集	测试集
开始时间	2008-01-05	2008-11-05	2000-04-26	2000-11-26
结束时间	2008-11-05	2009-01-05	2000-11-26	2001-04-26
评分数量	112930	9650	752863	60846
用户数量	1112	511	5146	912
项目数量	7088	3194	3681	3279

4.2 评价指标

本文采用目前主流的评价指标准确率(Precision)和召回率(Recall)来评价推荐算法的性能。准确率是指正确推荐项目数量与所有推荐项目数量的比值,它表示用户对系统推荐项目感兴趣的概率,衡量了对用户的推荐效果;召回率表示一个用户喜欢的项目被推荐的概率,定义为推荐列表中用户喜欢的项目与系统中用户喜欢的所有项目的比例,它主要是对系统级别的衡量。

推荐系统的准确率为:

$$Precision = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |R(u)|} \quad (9)$$

推荐系统的召回率为:

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

其中, $R(u)$ 表示用户 u 生成的推荐项目集合, $T(u)$ 表示测试集中用户 u 有评分行为的项目集合, U 为测试集中所有用户集合。

为了同时考虑准确率和召回率,以比较全面地评价算法的优劣,本文还采用 F1 指标作为算法性能的评测指标,其计算公式如下:

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \quad (11)$$

F1 的值越大,表示推荐算法的综合性能越好。

5 实验结果分析

为了验证提出的融合时间因素和用户评分特性的协同过滤算法 CF-TP 的性能,将其与文献[6]提出的加权相似度协同过滤算法(Pearson Correlation Coefficient with both Accordance and Compromise factors, AC-PCC)、文献[7]提出的使用用户偏好模型的协同过滤算法(prefUserKNN)、文献[9]提出的融合时间因素的协同过滤算法(Collaborative Filtering with Time-dependant Function, CF-TD)以及传统的基于用户的协同过滤算法 UBCF 在 HetRec2011 和 MovieLens1M 数据集上进行比较。

首先,通过设置不同的时间衰减权重参数来测试它对 CF-TP 算法的性能影响,图 1 和图 2 分别示出了不同时间衰减系数对 CF-TP 算法的准确率、召回率和 F1 在 HetRec2011 和 MovieLens1M 数据集上的影响。

¹⁾ <http://grouplens.org/datasets/hetrec-2011>

²⁾ <http://grouplens.org/datasets/movielens>

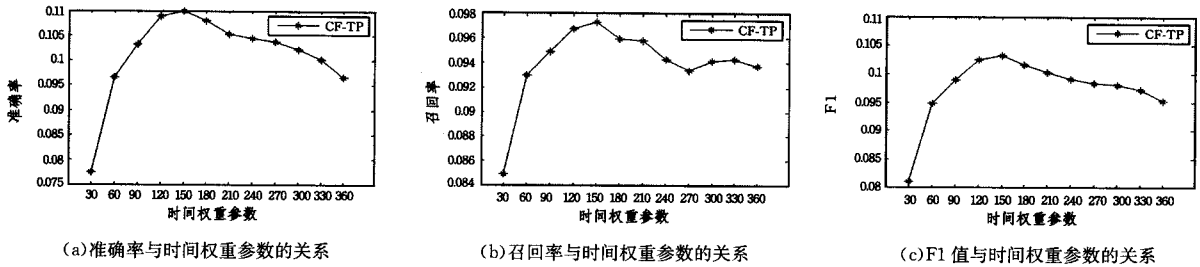


图 1 HetRec2011 数据集中不同时间权重参数对准确率、召回率、F1 值的影响

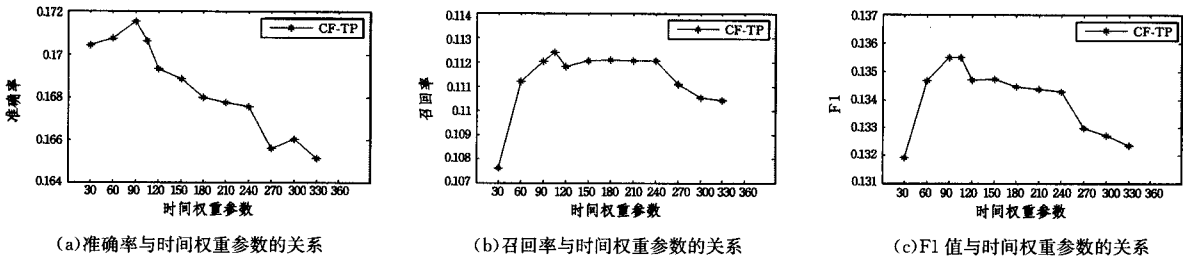


图 2 MovieLens1M 数据集上不同时间权重参数对准确率、召回率、F1 值的影响

从图 1 和图 2 可以看出,在 HetRec2011 数据集和 MovieLens1M 数据集上,本文提出的 CF-TP 算法的准确率、召回率、F1 指标均随着时间权重参数的增大先增加再逐渐减少。通过实验发现,推荐结果在时间权重参数为 $2/\Delta T$ 时达到最优,与文献[11]的实验分析结果一致,这里 ΔT 代表训练集中所有评分的时间周期。例如在 HetRec2011 数据集中,训练集时间周期为 306 天,各指标在时间权重参数为 150 左右时达到最优。在 MovieLens1M 数据集中,训练时间周期为 215

天,各指标在时间权重参数为 105 左右时达到最优。因此,在接下来的所有实验中,对于 HetRec2011 数据集的实验,将时间权重参数设置为 150;对于 MovieLens1M 数据集的实验,将时间权重参数设置为 105。

图 3 和图 4 分别给出在 HetRec2011 和 MovieLens1M 数据集上,当推荐项目个数一定且用户邻居数分别从 10 依次递增到 100 时,对本文提出的 CF-TP 算法与 CF-TD、PrefUserKNN、传统的 UBCF 算法、AC-PCC 的准确率、召回率及 F1 的影响。

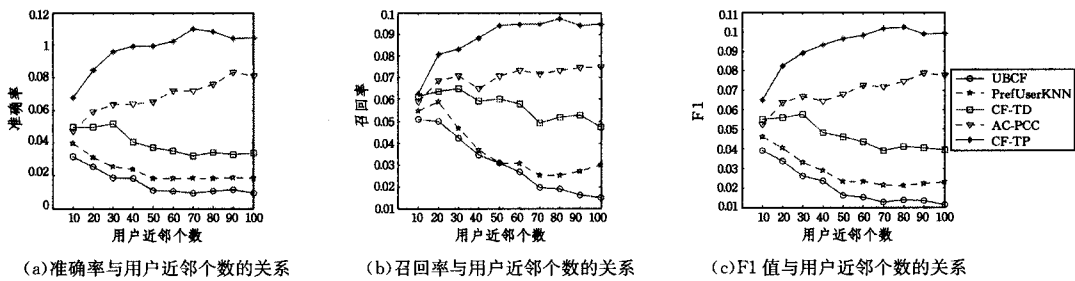


图 3 HetRec2011 数据集中不同近邻个数对准确率、召回率、F1 值的影响

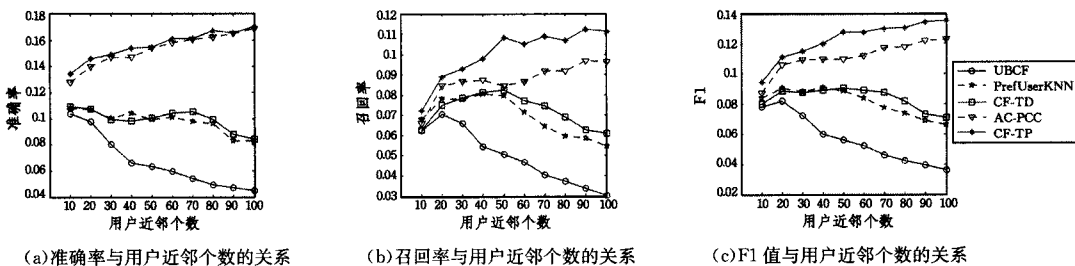


图 4 MovieLens1M 数据集上不同近邻个数对准确率、召回率、F1 值的影响

从图 3 和图 4 可以看出,在 HetRec2011 数据集和 MovieLens1M 数据集上,当近邻个数相同时,本文提出的 CF-TP 算法相比于其他算法,具有最高的准确率、召回率和 F1 值。并且当随着用户近邻个数的增加,UBCF、PrefUserKNN、CF-TD 算法的准确率、召回率、F1 值逐渐降低时,CF-TP 算法的各指标仍处于平稳状态,甚至有轻微的提升。这说明 CF-TP 算法在融合时间因素和用户评分特性后,通过使用用户偏好模

型,引用用户间非对称影响度和时间衰减函数,可以有效降低随着近邻个数增加所带来的数据稀疏性的影响,提高推荐质量。

从图 3 和图 4 可以看出,近邻数量对算法推荐结果具有一定的影响。因此,为了验证推荐项目数对各算法推荐性能的影响,本文固定用户近邻个数,进一步的实验结果如图 5 和图 6 所示。

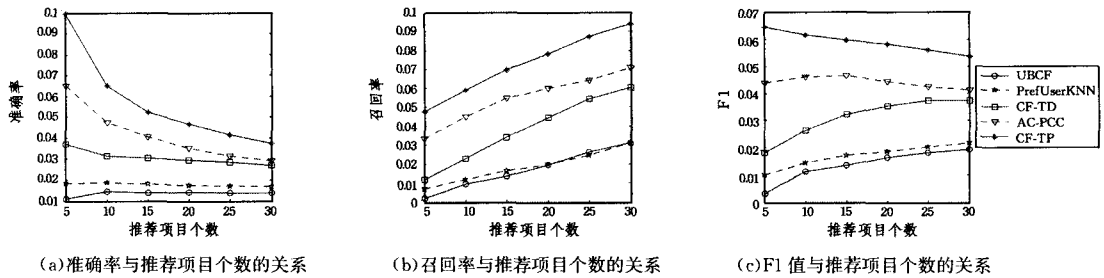


图5 HetRec2011数据集中不同推荐项目个数对准确率、召回率、F1值的影响

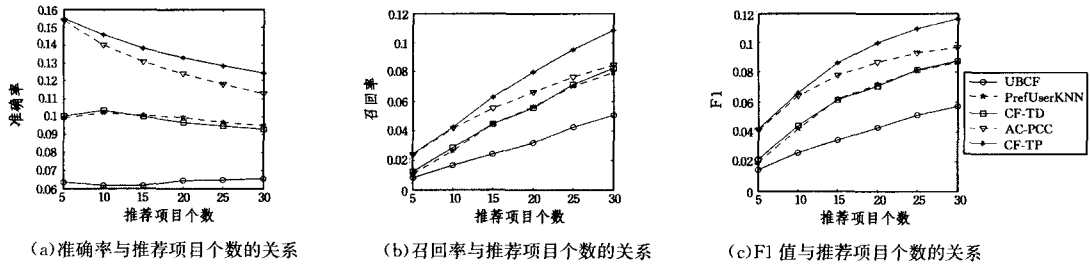


图6 MovieLens1M数据集中不同推荐项目个数对准确率、召回率、F1值的影响

图5和图6分别表示在 HetRec2011 和 MovieLens1M 数据集上,当用户近邻个数统一设定为 50,且推荐项目分别从 5 依次递增到 30 时,对各个算法的准确率、召回率及 F1 的影响。从图 5 和图 6 可以看出,当用户近邻个数一定时,在不同的推荐项目个数的情况下,本文提出的 CF-TP 算法与其他算法相比均有最高的准确率、召回率和 F1 值。并且随着推荐项目的增加,各算法的准确率均逐渐减小,召回率逐渐增大。

表 5 详细列出了在 HetRec2011 和 MovieLens1M 数据集上各算法的准确率、召回率、F1 值的最优值。

表 5 在不同数据集中各算法性能的比较

算法	HetRec2011			MovieLens1M		
	准确率	召回率	F1	准确率	召回率	F1
CF-TP	0.1100	0.0974	0.1024	0.1706	0.1124	0.1348
AC-PCC	0.0830	0.0749	0.0784	0.1686	0.0966	0.1225
CF-TD	0.0517	0.0650	0.0576	0.1094	0.0823	0.0904
PrefUserKNN	0.0395	0.0587	0.0459	0.1083	0.0805	0.0909
UBCF	0.0317	0.0509	0.0391	0.1037	0.0705	0.0819

从表 5 可以看出,本文提出的 CF-TP 算法在 HetRec2011 数据集上的准确率、召回率和 F1 值相对于 AC-PCC 分别提高了 32.53%, 30.04%, 30.61%, 相对于 CF-TD 分别提高了 112.77%, 49.85%, 77.78%; 在 MovieLens1M 数据集上相对于 AC-PCC 分别提高了 1.19%, 16.36%, 10.04%, 相对于 CF-TD 分别提升了 55.95%, 36.57%, 49.12%。实验结果表明,本文算法融合了用户评分时间因素、用户之间的非对称影响度和用户偏好模型的综合效果,能够从整体上提升算法的性能,从而提高推荐质量。

结束语 本文提出了一种融合时间因素和用户评分特性的协同过滤算法。在传统基于用户的协同过滤的基础上,充分挖掘用户评分特性,引入了用户偏好模型。在进行偏好得分预测时,考虑了用户之间的非对称影响度并引入了时间权重函数。通过在 HetRec2011 和 MovieLens1M 数据集上的实验结果表明,本文提出的算法能有效提高推荐系统的准确率、召回率和 F1 值。并且,这种考虑时间因素和用户偏好模型的思想是通用的,能够适用于其他的推荐算法。该算法的不足之处在于

未考虑冷启动问题,下一步的工作将考虑如何在冷启动的情况下提高推荐系统的性能。

参考文献

- [1] KHUSRO S, ALI Z, ULLAH I. Recommender Systems: Issues, Challenges, and Research Opportunities [M] // Information Science and Applications (ICISA) 2016. Springer Singapore, 2016.
- [2] KARATZOGLOU A, HIDASI, Bal, et al. RecSys'16 Workshop on Deep Learning for Recommender Systems (DLRS) [C] // ACM Conference on Recommender Systems. ACM, 2016: 415-416.
- [3] MENG X W, LIU S D, ZHANG Y J, et al. Research on Social Recommender Systems [J]. Journal of Software, 2015, 26 (6): 1356-1372. (in Chinese)
孟祥武, 刘树栋, 张玉洁, 等. 社会化推荐系统研究 [J]. 软件学报, 2015, 26(6): 1356-1372.
- [4] VAIRACHILAI S, KAVITHADEVI M K, RAJA M. Alleviating the Cold Start Problem in Recommender Systems Based on Modularity Maximization Community Detection Algorithm [J]. Circuits and Systems, 2016, 7(8): 1268-1279.
- [5] HUANG C G, YIN J, WANG J, et al. Uncertain Neighbors' Collaborative Filtering Recommendation Algorithm [J]. Chinese Journal of Computers, 2010, 33(8): 1369-1377. (in Chinese)
黄创光, 印鉴, 汪静, 等. 不确定近邻的协同过滤推荐算法 [J]. 计算机学报, 2010, 33(8): 1369-1377.
- [6] PIRASTEH P, HWANG D, JUNG J E. Weighted Similarity Schemes for High Scalability in User-Based Collaborative Filtering [J]. Mobile Networks & Applications, 2015, 20(4): 497-507.
- [7] LEE J, LEE D, LEE Y C, et al. Improving the accuracy of top-N recommendation using a preference model [J]. Information Sciences, 2016, 348(c): 290-304.
- [8] ADOMAVICIUS G, TUZHILIN A. Context-Aware Recommender Systems [C] // Recommender Systems Handbook Springer US. 2011: 217-253.
- [9] KARAHODZA B, DONKO D. Feature enhanced time-aware re-

commender system[C]//Xxv International Conference on Information, Communication and Automation Technologies. 2015;1-6.

[10] XIA C,JIANG X,LIU S,et al. Dynamic item-based recommendation algorithm with time decay[C]//International Conference on Natural Computation(Icnc 2010). Yantai, Shandong, China, 2010;242-247.

[11] KARAHODZA B,SUPIC H,DONKO D. An approach to design of time-aware recommender system based on changes in group user's preferences[C]// X International Symposium on Telecommunications. 2014;1-4.

[12] ZHANG Y C, CHEN C. A Collaborative Filtering Algorithm Based on Time Period Partition[C]//Third International Symposium on Intelligent Information Technology and Security Informatics. IEEE Computer Society,2010;777-780.

[13] CREMONESI P,KOREN Y,TURRIN R. Performance of recommender algorithms on top-n recommendation tasks[C]// ACM Conference on Recommender Systems(Recsys 2010). Barcelona, Spain,2010;39-46.

[14] JIN R,SI L,ZHAI C X,et al. Collaborative filtering with decoupled models for preferences and ratings[C]// ACM CIKM International Conference on Information and Knowledge Management. New Orleans, Louisiana, Usa,2003;309-316.

[15] ADIBI P,LADANI B T. A collaborative filtering recommender system based on user's time pattern activity[C]// Information and Knowledge Technology. 2013;252-257.

[16] BOBADILLA J,ORTEGA F,HERNANDO A,et al. A collaborative filtering approach to mitigate the new user cold start problem[J]. Knowledge-Based Systems,2012,26;225-238.

(上接第 229 页)

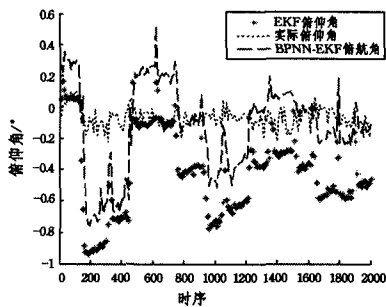


图 3 预测俯仰角与实际角度的对比曲线

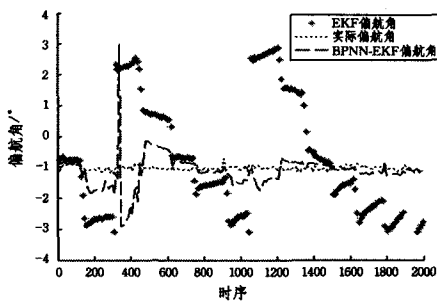


图 4 预测偏航角与实际角度的对比曲线

结束语 本文将神经网络的非线性映射能力运用到扩展卡尔曼滤波算法中,提出了一种改进的扩展卡尔曼滤波算法,并将该算法应用于无人机姿态解算中,经过神经网络输出值的误差补偿,改进后的扩展卡尔曼滤波器各个维度的预测值的误差分别降低了 41.86%,27.27%,6.98%,91.36%,88.31%,16.66%,79.50%,92.29%,95.57%;由式(14)计算得到姿态角 φ, θ, ψ 的解算精度分别提升了 41.63%,22.88%,64.33%。仿真结果证明,改进的 EKF 算法是一种非常有效的姿态解算算法。

参 考 文 献

[1] LIU M M, QIN P L, CHANG J, et al. New improved algorithm based on EKF information theory [J]. Application Research of Computers, 2015, 32(5): 1569-1571. (in Chinese)

刘毛毛,秦品乐,常江,等.基于多新息理论的 EKF 改进算法[J].计算机应用研究,2015,32(5):1569-1571.

[2] JIANG E S, LI M C, SUN L J. Improved Kalman filter method based on neural network [J]. Journal of Electronics & Information Technology, 2007, 29(9): 2073-2076. (in Chinese)

蒋恩松,李孟超,孙刘杰.一种基于神经网络的卡尔曼滤波改进方法[J].电子与信息学报,2007,29(9):2073-2076.

[3] GURUNG H, BANERJEE A. Self-sensing SMA Actuator Using Extended Kalman Filter and Artificial Neural Network[J]. Procedia Engineering, 2016, 144(25): 629-634.

[4] HAJIYEV C, CILDEN D, SOMOV Y. Gyro-free attitude and rate estimation for a small satellite using SVD and EKF [J]. Aerospace Science and Technology, 2016, 74(8): 325-328.

[5] ZHAO Y W. Performance evaluation of Cubature Kalman filter in a GPS/IMU tightly-coupled navigation system [J]. Signal Processing, 2016, 119(21): 178-182.

[6] HE Y, GUAN X, WANG G H. multi-sensor data fusion research progress and Prospect [J]. Journal of Astronautics, 2005, 26(4): 524-529. (in Chinese)

何友,关欣,王国宏.多传感器信息融合研究进展与展望[J].宇航学报,2005,26(4):524-529.

[7] 秦永元,张洪钺,汪叔华.卡尔曼滤波与组合导航原理(第三版)[M].西安:西北工业大学出版社,2015:199-200.

[8] TANG L. Neural Networks and Kalman filter algorithm [D]. Harbin: Harbin Institute of Technology, 2007. (in Chinese)

唐蕾.神经网络与卡尔曼滤波算法的研究[D].哈尔滨:哈尔滨工业大学,2007.

[9] GREWAL M S, ANDREWS A P. Kalman Filtering: Theory and Practice Using Matlab (3rd edition) [M]. Hoboken: Wiley & Sons, Inc, 2008.

[10] LI Z L, CHEN W B, ZOU H J, et al. Adaptive Neural Network Information Kalman filter [J]. Hunan University of Technology, 2011, 25(1): 105-108. (in Chinese)

李忠良,陈卫兵,邹豪杰,等.基于信息的神经网络自适应卡尔曼滤波[J].湖南工业大学学报,2011,25(1):105-108.

[11] JIANG C P. Design and Realization of a four-rotor UAV control system [D]. Harbin: Harbin Institute of Technology, 2014. (in Chinese)

姜成平.一种四旋翼无人机控制系统的设计与实现研究[D].哈尔滨:哈尔滨工业大学,2014.