计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.200300107

基于排序学习的软件众包任务推荐算法

余敦辉[1,2]　成　涛[1]　袁　旭[1]
1 湖北大学计算机与信息工程学院　武汉 430062
2 湖北省教育信息化工程技术中心　武汉 430062
(yumhy@163.com)

摘　要　为了更有效地实现软件众包任务推荐,提升软件开发质量,为工人推荐合适的任务,降低工人利益受损风险,以达到工人和众包平台双赢的效果,设计了一种基于排序学习的软件众包任务推荐方法。首先,基于改进的隐语义模型提取工人-任务间的隐含特征;然后,结合隐式信息对排序学习模型进行改进,并将提取的隐含特征进行排序学习训练,获得最优排序模型;最终通过排序模型对测试集任务进行排序得到任务推荐列表,从而为工人进行众包任务推荐,并采用 NDCG,MAP,Recall 推荐评价指标对推荐结果进行检验。实验表明,所设计的方法能有效提高软件众包任务推荐的精度,其推荐评价指标的 *NDCG*,*MAP*,*Recall* 值分别达到 0.722,0.326,0.169。与基于用户的协同过滤算法相比,推荐精度提升了 18.6%;与仅基于 RankNet 的排序学习算法相比,精度提升了 10.2%,因此能够有效指导软件众包任务推荐。

关键词:软件众包;任务推荐;隐语义模型;隐式反馈;排序学习
中图法分类号　TP311.52

Software Crowdsourcing Task Recommendation Algorithm Based on Learning to Rank

YU Dun-hui[1,2],CHENG Tao[1] and YUAN Xu[1]
1 College of Computer and Information Engineering,Hubei University,Wuhan 430062,China
2 Education Informationalization Engineering and Technology Center,Wuhan 430062,China

Abstract　In order to realize software crowdsourcing task recommendation more effectively,improve the quality of software development,recommend suitable tasks for workers,reduce the risk of workers' interests being damaged,and achieve a win-win result for workers and crowdsourcing platforms,a software crowdsourcing task recommendation method based on learning to rank is designed. First,the hidden features between workers and tasks are extracted based on the improved latent factor model. Then,the model of learning to rank is improved by combining implicit information,and the extracted hidden features are ranked and trained to obtain the optimal ranking model. The ranking model sorts the test set tasks to get a task recommendation list to perform crowdsourcing task recommendation for workers,and uses relevant evaluation indicators to verify the recommendation results. Experiments show that the proposed method can effectively improve the software crowdsourcing task recommendation accuracy. The *NDCG*,*MAP*,and *Recall* values of the recommended evaluation indicators reach 0.722,0.326,0.169,respectively. Compared with the user-based collaborative filtering algorithm,the recommendation accuracy is improved by 18.6%. Compared with rank learning algorithm based on RankNet only,the accuracy is improved by 10.2%,which can effectively guide software crowdsourcing task recommendation.

Keywords　Software crowdsourcing,Task recommendation,Latent factor model,Implicit feedback,Learning to rank

随着互联网技术的快速发展以及网络用户规模的爆发式增长,众包[1]这种通过群体智慧来解决问题的新兴模式应运而生。软件开发也不再局限于小型、孤立的开发者社区。相反,越来越多的人正在使用众包平台来竞争和完成软件开发任务。这种跨越时间和地域限制的新兴开发方式已经成为一种强有力的软件工程新方法——软件众包[2]。近年来,以 Topcoder 和开源众包平台为代表的、采用在线竞争机制寻找优秀的开发者来完成整个软件开发的众包应用正日益为人们所接受。

然而,现有软件众包平台存在以下几点不足:1)平台只提

到稿日期:2020-03-18　返修日期:2020-07-25　本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。
基金项目:湖北省技术创新重大专项(2018ACA13),国家自然科学基金(61572371,61832014)
This work was supported by the Technology Innovation Special Program of Hubei Province(2018ACA13) and National Natural Science Foundation of China(61572371,61832014).
通信作者:成涛(1475793186@qq.com)

供任务列表，排在列表后面的任务很难被工人发现和关注，而且工人通常选择最近发布的任务来参与竞标，导致有些任务不能在有效的时间内完成；2）众包平台在为任务分配工人时，通常是在所有竞标工人完成任务后，选择完成效果最好的工人进行分配，这种做法虽然确保了任务的完成效果，但是未充分考虑工人的自身利益。为此，必须设计一种有效的任务推荐方法，以便能够筛选出相对合适的工人来完成任务，进而提升软件开发质量，这是众包平台亟待解决的一项极具价值和挑战性的工作。

当前，学术界对于软件众包任务推荐问题展开了积极的研究，一些研究人员从历史信息入手挖掘工人和任务的联系，从而为工人进行推荐。例如，Karim 等[3]基于众包工作者的偏好和技能，提出一种任务推荐系统，以便工作者选择合适的任务，但并未考虑众包平台方的利益；Ambati 等[4]利用任务和工人的历史信息构建工人的技能兴趣模型，从而为工人推荐其感兴趣的任务，该方法仅考虑了工人的技能属性，而未考虑到工人其他各方面的属性；Yuen 等[5]将工人的历史信息转化为评级，以此构建工人-任务矩阵，并提出了基于概率矩阵分解的任务推荐框架来实现任务的推荐，但其并未考虑工人任务的隐式反馈信息。一些研究人员额外考虑了隐式信息，如 Liu 等[6]在显示信息的基础上考虑了隐式反馈信息，设计了一种引入隐式反馈信息的多维度推荐算法来对用户进行推荐，但缺乏对数据量大时推荐准确性的考虑；Zhu 等[7]从历史任务中提取开发者的特征，并基于主题特征的排序方法对工人能力进行排序来完成开发人员的推荐，但并未考虑任务特征；Fu 等[8]借助可扩展的元模型来描述工人的技能水平，提出了一种自适应的众包任务-工人匹配方法，从而为任务选择符合要求的工人，但该方法缺乏对任务技能权重的考虑；Li 等[9]根据开发人员的历史行为构建社交网络，提出一种基于社交网络的方法，为活跃和不活跃的开发者推荐合适的任务，但缺乏对工人偏好和任务效用的考虑。一些研究人员在传统推荐算法的基础上进行任务推荐，如 Zhong 等[10]基于协同过滤推荐思想，提出考虑用户兴趣和能力的众包任务推荐方法，根据融合了兴趣和能力的综合相似度选取近邻工人以生成任务推荐，但只考虑了工人的属性，并未考虑任务的属性；Mao 等[11]采用基于内容的推荐技术来自动匹配任务和开发人员，从历史任务信息中挖掘兴趣偏好，来推荐合适的开发者，但该方法是基于任务的属性相似度来进行推荐，并未考虑工人与任务的匹配程度；Shao 等[12]提出软了件众包任务特征模型，结合了神经网络和基于内容的方法来推荐开发人员，但该方法过分依赖静态属性，未考虑工人与任务间的交互信息。

上述研究虽然对众包任务推荐做出了一定的贡献，但大多仅考虑工人或者任务单方面的特征，未整合工人和任务的特征，且没有充分利用众包平台中工人与任务间的隐式反馈信息，导致推荐准确率不足且解释性不强。

为了弥补上述研究存在的不足，本文提出一种基于排序学习的软件众包任务推荐算法（Task Recommendation of Software crowdsource based on Learning to rank，TRSL）。该算法综合考虑工人和任务各方面的显式特征和隐式特征，基于排序学习 RankNet 算法构建适合工人的任务模型，同时针对 RankNet 未考虑单个任务的匹配度大小问题，引入隐语义模型对 RankNet 训练集数据进行处理，通过隐语义模型与 RankNet 算法相结合的方式来解决 RankNet 对单个任务的匹配度预测错误的问题，从而提升了 RankNet 排序模型的预测准确率。

本文的主要贡献有以下 3 点：

（1）为解决 RankNet 对单个任务的匹配度预测错误问题，引入隐语义模型对训练集数据进行预处理，将基于隐语义模型提取的特征数据作为 RankNet 模型的输入，通过隐语义模型与 RankNet 排序学习相结合的方式，来提升 RankNet 排序模型的性能。

（2）针对隐语义模型正样本较少、无负样本且对隐语义模型性能影响较大的问题，基于工人隐式反馈增加正样本，并且选取部分负样本，调整正负样本比例使其保持均衡。隐语义模型通过矩阵相乘提取特征，易导致提取特征精度不足，鉴于此本文基于余弦相似度计算匹配度。以上两点改进使推荐结果更准确，隐语义模型性能更优。

（3）针对 RankNet 真实匹配性概率粒度较大的问题，采用隐语义模型得出的任务对匹配度进行计算，作为 RankNet 真实匹配性概率，并结合基于众包环境下的工人行为与任务效用来改进 RankNet 真实匹配性概率，以提高排序模型的性能，提升 RankNet 的预测准确率。

1 系统整体解决方案

1.1 相关定义

定义 1（软件众包工人） 众包工人指完成软件众包任务的开发人员，设为 U_m，工人集合为 $U=\{u_1,u_2,\cdots,u_m\}$。

定义 2（软件众包任务） 软件众包任务是众包平台发布的众包任务，其集合作为训练集中的任务集合，为工人集合完成的历史任务集合，设为 I_n，任务集合为 $I=\{i_1,i_2,\cdots,i_n\}$。

定义 3（工人特征） 工人特征是从工人历史完成任务、工人注册静态属性、工人行为等信息中提取的工人特征，设为 F_u，$F_u=\{f'_{u1},f'_{u2},\cdots,f'_{uc}\}$。其中包括显式特征和隐式特征，显式特征为技能集合、学历、岗位等特征，隐式特征为工人参与竞标行为、工人点击、工人浏览等特征。

定义 4（任务特征） 任务特征为软件众包任务中的特征，包括技能要求、任务回报、完成时限等。任务特征设为 F_i，$F_i=\{f'_{i1},f'_{i2},\cdots,f'_{ic}\}$，其中包括显式特征和隐式特征，显式特征为任务的技能要求、学历要求、岗位要求等，隐式特征为任务紧急度、回报率等。

定义 5（工人-任务特征） 工人-任务特征集是联系工人与任务的相关特征，是工人特征和任务特征中相同的特征，即工人特征和任务特征的交集。设工人-任务特征集为 $F=F_u\cap F_i=\{\boldsymbol{f}_1,\boldsymbol{f}_2,\cdots,\boldsymbol{f}_k\}$。

1.2 解决方案

传统的推荐方法如协同过滤算法已经不能满足实际需求，而排序学习算法已得到深入研究并被广泛应用于推荐系统中。RankNet 是一种重要的排序学习方法，具有良好的性

能。本文基于 RankNet 排序学习算法对软件众包任务形成推荐。然而 RankNet 排序学习算法也存在缺陷，因此本文引入隐语义模型进行改进。本文提出了一种基于隐语义模型和 RankNet 排序学习的软件众包任务推荐方法，其主要解决方案如下。

(1)设工人集合 $U=\{u_1,u_2,\cdots,u_m\}$，任务集合 $I=\{i_1,i_2,\cdots,i_n\}$，将其基于隐式反馈信息筛选工人-任务正负样本对，其匹配度为 $md(u_m,i_n)=\{0,1\}$。

(2)筛选后的工人-任务正负样本对通过改进后的隐语义模型提取工人-任务对(u_m,i_n)的特征，形成工人-任务对特征集 $F=\{\boldsymbol{f}(u_m,i_n)\mid m=1,2,\cdots,M,n=1,2,\cdots,N\}$。

(3)将提取出的工人-任务特征集中的任务特征向量集与匹配度结合，做进一步处理后形成用于 RankNet 排序模型的数据集，然后将数据集按照 4∶1 的比例划分为训练集和测试集。

(4)应用改进的 RankNet 排序学习算法对训练集数据进行模型训练，得到排序模型。

(5)基于排序模型为工人 U_m 在测试任务集合 I_n' 中对每个任务进行排序，将 Top-k 个任务作为推荐子集推荐给工人，如图 1 所示。

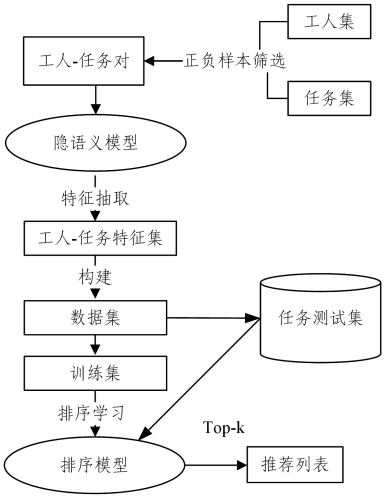


图 1 基于排序模型的推荐框架

Fig. 1 Recommendation framework based on ranking model

2 基于隐语义模型的特征提取

2.1 隐语义模型的初始化

在众包环境下，工人与任务之间只有简单的行为信息以及各自的基本属性，而排序学习需要工人-任务对的特征向量作为训练数据。隐语义模型(Latent Factor Model，LFM)为近年来机器学习领域中特征提取的常用方法之一[13]，因此本文采用隐语义模型来提取工人和任务的隐含特征，以描述工人与任务之间的内在联系。其特点在于：通过隐含特征来联系工人和任务，使用降维的矩阵因式分解方法提取工人-任务特征向量。将工人-任务匹配度矩阵分解为工人-隐含特征矩阵和任务-隐含特征矩阵，如图 2 所示。该模型主要解决图 1 中从样本筛选到数据集的前半部分问题，然后将工人与任务间匹配度为 1 的数据初始化到隐语义模型中的 ***MD*** 矩阵中。

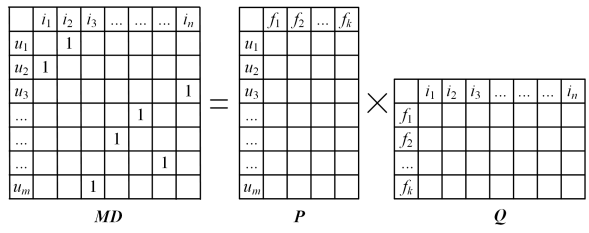


图 2 隐语义矩阵分解模型

Fig. 2 Latent factor model matrix factorization

图 2 中，***MD*** 矩阵为工人-任务匹配度矩阵。***P*** 矩阵为工人-隐含特征矩阵，对于任一工人 u，满足 $\boldsymbol{p}_u=\sum_{f=1}^{k}\boldsymbol{p}_{uf}=1$。***Q*** 矩阵为任务-隐含特征矩阵，对于任一任务 i，满足 $\boldsymbol{q}_i=\sum_{f=1}^{k}\boldsymbol{q}_{fi}=1$。其中 $\boldsymbol{f}_k\in F$，是联系工人任务的特征，即工人和任务均具有的特征。

2.2 基于隐式反馈的矩阵填充

软件众包领域中，软件众包任务一般由众包平台分配或者工人自主竞标获得，因此 ***MD*** 矩阵稀疏性较大，而当数据集非常稀疏时，隐语义模型的性能会明显下降。但在众包平台中有大量工人与任务间的隐式反馈信息，其中包括工人参与过任务的竞标行为。基于工人行为的考虑，工人参与过竞标的任务与工人的匹配度为 1，即正样本值为 1。如对于任一工人 $u\in U$，在任务集$\{i_1,i_2,\cdots,i_n\}$中，工人 u 对任务 i 有参与竞标行为，则工人-任务对(u,k)为正样本，其正样本值md_{ui}为 1。

文献[14]的研究表明，正负样本比例 $radio=1$ 时，其对隐语义模型的影响最小，则对于任一工人 u，其正样本就是历史完成任务与参与竞标的任务，其数量为 N，负样本则参照文献[15]的方法从任务集中选取 n 个任务，其负样本值md_{ui}为 0，满足 $N=n$，然后将正负样本填充到 ***MD*** 矩阵，降低矩阵稀疏性，提升模型性能，最后通过将正负样本代入隐语义模型中来提取工人任务特征。

2.3 基于余弦相似度的特征提取

基于图 2 的工人-任务匹配度矩阵分解，工人 u 与任务 i 的预测匹配度$\hat{md}_{ui}$为：

$$\hat{md}_{ui}=\boldsymbol{p}_u\boldsymbol{q}_i=\sum_{f=1}^{k}\boldsymbol{p}_{uf}\boldsymbol{q}_{fi} \tag{1}$$

MD 矩阵中的匹配度，即任一工人 u 与任务 i 之间的匹配度即为工人 u 的特征向量和任务 i 的特征向量的乘积。但是通过这种方式提取的特征向量精确度不够高，而向量间相似度通常采用余弦相似度来表达。因此参照余弦相似度计算公式，得到改进后的工人 u 与任务 i 之间的匹配度计算方式(见式(2))，以提高隐语义模型特征向量的提取精度。

$$\hat{md}_{ui}=\frac{\boldsymbol{p}_u\boldsymbol{q}_i}{\|\boldsymbol{p}_u\|\|\boldsymbol{q}_i\|} \tag{2}$$

其中，$\boldsymbol{p}_u$ 为工人 u 的隐含特征向量，$\boldsymbol{q}_i$ 为任务 i 的隐含特征向量。下面可以定义预测匹配度与实际匹配度之间的损失函数，然后对损失函数进行迭代优化，并加入正则化因子以避免过拟合现象，找到最合适的参数 p 和 q，损失函数如式(3)所示：

$$\min_{p,q}=\min_{p,q}\{\sum_{u,i}(md_{ui}-\hat{md}_{ui})^2+\lambda\|\boldsymbol{p}_u\|^2+\|\boldsymbol{q}_i\|^2\} \tag{3}$$

其中，md_{ui} 为工人 u 和任务 i 的实际匹配度，$\hat{md}_{ui}$ 为工人 u 和任务 i 的预测匹配度，λ 为正则化参数。

本文使用随机梯度下降算法求得损失函数的全局最小值。首先，通过求参数的偏导数找到最快速的下降方向，分别得到参数 p 和 q 的更新方向；然后将参数沿着最快速下降方向向前推进，通过迭代法不断优化参数，直至参数收敛；最终提取出最优的工人-隐含特征矩阵和任务-隐含特征矩阵，得到工人-任务匹配度矩阵。

3 基于 RankNet 排序模型的众包任务推荐

任务隐含特征矩阵由任务特征向量组成，矩阵的每一列对应任务 i_n 的特征向量（$\boldsymbol{f}_1,\boldsymbol{f}_2,\cdots,\boldsymbol{f}_k$），将任务特征矩阵按照每列分解成任务特征向量，再结合任务对应的匹配度矩阵的值做进一步处理，就能够构建 RankNet 算法的训练样例，接下来就可以利用 RankNet 算法训练排序模型进行任务推荐。

3.1 RankNet 排序模型

RankNet 排序学习是一种有监督的机器学习方法，其任务是在给定标注的训练数据的基础上，利用机器学习的方法训练得到一个排序模型，使之后的目标对象能够根据这个排序模型进行排序[16]。RankNet 的核心思想是提出一种概率损失函数来学习排序模型，将排序问题转化为概率问题，通过缩小预测概率和真实概率间的损失来训练模型，并应用排序模型对目标对象进行排序[17]。图 3 所示为本文排序学习模型的基本框架，该框架主要用于解决图 1 中如何训练得到排序模型，并为工人进行任务推荐的问题。图 3 中的学习系统是 RankNet 通过机器学习方法训练得到的排序模型，排序系统是通过排序模型对任务进行排序。首先通过学习系统对训练数据集进行训练得到排序模型，然后排序系统通过排序模型对工人 u 对应的测试任务集进行打分排序，最后输出排序后的任务列表。其中任务集 $I=\{i_1,i_2,\cdots,i_n\}$。

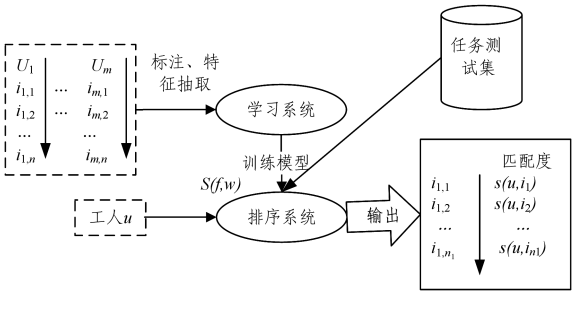


图 3 排序学习模型框架

Fig. 3 Learning to rank model framework

通过隐语义模型对任务进行特征抽取得到特征向量 $\boldsymbol{f}$，基于隐语义模型计算得到的匹配度对任务对进行标注得到 *label* 值，向量 $\boldsymbol{f}$ 与标注 *label* 组合，形成该任务对的一个训练示例（$f_i,f_j,label_{ij}$）；然后将用于训练的任务对示例进行排序学习，通过机器学习的方式缩小模型预测匹配性概率与真实匹配性概率之间的损失，得到最适合的模型 $s(\boldsymbol{f},w)$；最后以测试任务集 $i'=\{i_1',i_2',\cdots,i_n'\}$ 为输入，模型 s 将按顺序依次输出所有排序任务的匹配度打分 $s(u,i_n')$。

RankNet 是从概率的角度解决排序问题，需要定义两个概率：预测匹配性概率和真实匹配性概率。预测匹配性概率为排序模型输出计算得到的概率，真实匹配性概率则通过隐语义模型得出的匹配度计算得到。但是隐语义模型考虑的工人特征与任务特征的交集，只属于工人或任务的特征未予以考虑。因此，为提高 RankNet 模型性能，需要加入工人行为特征和任务效用特征来对真实匹配性概率进行进一步优化。

3.2 基于工人行为和任务效用的真实性匹配概率

根据隐语义模型处理得到的训练集数据中，对于工人 u，任意一个任务对$\langle i_i,i_j\rangle$都有与之相应的匹配度，如果 i_i 的匹配度大于 i_j 的匹配度，那么 i_i 比 i_j 更加匹配。计算 i_i 比 i_j 更匹配的概率如式(4)所示：

$$\overline{p}_{ij}=\frac{1}{2}(1+\hat{md}_{ui}-\hat{md}_{uj}) \tag{4}$$

在实际情况中，工人-任务匹配度不仅由相同特征所决定，工人对任务的点击次数以及浏览时间等工人隐式行为特征也影响着工人-任务的匹配度。任务的剩余完成时间、回报等关乎任务效用的特征同样影响着工人-任务匹配度。对于相同匹配度的任务对$\langle i_i,i_j\rangle$，如果工人浏览过任务 i_i 而没有浏览过任务 i_j，那么认为任务 i_i 的匹配度大于任务 i_j；同样地，若任务 i_i 是紧急任务而任务 i_j 不是紧急任务，则 i_i 优先于 i_j 匹配。因此真实匹配性概率需要在隐语义模型得到的匹配度上再加上工人行为和任务效用的影响值。本文认为工人行为和任务效用对任务匹配度产生的影响相同。

3.2.1 基于熵值法的工人影响值计算

工人影响值与工人隐式行为有关，通过对数据集进行预处理得到能够对匹配度产生影响的工人行为，具体包括点击任务、浏览任务、收藏行为、交流行为。工人行为集合为 $A=\{A_a\mid a=1,2,3,4\}$。但是不同行为对匹配度的影响不同，即其权重不同。熵值法是应用较为广泛的客观赋权法之一[18]，熵值法根据各种工人行为观测值所提供的信息量来确定工人行为的权重。

首先计算工人行为的信息熵，其定义如式(5)和式(6)所示：

$$E_a=-\frac{1}{\ln(m)}\sum_{u=1}^{m}K_{ua}\ln(K_{ua}) \tag{5}$$

$$K_{ua}=\frac{X_{ua}}{\sum_{u=1}^{m}X_{ua}} \tag{6}$$

其中，E_a 表示第 a 种工人行为的信息熵，且 $E_a\geqslant 0$；K_{ua} 表示在第 a 个行为下工人 u 此行为的比重；X_{ua} 表示工人 u 对第 a 种行为的操作数。

得到工人行为的信息熵后，通过式(7)计算第 a 种行为的权重 W_a：

$$W_a=\frac{1-E_a}{\sum_{a=1}^{4}(1-E_a)} \tag{7}$$

计算工人影响值的公式如式(8)所示：

$$b_{ui}=\frac{1}{2}\sum_{a=1}^{4}W_a(1-\hat{md}_{ui}) \tag{8}$$

3.2.2 基于序关系分析法的任务影响值计算

任务影响值与任务自身因素有关，通过对数据集进行预

处理得到对匹配度产生影响的任务因素，具体包括会员任务、优质任务、紧急任务、任务回报率，记为 G_1，G_2，G_3，G_4，任务因素集合记为 $G=\{G_c \mid c=1,2,3,4\}$。不同的因素对匹配度的影响不同，其影响值主要由众包平台决定，具有主观性，而序关系分析法是应用较为广泛的主观赋权法[19]。

首先确定序关系，在任务因素集合 $\{G_1,G_2,G_3,G_4\}$ 中选出对匹配度影响最大的因素，记为 G_1^*，接着在余下的 3 个因素中选出对匹配度影响最大的因素，记为 G_2^*，以此类推，选出 G_3^*，G_4^*，这样就唯一确定了序关系。

从众包平台考虑，在任务因素集合 G 中，会员任务对匹配度的影响最大，其影响权重记为 G_1^*；其次是优质任务，记为 G_2^*；接下来是紧急任务 G_3^* 和任务回报率 G_4^*。其序关系表示为 $G_1^* > G_2^* > G_3^* > G_4^*$。然后对任务因素做相对重要性判断。设主观上对因素 G_{c-1}^* 的重要性判断为 w_{c-1}^*，对 G_c^* 的重要性判断为 w_c^* $(c=2,3,4)$，则二者之比为 $w_{c-1}^*/w_c^* = r_c^*$。最后经过计算得到任务集合 G 中元素对应的影响权重 G_c^*，则任务影响值计算公式如式(9)所示。最终可以得到改进后的真实匹配性概率 $\overline{P}_{ij}$。

$$t_i=\frac{1}{2}\sum_{c=1}^{4}G_c^*(1-\hat{md}_{ui}) \tag{9}$$

3.2.3 真实性匹配概率计算

改进后的真实匹配性概率的计算公式如下：

$$\overline{P}_{ij}=\frac{1}{2}[1+(\hat{md}_{ui}+b_{ui}+t_i)-(\hat{md}_{uj}+b_{uj}+t_j)] \tag{10}$$

其中，b_{ui} 为工人影响值，表示工人 u 对于任务 i 隐式行为中的自身因素；t_i 为任务影响值，表示任务推荐中任务 i 的自身因素。

3.3 基于改进排序模型的众包任务推荐

通过综合考虑工人-任务各方面的特征得到真实匹配性概率，接下来构建预测匹配性概率和真实匹配性概率间的损失函数(见式(11))，进行训练优化得到排序模型，然后基于排序模型进行推荐。RankNet 排序模型通过神经网络和梯度下降算法优化交叉熵损失函数[20]。

$$C=-\overline{P}_{ij}\log P_{ij}-(1-\overline{P}_{ij})\log(1-P_{ij}) \tag{11}$$

其中，$\overline{P}_{ij}$ 为改进后的真实匹配性概率，P_{ij} 为模型预测概率。经排序学习训练得到排序模型后，排序模型为工人 u 在测试任务集合 I_n' 中对每个任务的得分进行排序，将排序后的 Top-k 个任务作为推荐子集 $I'=\{i_1,i_2,\cdots,i_k\}$ 推荐给工人 u。

4 实验结果与分析

4.1 实验准备

本文实验采用开源社区平台 SourceForge 提供的数据集，该数据集由 FLOSSmole 收集提供，为已有研究中的常用数据集，如文献[21]就使用了该数据集。该数据集包含工人和任务的相关数据以及日志文件，实验通过本文算法为工人推荐合适的众包任务。在进行实验之前，对数据集进行预处理，将信息不完整的工人任务排除，得到 10 127 名工人和 11 364 个任务。每次实验选取部分数据作为实验数据集，将数据集按照 4∶1 的比例划分为两个部分，分别用于训练和测试，并使用交叉验证方法进行实验。

为了验证本文任务推荐算法的有效性，选取 3 种基准方法进行对比，分别是基于用户的协同过滤算法(Baseline1)、基于 RankNet 的排序学习算法(Baseline2)以及基于隐语义模型的协同过滤算法(Baseline3)[22]。实验从特征维度 F、数据集大小 DS、推荐个数 K 3 个方面比较上述算法与本文算法在相关指标上的表现。其中 Baseline1 和 Baseline3 方法中协同过滤时的近邻数取默认值 10。根据经验，在本文算法和 Baseline3 方法中设置隐语义模型 LFM 的学习率 $\alpha=0.02$，正则化参数 $\lambda=0.01$。实验参数如表 1 所列。

表 1 实验参数

Table 1 Experimental parameters

参数	取值范围
DS	2000,4000,6000,8000,10000
F	20,25,30,35,40
K	2,4,6,8,10

4.2 评价指标

本实验主要衡量排序模型 Top-k 的推荐效果，采用召回率(Recall)、归一化折扣累积增益(Normalized Discounted Cumulative Gain，NDCG)以及平均准确度(Mean Average Precision，MAP)指标来评价推荐结果。

召回率是用来评价推荐效果的常用指标，其公式为：

$$Recall=\frac{1}{m}\sum_{u=1}^{m}\frac{L(u)}{T(u)} \tag{12}$$

其中，$L(u)$ 表示推荐给工人 u 正确的任务集合，$T(u)$ 为与工人 u 相匹配的任务集合。

NDCG 是衡量排序质量的指标之一，其公式为：

$$NDCG@K=Z_k\sum_{k=1}^{K}\frac{2^{r(k)}-1}{\log_2(1+k)} \tag{13}$$

其中，$r(k)$ 表示目标工人返回的第 k 个任务的标记；K 为排序选取的前 K 个任务；Z_k 是标准化因子，为输出理想排序时 $\sum_{k=1}^{K}\frac{2^{r(k)}-1}{\log_2(1+k)}$ 结果的倒数。该指标对靠前的结果更为敏感。

MAP 是在计算每个推荐结果准确率均值的基础上再计算算术平均值。相关的推荐越靠前，MAP 值就越大。其公式为：

$$MAP=\frac{1}{m}\sum_{u=1}^{m}AveP_u \tag{14}$$

$$AveP_u=\frac{1}{M}\sum_{k=1}^{K}[P(k)*rl(k)] \tag{15}$$

其中，m 为工人个数；k 为位置；M 是为工人 u 推荐的 K 个任务中被选择的个数；$rl(k)$ 表示处在 k 位置的准确率，即在 k 位置时返回结果的准确率；$p(k)$ 为 k 位置处工人选择的任务数与 k 的比值，即工人参与过的任务数量与 k 的比值。对所有工人的平均准确率 $AveP_u$ 求平均即为 MAP。

4.3 实验结果分析

4.3.1 NDCG 指标

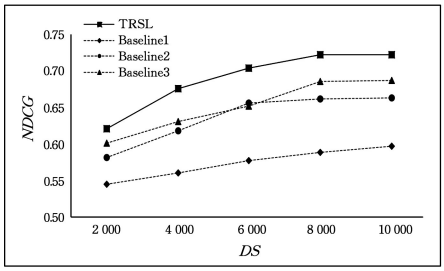
在推荐领域中，NDCG 评价指标是衡量推荐结果好坏最重要的指标之一，它表明了推荐结果的精度，也直接反映了推荐结果与工人的契合度。本组实验分别比较了 4 种算法下，

数据集大小、特征维度数量、推荐个数这 3 种条件对 NDCG 的影响。如图 4 所示，各方法的 *NDCG* 值基本都随着参数的增加而提升，推荐精度有所提高，而本文所提出的基于排序学习的软件众包任务推荐算法（TRSL）整体上的推荐精度比其他基准方法的推荐精度更高。

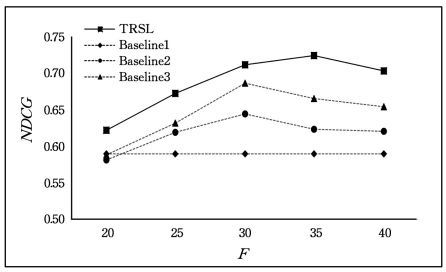
图 4(a)中，随着数据集的增大，TRSL 算法的推荐精度逐步提高，但数据集较大时，提升效果并不明显，呈平滑状态。这是因为在模型训练过程中，在数据集较大的情况下，推荐精度对于数据集的增长并不敏感，基准方法 Baseline2 和 Baseline3 呈现同样的状态。传统的协同过滤算法 Baseline1 的推荐精度随着数据集的增大而提升，但提升效果并不明显，较为缓慢。

图 4(b)中，随着特征维度的增加，TRSL 算法的推荐精度先提升后下降，说明特征维度 F 的取值并不总是越大越好。结果显示 F 值为 35 左右时，TRSL 的推荐精度最高。基准方法 Baseline2 和 Baseline3 在 F 值为 30 左右时推荐精度达到最高。由于基准方法 Baseline1 不受特征维度的影响，因此其 *NDCG* 值是固定值。

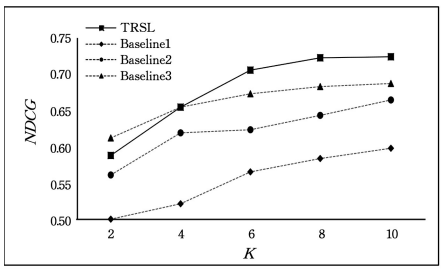
图 4(c)中，随着推荐个数 K 的增加，TRSL 算法基本优于其他算法。结果显示，在推荐个数较少时，基准方法 Baseline3 的效果较好，但 TRSL 算法也优于 Baseline1 和 Baseline2，且随着 K 值的增加，推荐精度持续提高。



(a)数据集对 NDCG 的影响



(b)特征维度对 NDCG 的影响



(c)推荐个数对 NDCG 的影响

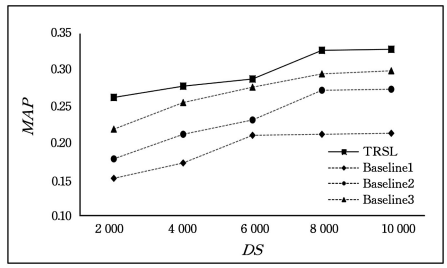
图 4　4 种算法在 NDCG 指标上的结果

Fig. 4　Results of four algorithms on NDCG indicator

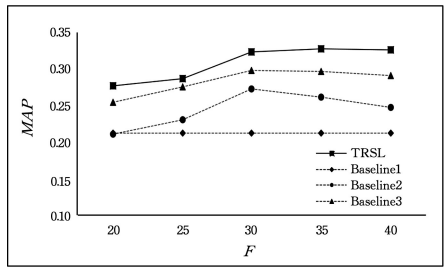
当 $K=10$ 时，本文 TRSL 算法的推荐精度 NDCG@10 达到 0.722，相比其他 3 种基准方法，分别提高了 18.6%，10.2%，5.2%。同时，基准方法 Baseline2 使用的是基于 RankNet 的排序学习算法，这也说明了本文的改进有助于提升推荐结果的精度。

4.3.2　MAP 指标

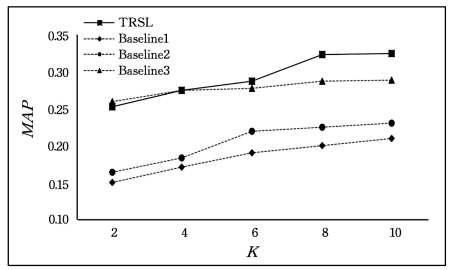
MAP 是衡量推荐结果质量的重要指标，其反映了返回结果中任务是否适合工人，且任务在结果中的位置越靠前，*MAP* 值就越大。这组实验对比了 4 种算法下不同参数对于 MAP 的影响。从图 5 可以看出，本文算法 TRSL 的 *MAP* 整体上比其他 3 种方法更具有优势，但与基准方法 Baseline1 相比优势并不明显，结果差距并不大。随着各参数的增加，大部分方法的 *MAP* 呈增长趋势。



(a)数据集对 MAP 的影响



(b)特征维度对 MAP 的影响



(c)推荐个数对 MAP 的影响

图 5　4 种算法在 MAP 指标上的结果

Fig. 5　Results of four algorithms on MAP indicator

图 5(b)中，TRSL 算法的 *MAP* 值并不是随着特征维度的增加而持续提高，当特征维度达到 35 左右时，*MAP* 值最大，之后 MAP 值基本不变。而 Baseline2 和 Baseline3 在 F 值为 30 时 *MAP* 值最大，之后开始降低，并且 Baseline3 算法的 *MAP* 值降低效果不明显，Baseline2 算法的 *MAP* 值缓慢降低。同样，基准方法 Baseline1 不受特征维度的影响，MAP 值为固定状态。

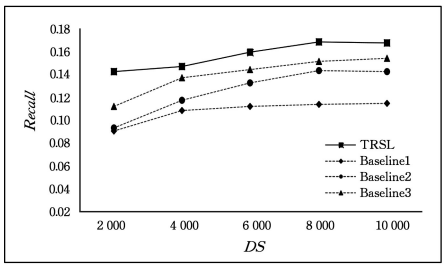
图 5(c)中，随着推荐个数 K 的增加，TRSL 算法与基准方法 Baseline3 的 *MAP* 值差距并不大，并且在推荐个数较少

时，TRSL 算法的 *MAP* 值略低于 Baseline3，但差距很小。

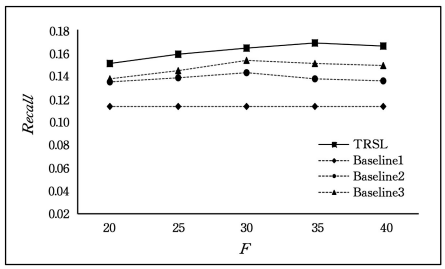
4.3.3 Reacll 指标

召回率为衡量推荐结果的指标之一，在一定程度上体现了推荐结果的质量。Recall 反映了推荐结果中工人相关任务与所有工人相关任务的比值。这组实验对比了 4 种算法下不同参数对 Recall 的影响。由图 6 可以看出，本文算法 TRSL 在 Recall 上的效果整体上优于其他 3 种基准算法，并且随着参数的增加，Recall 值基本上都逐渐提升。

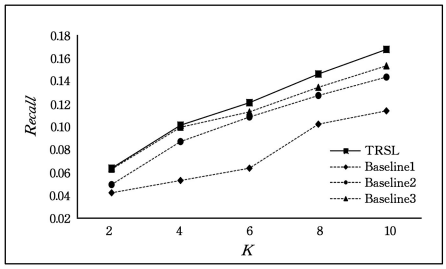
同样地，在图 6(b)中，随着特征维度越来越大，特征冗余会对结果造成影响，TRSL 的 *Recall* 值在特征维度 F 达到 35 左右时最大，Baseline2 和 Baseline3 在 F 值为 30 左右时 *Recall* 值最大，Baseline1 方法不受 F 值影响，*Recall* 值为固定值且较低。



(a)数据集对 Recall 的影响



(b)特征维度对 Recall 的影响



(c)推荐个数对 Recall 的影响

图 6 4 种算法在 Recall 指标上的结果

Fig. 6 Results of four algorithms on Recall indicator

4.3.4 实验小结

上述实验结果表明，本文算法 TRSL 与其他 3 种算法相比，在推荐精度上都有提升，尤其是相对于传统的协同过滤算法，提升效果较好。相比基于 RankNet 的算法，TRSL 同样有明显的提升效果，说明了本文改进的正确性相对比结合隐语义模型的协同过滤，TRSL 整体上的推荐精度依然更高，但当推荐个数较少时，在 MAP 和 NDCG 上效果较差，但差距并不明显。

综上可以得出结论：当使用其他 3 种对比算法进行任务推荐时，结果较为有限；而利用机器学习中的排序学习方法将推荐问题转化为排序问题，并综合考虑工人-任务显式特征和隐式特征时，推荐精确度能取得更好的效果。

结束语 推荐技术作为缓解互联网时代信息过载问题的有效手段，在众多领域深受关注。本文面向互联网上开源开放的众包平台，将机器学习中的排序学习方法应用于工人与任务之间的推荐，以实现降低工人利益受到损害的风险和提高众包平台软件开发质量的双赢效果。本文采用隐语义模型和 RankNet 算法相结合的方式，基于隐式反馈信息对隐语义模型和 RankNet 算法分别加以改进，最后通过 NDGG，MAP 以及 Recall 来综合考量改进后的 RankNet 排序模型。实验结果表明：与基准推荐算法相比，基于排序学习的推荐算法可以取得更高的推荐精度，最高提升 18.6%，并且与仅基于 RankNet 的排序学习算法相比，提升幅度为 10.2%，证明了本文对于 RankNet 的改进对推荐精度具有提升效果。在下一步的工作中，可以考虑：1)建立工人兴趣迁移模型，通过与排序模型融合的方式向工人推荐合适的任务；2)采用更为优化的排序学习模型(如 LambdaRank)来构建排序模型并进行推荐；3)在保证推荐精度的前提下，将推荐多样性指标加入进来，使推荐结果具有多样性。

参 考 文 献

[1] STOL K J, LATOZA T D, BIRD C. Crowdsourcing for Software Engineering[J]. IEEE Software, 2017, 34(2): 30-36.

[2] KE M, LICIA C, MARK H, et al. A survey of the use of crowdsourcing in software engineering[J]. The Journal of Systems & Software, 2017: 57-84.

[3] KARIM M R, YANG Y, MESSINGER D, et al. Learn or Earn? -Intelligent Task Recommendation for Competitive Crowdsourced Software Development[C]// Hawaii International Conference on System Sciences. 2018.

[4] AMBATI V, VOGEL S, CARBONELL J. Towards task recommendationin micro-task markets[C]// Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence. 2011: 80-83.

[5] YUEN M C, KING I, LEUNG K S. Taskrec: probabilistic matrix factorization in task recommendation in crowdsourcing systems[C]// International Conference on Neural Information Processing. Springer, Berlin, Heidelberg, 2012: 516-525.

[6] LIU M B, MAN J F, PENG C, et al. Multidimensional recommendation algorithm with implicit feedback[J]. Application Research of Computers, 2020, 37(1): 158-162.

[7] ZHU J, SHEN B, HU F. A learning to rank framework for developer recommendation in software crowdsourcing[C]// 2015 Asia-Pacific Software Engineering Conference (APSEC). IEEE, 2015: 285-292.

[8] FU Y, CHEN H, SONG F. STWM: A solution to self-adaptive task-worker matching in software crowdsourcing[C]// International Conference on Algorithms and Architectures for Parallel Processing. Springer, Cham, 2015: 383-398.

[9] LI N, MO W, SHEN B. Task recommendation with developer

social network in software crowdsourcing[C]// 2016 23rd Asia-Pacific Software Engineering Conference (APSEC). IEEE, 2016:9-16.

[10] ZHONG Q Y,ZHANG Y,LI C,et al. Task recommendation method based on workers' interest and competency for crowdsourcing[J]. Systems Engineering Theory & Practice, 2017,37(12):3270-3280.

[11] MAO K,YANG Y,WANG Q,et al. Developer recommendation for crowdsourced software development tasks[C]// 2015 IEEE Symposium on Service-Oriented System Engineering. IEEE, 2015:347-356.

[12] SHAO W,WANG X,JIAO W. A developer recommendation framework in software crowdsourcing development[C]// National Software Application Conference. Springer, Singapore, 2016:151-164.

[13] AANCHAL M,NEHA J,EMILIE C,et al. Deep latent factor model for collaborative filtering[J]. Signal Processing, 2019, 169:107366.

[14] WU K. Research on Personalized Recommendation Algorithm Based on Latent Factor Model[D]. Guangzhou:Guangdong University of Technology,2016.

[15] FAN H T,ZHONG C L,GONG H H. Latent Factor Model Based Personalized Recommendation[J]. Computer Applications and Software,2017,34(12):206-210.

[16] HUANG Z H,ZHANG J W,TIAN C Q,et al. Survey on Learning-to-Rank Based Recommendation Algorithms[J]. Journal of Software,2016,27(3):691-713.

[17] HANG L I. A short introduction to learning to rank[J]. IEICE TRANSACTIONS on Information and Systems,2011,94(10): 1854-1862.

[18] WANG H,XU C,XU Z S. An approach to evaluate the methods of determining experts' objective weights based on evolutionary game theory [J]. Knowledge-Based Systems, 2019, 182(15): 104862.1-104862.16.

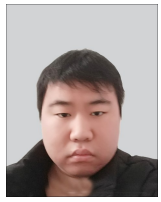
[19] WU Y K,SONG R S,CHEN B. A Comprehensive Weight Method Based on the Game Theory for Information Security Risk Assessment[J]. Computer Engineering and Science,2011, 33(5):9-13.

[20] SONG Y,WANG H,HE X. Adapting deep RankNet for personalized search[C]// Acm International Conference on Web Search & Data Mining. ACM,2014.

[21] YANG X H,LI B,HE P,et al. An Approach to Project Recommendation in Collective Software Develoment[J]. Journal of Chinese Computer Systems,2015,36(4):671-676.

[22] LU L,WEI D Y. A Collaborative Filtering Algorithm Based on Latent Factor Model[J]. Microelectronics Computer,2015(2): 73-75.

YU Dun-hui, born in 1974, Ph.D, professor, is a member of China Computer Federation. His main research interests include service computing and big data.



CHENG Tao, born in 1995, M. S. candidate. His main research interests include big data and so on.