软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software [doi: 10.13328/j.cnki.jos.006994]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于工人长短期时空偏好的众包任务分配*

王府鑫[1,2], 王 宁[1,2], 曾奇雄[1,2]

[1](北京交通大学 计算机与信息技术学院, 北京 100044)

[2](交通数据分析与挖掘北京市重点实验室 (北京交通大学), 北京 100044)

通信作者: 王宁, E-mail: nwang@bjtu.edu.cn

摘 要: 近年来, 随着移动设备的计算能力和感知能力的提高, 基于位置信息的时空众包应运而生, 任务分配效果的提升面临许多挑战, 其中之一便是如何给工人分配他们真正感兴趣的任务. 现有的研究方法只关注工人的时间偏好而忽略了空间因素对偏好的影响, 仅关注长期偏好却忽略了短期偏好, 同时面临历史数据稀疏导致的预测不准的问题. 研究基于长短期时空偏好的任务分配问题, 从长期和短期两个角度以及时间和空间两个维度全面考虑工人的偏好, 进行时空众包任务分配, 提高任务的成功分配率和完成效率. 为提升时空偏好预测的准确性, 提出分片填充的张量分解算法 (SICTD) 减小偏好张量的空缺值占比, 提出时空约束下的 ST-HITS 算法, 综合考虑工人短期活跃范围, 计算短期时空偏好. 为了在众包任务分配中最大化任务总收益和工人偏好, 设计基于时空偏好的贪心与 Kuhn-Munkres (KM) 算法, 优化任务分配的结果. 在真实数据集上的大量实验结果表明, 提出的分片填补张量分解算法对时间和空间偏好的 RMSE 预测误差较基线算法分别下降 22.55% 和 24.17%; 在任务分配方面, 提出的基于偏好的 KM 算法表现出色, 对比基线算法, 在工人总收益和工人完成任务平均偏好值上分别提升 40.86% 和 22.40%.

关键词: 时空众包; 任务分配; 张量分解; 偏好预测

中图法分类号: TP311

中文引用格式: 王府鑫, 王宁, 曾奇雄. 基于工人长短期时空偏好的众包任务分配. 软件学报. http://www.jos.org.cn/1000-9825/6994.htm

英文引用格式: Wang FX, Wang N, Zeng QX. Long- and Short-term Spatio-temporal Preference-aware Task Assignment in Spatial Crowdsourcing. Ruan Jian Xue Bao/Journal of Software (in Chinese). http://www.jos.org.cn/1000-9825/6994.htm

Long- and Short-term Spatio-temporal Preference-aware Task Assignment in Spatial Crowdsourcing

WANG Fu-Xin[1,2], WANG Ning[1,2], ZENG Qi-Xiong[1,2]

[1](School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China)

[2](Beijing Key Lab of Traffic Data Analysis and Mining (Beijing Jiaotong University), Beijing 100044, China)

Abstract: With the development of mobile services' computing and sensing abilities, spatial crowdsourcing, which is based on location information, comes into being. There are many challenges to improving the performance of task assignments, one of which is how to assign workers the tasks that they are interested in. Existing research methods only focus on workers' temporal preference but ignore the impact of spatial factors on workers' preference, and they only focus on long-term preference but ignore workers' short-term preference and face the problem of inaccurate predictions caused by sparse historical data. This study analyzes the task assignment problem based on long-term and short-term spatio-temporal preference. By comprehensively considering workers' preferences from both long-term and short-term perspectives, as well as temporal and spatial dimensions, the quality of task assignment is improved in task assignment success rate and completion efficiency. In order to improve the accuracy of spatio-temporal preference prediction, the study proposes a sliced imputation-based context-aware tensor decomposition algorithm (SICTD) to reduce the proportion of missing values in preference tensors

* 基金项目: 国家重点研发计划 (2018YFC0809800)
收稿时间: 2022-11-23; 修改时间: 2023-04-12; 采用时间: 2023-06-29; jos 在线出版时间: 2023-11-08

and calculates short-term spatio-temporal preference through the ST-HITS algorithm and short-term active range of workers under spatio-temporal constraints. In order to maximize the total task reward and the workers' average preference for completing tasks, the study designs a spatio-temporal preference-aware greedy and Kuhn-Munkres (KM) algorithm to optimize the results of task assignment. Extensive experimental results on real datasets show the effectiveness of the long- and short-term spatio-temporal preference-aware task assignment framework. Compared with baselines, the RMSE prediction error of the proposed SICTD for temporal and spatial preferences is decreased by 22.55% and 24.17%, respectively. In terms of task assignment, the proposed preference-aware KM algorithm significantly outperforms the baseline algorithms, with the workers' total reward and average preference for completing tasks averagely increased by 40.86% and 22.40%, respectively.

Key words: spatial-temporal crowdsourcing; task assignment; tensor decomposition; preference prediction

近年来, 随着移动互联网和共享经济的蓬勃发展, 众包技术融入具有时空数据的应用场景中, 传统的在线众包平台模式转变为一种新型的服务模式, 即"时空众包"[1]. 通过时空众包, 任务请求者可以向服务器发布时空任务 (例如接送乘客或监控车流), 服务器会将这些任务分配给工人, 由工人移动到指定位置完成任务. 随着用户规模的增大, 用户个性和偏好呈现多样化[2−4]. 在实际生活中, 工人对任务的偏好会影响任务的完成效果, 满足用户偏好的分配方案能促使工人高效高质量地完成任务, 因此, 在时空众包任务分配中考虑工人的偏好十分重要.

目前的研究主要通过工人过去完成任务的模式或者显式反馈来推测其偏好, 比如将工人历史完成任务的记录和时间相结合, 通过张量分解预测每个工人在各时间段对某类任务的时间偏好[5], 并在此基础上, 将传统的单人完成任务场景扩展至组任务分配, 考虑组时间偏好与组间工人的关系进行任务分配[6]; 为了解决数据稀疏问题, 有研究提出工人间共有信息的概念, 通过最大化工人间共有信息求得组时间偏好[7]. 但是, 目前基于偏好的任务分配研究仅考虑了时间偏好, 却忽略了工人的空间偏好对任务分配的影响, 也没能捕捉工人的短期偏好. 实际上, 除了时间偏好, 工人也有特定的空间偏好, 比如更愿意去熟悉的地点, 因此联合考虑时空偏好是必要的. 此外, 偏好还有长短期之分, 长期偏好反映工人完成任务的一般规律, 而某些突发情况会影响工人的短期偏好. 短期偏好往往和工人短期的活跃范围和区域热门程度有关, 比如工人更愿意前往人流量较大的地点完成接送类任务, 因此需要捕捉工人短期完成任务的偏好, 方可给出更好的分配方案.

图 1 展现了某个时间段的任务分配实例, 包括分布在 4 个地理网格 $\{g_1, g_2, g_3, g_4\}$ 内的 3 名工人 $\{w_1, w_2, w_3\}$ 和 4 个任务 $\{s_1, s_2, s_3, s_4\}$、工人的时间和空间偏好、工人的短期活跃范围和当前时段网格热门程度. 如果仅根据工人的时间偏好进行任务分配, 可以得到分配结果对 $\{< w_1, s_3 >, < w_2, s_2 >, < w_3, s_4 >\}$. 然而, 该分配结果忽视了空间偏好和短期偏好对工人完成任务的影响. 比如 w_3 虽然对 s_4 的时间偏好为 0.9, 但 s_4 并不在 w_3 的短期活跃范围内, 并且 w_3 对去网格 g_2 完成任务的空间偏好也只有 0.3, 尽管给 w_3 分配了任务, 如果他不愿意去做, 会导致 w_3 完成该任务的效率和质量降低, 甚至放弃该任务.

本文提出基于工人长短期时空偏好的众包任务分配问题 LSPTA (long- and short-term spatio-temporal preference-aware task assignment), 从长期和短期两个角度以及时间和空间两个维度全面考虑工人的偏好, 最大化工人对任务的偏好值以及总的收益. 我们提出一个两阶段框架解决 LSPTA 问题, 包括偏好预测与任务分配两个阶段. 第 1 阶段将时间偏好与空间偏好建模成三维张量, 提出 SICTD (sliced imputation-based context-aware tensor decomposition) 算法, 以分片填补的方式解决数据稀疏问题, 并在辅助上下文矩阵的帮助下进行张量分解, 更加准确地预测工人长期时空偏好; 同时我们提出时空众包场景下的 ST-HITS 算法, 综合考虑工人的短期活跃范围和网格的热门程度, 求得工人的短期时空偏好. 第 2 阶段基于工人的长短期时空偏好进行任务分配, 设计基于时空偏好的贪心和 Kuhn-Munkres (KM) 算法, 使工人总任务收益最大的同时, 最大化工人对分配任务的总偏好值. 图 1 中红色箭头表示基于我们提出的 KM+Pre 算法的任务分配结果, 即 $\{< w_1, s_2 >, < w_2, s_4 >, < w_3, s_1 >\}$. 我们的算法全面考虑工人的长短期时空偏好, 第 6 节给出的大量实验结果表明, 与现有算法相比, KM+Pre 算法能显著提高任务分配的质量、任务的执行率和完成效率.

综上所述, 本文的贡献如下.

(1) 首次提出从长期和短期两个角度以及时间和空间两个维度全面考虑工人的偏好, 进行时空众包任务分配, 提高任务分配质量、成功率和完成效率.

(2) 提出分片填补的方法解决偏好张量分解面临的数据稀疏问题, 并引入上下文辅助矩阵减小工人长期时空偏好预测的误差; 同时提出时空众包场景下的 ST-HITS 算法, 综合考虑工人短期活跃范围, 求得工人的短期时空偏好.

(3) 提出基于长短期时空偏好的贪心算法, 优先将报酬最大的任务分配给对其偏好最大的工人; 为了避免已有任务分配结果对后续分配产生影响, 我们又提出基于长短期时空偏好的 KM 算法, 在工人-任务二分图中进行深度优先搜索寻找最大匹配, 进一步优化任务分配结果.

(4) 在两个真实数据集上完成了大量实验, 实验结果显示, 与普通的张量分解算法相比, 我们提出的基于分片填补的张量分解算法在时间和空间偏好的预测准确性上分别提升了 22.55% 和 24.17%; 任务分配方面, 基于长短期时空偏好的 KM 算法表现出色, 在工人总收益和工人完成任务的平均偏好值上相比基线方法分别提升 40.86% 和 22.40%.

网格热门程度

g	*popularity*
g_1	0.7
g_2	0.8
g_3	0.5
g_4	0.6

工人长期时间偏好

w	$LP^t_{(w,s_1)}$	$LP^t_{(w,s_2)}$	$LP^t_{(w,s_3)}$	$LP^t_{(w,s_4)}$
w_1	0.5	0.65	0.8	0.4
w_2	0.3	0.7	0.6	0.5
w_3	0.6	0.4	0.5	0.9

工人长期空间偏好

w	$LP^g_{(w,g_1)}$	$LP^g_{(w,g_2)}$	$LP^g_{(w,g_3)}$	$LP^g_{(w,g_4)}$
w_1	0.95	0.75	0.4	0.2
w_2	0.6	0.8	0.2	0.5
w_3	0.4	0.3	0.6	0.85

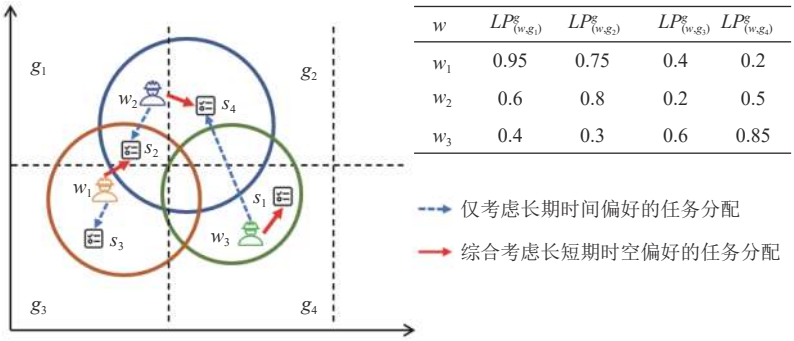

图 1 任务分配实例

本文第 1 节介绍时空众包任务分配和偏好预测的相关工作和研究现状. 第 2 节介绍本文所需的基础知识并定义本文要解决的 LSPTA 问题. 第 3 节介绍工人长短期时空偏好的任务分配框架. 第 4 节介绍长短期时空偏好模型以及实现预测和求解偏好算法. 第 5 节介绍基于时空偏好的任务分配算法. 第 6 节通过对比实验与消融实验验证了本文提出的方法的有效性. 第 7 节总结全文.

1 相关工作

时空众包在近几年获得大量关注, 任务分配成为核心的研究问题, 要求服务器基于系统的优化目标将任务分配给附近的工人, 优化目标可以是最大化总任务分配数量、最大化工人或平台所获收益和最大化分配结果的多样性等. Kazemi 等人[8]将工人任务二分图归约为最小费用最大流问题 (MCMF) 的一个实例, 设立额外的起点与终点, 并使用 Ford-Fulkerson 算法和线性规划来获得任务分配结果, 目的是最大化总任务分配数目. Ye 等人[9]则提出多中心的任务分配问题, 在一个两阶段框架中, 引入维诺图以任务分配中心为生成元进行区域划分, 调整划分标准以平衡每个区域的工人和任务数目, 并使用强化学习进行任务分配. Wang 等人[10]提出一个两阶段框架以解决工人流失的任务分配问题, 第 1 阶段基于工人完成任务历史数据, 使用 LSTM 捕捉工人潜在情感, 预测工人空闲的时间, 并在第 2 阶段将其与任务收益共同作为工人-任务二分图的权重, 使用 KM 算法及时给将要流失的工人分配任务以最大化工人所获收益. Xia 等人[11]首次考虑任务分配时最大化平台收益, 使用贪心算法分配单位工作量收益

最大的任务给距离最近的工人, 并提出粗粒度与细粒度微调策略, 舍弃了低价值的任务并对少量已分配的工人进行重新分配, 从而使平台收益最大化. 为解决基于可靠性和多样性的任务分配问题 RDB-SC, Cheng 等人[12]提出一个三阶段分治算法, 首先递归地将 RDB-SC 问题分解为两个更小的子问题, 接着使用贪心算法或采样算法进行子任务求解, 最后将子问题解合并, 给出可靠性和任务多样性最大的任务分配方案. 此外, Wang 等人[13]提出了一个工人激励模型, 使用遗传算法与蚁群算法最优化任务的完成质量, 并且最小化激励费用的开销. Zhao 等人[14]基于树结构解决具有最晚时间约束的任务分配问题, 提出最大化任务分配数量的树分解算法, 将无依赖的工人分解至不同节点中, 对树中每个节点独立地进行最优分配, 最后合并子问题结果从而求得最终分配方案; 他们进一步提出优化策略以降低工人旅行开销, 并重新设计了工人分解算法, 将算法应用范围扩大至多工人场景[15]. Cheng 等人[16]研究了合作模式众包, 在该模式下, 工人需要组队完成任务, 该研究提出贪心算法与博弈论算法, 为每个任务求得最优的工人组合, 以达到最高的任务完成质量. Zhao 等人[17]认为在合作模式下, 工人不会心甘情愿地完成任务除非得到满意的报酬, 提出基于同盟的组任务分配问题 CTA, 贪心地选取与任务相近的能产生更多收益的工人进行组队以最大化小组共同收益.

上述工作都没有考虑工人的意愿, 然而在现实生活中, 工人能否高质量完成任务与其对任务的偏好有很大的关系. Zhao 等人[5]首次考虑时间偏好对任务分配的影响, 设计了一个基于历史数据的张量分解算法 HCTD 去预测工人时间偏好, 进一步基于偏好将任务分配问题转化成最小费用最大流问题; 除此之外, 他们进一步将上述工作扩展到多人小组任务分配[6], 利用模糊逻辑求出组中工人可容忍的等待时间, 在此基础上生成有效的工人组并计算小组共识, 最后使用树分解算法进行任务分配以最大化组时间偏好与小组共识. Li 等人[7]也关注基于小组的多人任务分配, 考虑到小组内每名成员对任务有着不同的偏好, 由于任务不同, 小组组成具有多变性和偶然性, 会面临历史数据稀疏的问题. 他们使用表示学习建模工人的偏好向量, 并结合注意力机制, 学习每个小组成员对任务的偏好给小组整体偏好带来的影响, 还使用对比表示学习捕获不同小组之间的差异和小组成员之间的联系, 训练并预测得到小组偏好. Li 等人[18]考虑到组任务分配中每个小组成员的社会影响力不同, 提出基于社会影响力的偏好, 通过二分图嵌入模型 BGEM 建模小组偏好, 使用工人与任务、小组与任务之间的交互信息构建社交网络, 通过堆叠降噪自编码器 SDAE 挖掘和整合全局和局部社交网络结构信息, 并联合优化该信息与工人、任务、小组间的交互信息以减轻数据稀疏性. Zhou 等人[19]从工人与任务双边偏好的角度切入, 设定工人完成任务绕路距离的多少直接影响工人偏好, 任务偏好设定为工人的声望, 基于工人的日常轨迹, 提出工人驱动和任务驱动的贪心算法进行任务分配, 并进一步提出两个延迟接受度算法, 同时考虑双边偏好进行任务分配.

尽管现有研究已对时空众包中的偏好预测问题进行了深入研究, 但仍存在一些局限性. 具体而言, 时空众包领域中, 时间和空间是影响任务分配最重要的两个因素, 因此工人的偏好不应忽略空间偏好, 然而现有工作只关注工人的时间偏好, 或者以工人满意度、工人绕路距离等表示工人偏好, 无法捕捉工人的空间偏好. 其次, 目前时空众包相关研究无法从长短期角度共同考量工人偏好. 本文的研究工作将填补这两点不足.

2 问题定义

我们首先阐明必要的基本概念, 然后再定义本文研究的问题.

定义 1 (时空任务). 一个时空任务用 $s=(l,p,e,c,r)$ 表示, 它有 5 个属性, 分别是任务地点 l, 任务发布时间 p, 任务过期时间 e, 任务种类 c, 任务报酬 r. l 由任务的具体位置坐标 l_x 和 l_y 构成.

定义 2 (工人). 一个工人用 $w=(l,d)$ 表示, 它有两个属性, 地点 l 与可达距离 d. 工人的可达范围是以他所处地点 l 为圆心, 可达距离 d 为半径的圆. l 由工人的具体位置坐标 l_x 和 l_y 构成.

工人会接受可达范围内的任务. 在我们的研究中, 工人不能同时完成多个任务, 每个任务由一名工人完成.

定义 3 (任务完成历史). 假设工人 w 在一段时间内完成了 n 个任务, 其任务完成历史定义为一个集合 $S_w=\{(s_1,t_1,g_1),(s_2,t_2,g_2),\ldots,(s_n,t_n,g_n)\}$, 集合中每个三元组代表 w 完成的某个任务 s_i, 该任务完成的时间段 t_i 和所处的地理网格 g_i. w 的任务完成历史可以简写为 $S_w=\{s_1,s_2,\ldots,s_n\}$.

定义 4 (长期时空偏好). 给定任务类别 c 和工人 w 的任务完成历史记录 $S_w=\{s_1,s_2,\ldots,s_n\}$, 工人 w 在时间段 t

对任务类别 c 的长期时空偏好由他的时间偏好 $LP_w^t(c)$ 和空间偏好 $LP_w^g(c)$ 两部分组成. 其中, $LP_w^t(c)$ 为工人 w 在时间段 t 完成 c 类别任务的次数与他在同时间段完成的任务总数的比值; $LP_w^g(c)$ 为工人 w 在地理网格 g 完成 c 类别任务的次数与他在同网格完成的任务总数的比值. 公式表示如下:

$$LP_w^t(c)=\frac{\sum_{s_i\in S_w} f_t(s_i.c)}{N^t(S_w)},\quad f_t(s_i.c)=\begin{cases}1, & s_i.c=c \text{ and } s_i.p\in t\\ 0, & \text{otherwise}\end{cases} \tag{1}$$

$$LP_w^g(c)=\frac{\sum_{s_i\in S_w} f_g(s_i.c)}{N^g(S_w)},\quad f_g(s_i.c)=\begin{cases}1, & s_i.c=c \text{ and } s_i.l\in g\\ 0, & \text{otherwise}\end{cases} \tag{2}$$

$N^t(S_w)$ 与 $N^g(S_w)$ 分别代表工人 w 在时间段 t 与网格 g 中完成的任务次数, $f_t(s_i.c)$ 与 $f_g(s_i.c)$ 分别表示工人 w 在相应时间段与网格中是否完成类别为 c 的任务, 如果完成了则为 1, 反之为 0.

定义 5 (短期任务完成记录). 假设工人 w 短期内完成了 m 个任务, 其短期任务完成记录是任务完成历史的子集 $L_w=\{(s_q,t_q,g_q),(s_{q+1},t_{q+1},g_{q+1}),\ldots,(s_{q+m-1},t_{q+m-1},g_{q+m-1})\}$, 表示从任务完成历史的第 q 条开始记录, 集合中每个三元组代表 w 完成的某个任务 s_j, 该任务的完成时间 t_j 和所处的地理网格 g_j. w 的短期任务完成记录可以简写为 $L_w=\{s_q,s_{q+1},\ldots,s_{q+m-1}\}$.

定义 6 (短期活跃范围). 给定工人 w 的短期任务完成记录 $L_w=\{s_q,s_{q+1},\ldots,s_{q+m-1}\}$, 其短期活跃范围 O_w 刻画了工人短期内的活动规律, 表示为以 l_w^* 为中心, r_{O_w} 为半径的圆, 其中 l_w^* 为短期内完成任务位置点的中心点, r_{O_w} 为中心点与最远任务记录位置点的距离. 公式表示如下:

$$l_{wx}^*=\frac{\sum_{i=q}^{q+m-1} s_i.l_{ix}}{m}\qquad l_{wy}^*=\frac{\sum_{i=q}^{q+m-1} s_i.l_{iy}}{m} \tag{3}$$

$$r_{O_w}=\max_{i=q}^{q+m-1} dis(s_i.l,l_w^*) \tag{4}$$

l_{wx}^* 与 l_{wy}^* 分别为短期活跃范围的中心点横坐标与纵坐标, $dis(s_i.l,l_w^*)$ 为任务 s_i 位置与中心点 l_w^* 的距离.

定义 7 (短期时空偏好). 给定工人 w 以及任务 s, w 对 s 的短期时空偏好 $SP_{(w,s)}$ 由工人近期的活跃范围以及任务位置的热门程度决定, 表示为:

$$SP_{(w,s)}=\alpha\times g_s.popularity,\ \alpha=\begin{cases}1, & dis(s.l,l_w^*)\leqslant r_{O_w}\\ 1-\min\left(1,\dfrac{dis(w.l,s.l)}{w.d},\dfrac{dis(s.l,l_w^*)-r_{O_w}}{r_{O_w}}\right), & \text{otherwise}\end{cases} \tag{5}$$

其中, α 为距离参数, 当任务与活跃范围中心 l_w^* 的距离小于等于工人的活跃范围半径 r_{O_w} 时, 设置为 1; 反之, α 随着任务与活跃范围中心距离 $dis(s.l,l_w^*)$、工人距离 $dis(w.l,s.l)$ 的增加而减小. $g_s.popularity$ 为任务 s 所处网格的热门程度, 相关算法将在第 4 节介绍.

问题定义: 给定当前时间 t_{now}, 当前时间所处的时间段 t_i, 在线工人集合 W_i 和任务集合 S_i, 所有可能出现的工人和任务分配方案集合表示为 A_i, $A_i^j\in A_i$ 代表第 j 种任务分配方案, $(w,s,pre)\in A_i$ 表示 A_i^j 中工人 w ($w\in W_i$) 对任务 s ($s\in S_i$) 的偏好为 pre. 将任务 s 分配给 w, w 与 s 之间需要满足以下时空约束.

(1) 工人 w 与任务 s 之间的距离小于等于工人的可达范围, 即 $dis(w.l,s.l)\leqslant w.d$;

(2) 工人 w 前往任务 s 的地点所需时间小于等于任务持续时间: $t_{\text{now}}+t(w.l,s.l)\leqslant s.e-s.p$.

基于工人长短期时空偏好的众包任务分配问题 LSPTA 是指, 在满足以上时空约束的情况下, 找到当前时间段 t_i 最优的任务分配方案 $A_{opt}\in A_i$, 满足以下优化目标:

$$\forall A_i^j\in A_i,(A_{opt}.r\geqslant A_i^j.r)\wedge(A_{opt}.pre\geqslant A_i^j.pre).$$

3 总体框架

基于工人长短期时空偏好的任务分配框架见图 2, 该框架包括时空偏好建模和基于偏好的时空众包任务分配.

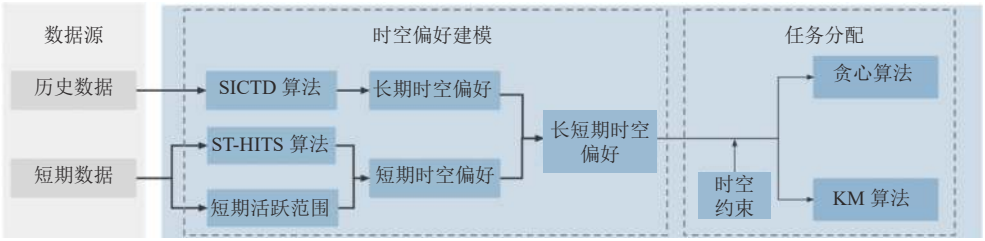

图 2 基于长短期时空偏好的众包任务分配框架图

时空偏好模型需要分别建模工人的长期时空偏好与短期时空偏好. 对于长期时空偏好, 我们首先基于工人的历史完成任务记录, 从时间与空间两个维度分别构建三维长期偏好张量, 接着构建辅助上下文矩阵, 共同参与张量分解, 减小偏好预测的误差. 考虑到张量数据稀疏问题, 我们提出基于分片填补的张量分解算法 SICTD, 通过分片填补增加张量中的数据比例, 提高工人长期偏好的预测准确性. 对于短期时空偏好, 我们对工人近期完成任务的位置进行分析, 求得工人近期的活跃范围, 并且设计了时空约束下的 ST-HITS 算法, 计算当前时间段任务所处地理网格的热门程度; 工人的短期时空偏好由工人近期的活跃范围和任务所处网格的热门程度共同决定. 工人的长短期时空偏好建模成长期与短期偏好的带权和.

在任务分配阶段, 为了完成本文的优化目标, 我们首先提出基于时空偏好的贪心算法, 优先分配报酬最大的任务, 选择对其偏好值最大的工人完成任务分配. 为了克服贪心算法可能引起的局部最优, 我们进一步提出基于时空偏好的 KM 算法, 设置工人-任务二分图中节点连线为任务报酬与工人偏好的带权和, 通过深度优先搜索在图中寻找最大匹配, 在给工人分配偏好值最大的任务的同时, 考虑工人完成任务的收益, 使得总的任务收益最大.

4 工人长短期时空偏好模型

4.1 长期时空偏好

4.1.1 偏好张量与辅助上下文矩阵构建

我们凭借工人的历史完成任务记录构建时间偏好张量 $\gamma_T \in R^{|W|\times|C|\times|T|}$ 和空间偏好张量 $\gamma_G \in R^{|W|\times|C|\times|G|}$, $|W|$表示工人总数, $|C|$表示任务类别总数, $|T|$表示时间段总数, $|G|$表示地理网格总数. 还构建了时间辅助矩阵 $X_T \in R^{|T|\times|C|}$、空间辅助矩阵 $X_G \in R^{|G|\times|C|}$ 和任务类别关系矩阵 $X_C \in R^{|C|\times|C|}$ 共同完成张量分解. X_T 和 X_G 从统计意义上分别刻画了历史任务完成情况与时间段和地理网格的关系, X_C 刻画任务类别之间的关联度, 通过在搜索引擎上对两个不同任务类别进行联合搜索, 搜索结果条目个数越多, 则认为任务类别关联度越大. X_C 联合 X_T 或 X_G 共同帮助减小张量分解的误差.

4.1.2 基于分片填补的张量分解与补全

尽管我们可以根据张量 γ_T 和 γ_G 中的非零值直接完成分解, 但张量中的数据过于稀疏会降低预测准确率. 根据观察, 使用 Foursquare 的纽约签到数据集[21]中 4 周的工人完成任务历史数据生成张量时, 非零值占比仅为 0.4%. 然而, 若单独考虑张量中的每个时间段或每个地理网格按维度分片后的矩阵, 其非零值最多能达到 1.5%, 因此我们提出基于分片填补的张量分解算法 (SICTD), 若当前分片矩阵非零值占比大于整个张量的非零值占比, 则采用矩阵分解去进行分片矩阵的填补, 并将填补后的分片矩阵还原至原张量, 通过增大张量的非零值占比提升预测准确性.

图 3 以空间偏好预测为例, 给出 SICTD 算法的流程. SICTD 对每个非零值占比大于张量非零值占比的分片矩阵, 采用 Bias-SVD 与随机梯度下降法将分片矩阵 $Sliced_i^G$ 分解为用户特征矩阵 P_i 和任务类别特征矩阵 Q_i, 并按一定比例填补矩阵中的空缺值, 用填补后的矩阵 $Filled_i^G$ 还原替换至原张量中. 对整个张量进行分片填补后, SICTD 使用 Tucker 分解将分片填补后的空间偏好张量 γ_G^f 分解为核张量 $S_g \in R^{d_W\times d_C\times d_G}$ 和 3 个低秩矩阵 $U \in R^{|W|\times d_W}$, $C \in R^{|C|\times d_C}$, $G \in R^{|G|\times d_G}$ 的乘积. d_W, d_C, d_G 表示隐因子的维度. 张量分解如公式 (6) 所示.

$$\gamma_G^f \approx S_g \times_U U \times_C C \times_G G \tag{6}$$

其中, $\times_U U$, $\times_C C$, $\times_G G$ 分别表示核张量 S 与低秩矩阵 U, C, G 的模态积. SICTD 的优化目标函数如公式 (7) 所示.

$$L_G(S,U,C,G) = \frac{1}{2}\left\|\gamma_G^f - S\times_U U\times_C C\times_G G\right\|^2 + \frac{\lambda_1}{2}\left\|X_G - GC^{\mathrm{T}}\right\|^2 + \frac{\lambda_2}{2}tr(C^{\mathrm{T}}L_{X_C}C) + \frac{\lambda_3}{2}(\|S\|^2+\|U\|^2+\|C\|^2+\|G\|^2) \tag{7}$$

其中, $\left\|\gamma_G^f - S\times_U U\times_C C\times_G G\right\|^2$ 控制张量分解的误差, $\left\|X_G - GC^{\mathrm{T}}\right\|^2$ 控制辅助矩阵 X_G 的误差, $tr(C^{\mathrm{T}}L_{X_C}C)$ 是从流形对齐[20]推导而得, $tr(\cdot)$ 代表矩阵的秩, $D(D_{ii}=\sum_i X_C(i,j))$ 是正交矩阵, $L_{X_C}=D-X_C$ 是任务类别关系图的拉普拉斯矩阵, $\|S\|^2+\|U\|^2+\|C\|^2+\|G\|^2$ 是防止过拟合的正则项惩罚, λ_1, λ_2, λ_3 控制着 SICTD 中每部分的贡献.

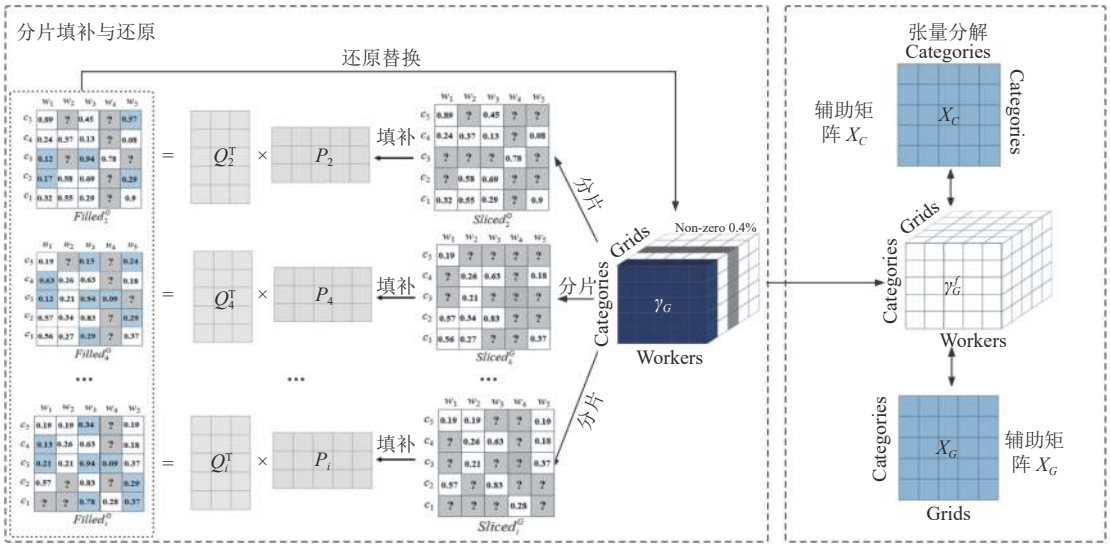

图 3 SICTD 算法示意图

算法 1 展示了在空间维度采用 SICTD 进行张量分解的过程. 我们首先计算原始张量 γ_G 的非零值占比 ϕ, 然后遍历空间维度, 判断每个空间片矩阵的非零值占比是否大于 ϕ, 是则按比例 σ 进行分片填补, 并还原至张量 γ_G 中, 遍历结束后得到分片填充后的空间张量 γ_G^f(第 1–6 行). 接着, 初始化张量分解的核张量 S 和 3 个低秩矩阵 U、C、G, 并且初始化迭代的步长 η、正交矩阵 D 和拉普拉斯矩阵 L_{X_C}. 我们基于 γ_G^f 中的非零值, 采用梯度下降法以迭代的方式最小化损失函数, L_{G_i} 代表第 i 轮迭代的损失函数值. 最后通过公式 (4) 复原 γ_G 中的缺失值, 得到还原后的完整张量 γ_G^{rec} (7–17 行).

算法 1. SICTD Algorithm.

输入: 原始张量 γ_G, 辅助矩阵 X_G, X_C, 迭代精度 ε, 填补比例 σ;
输出: 预测的完整张量 γ_G^{rec}.

1. 计算 γ_G 中的非零值占比 ϕ
2. **For** (每个网格 g)
3. **If** (g 的非零值占比 $\geqslant \varphi$)
4. 通过 Bias-SVD 按 σ 填补 γ_g 并还原至 γ_G 中得到 γ_G^f
5. **Endif**
6. **Endfor**
7. 初始化 S_g、U、C、G, 迭代步长 η, $D_{ii}=\sum_i X_C(i,j)$, $L_{X_C}=D-X_C$
8. **While** ($L_{Gi}-L_{Gi+1}>\varepsilon$)

9. **For** ($\gamma_{Gijk}^{f} \neq 0$)
10. $X_{Gijk} = S_g \times_U U_{i*} \times_C C_{j*} \times_G G_{k*}$
11. $U_{i*} \leftarrow U_{i*} - \eta\lambda_3 U_{i*} - \eta\left(X_{Gijk} - \gamma_{Gijk}^{f}\right) \times S_g \times_C C_{j*} \times_G G_{k*}$
12. $C_{j*} \leftarrow C_{j*} - \eta\lambda_3 C_{j*} - \eta\left(X_{Gijk} - \gamma_{Gijk}^{f}\right) \times S_g \times_U U_{i*} \times_G G_{k*} - \eta\lambda_2\left(L_{X_C} \times C_{j*}\right) - \eta\lambda_3\left(C_{j*} \times G^{\mathrm{T}} - X_{G*j}^{\mathrm{T}}\right) \times G$
13. $G_{k*} \leftarrow G_{k*} - \eta\lambda_3 G_{k*} - \eta\left(X_{Gijk} - \gamma_{Gijk}^{f}\right) \times S_g \times_U U_{i*} \times_C C_{j*} - \eta\lambda_3\left(G_{k*} \times C^{\mathrm{T}} - X_{Gk*}^{\mathrm{T}}\right) \times C$
14. $S_g \leftarrow S - \eta\lambda_3 S_g - \eta\left(X_{Gijk} - \gamma_{Gijk}^{f}\right) \times U_{i*} \otimes C_{j*} \otimes G_{k*}$
15. **Endfor**
16. **Endwhile**
17. **Return** $\gamma_G^{rec} = S_g \times_U U \times_C C \times_G G$

算法 1 同样适用于时间维度. 因此, 给定工人 w, 任务 s, 地理网格 g, 时间段 t, 工人 w 对任务 s 的长期时空偏好为 $LP_{(w,s)} = \gamma_T^{rec}(w, s.c, t) \times \gamma_G^{rec}(w, s.c, g)$, 长期时空偏好将作为工人偏好的一部分用于优化任务分配.

4.2 短期时空偏好

热门的地方会聚集更多的人, 任务的需求量更大, 工人也更有可能接收到新的任务, 因此, 我们认为任务地点的热门程度会对工人的短期偏好产生影响. 我们基于传统 HITS 算法进行改进, 提出 ST-HITS 算法去计算不同时间段地理网格的热门程度. 在 ST-HITS 算法中, 工人和地理网格作为两类不同的节点, 工人节点的 *active* 属性表示工人的活跃度, 工人完成任务的地理网格越多, 其 *active* 值越大, 则近期越活跃; 地理网格节点的 *popularity* 属性表示网格的热门程度, *popularity* 值越大, 该网格可以视为当前时间段热门的地理网格.

给定时间段 t_i 的工人集合 W_i 和地理网格集合 G_i, 定义节点关系矩阵 $M_W \in R^{|W_i|\times|G_i|}$ 和 $M_G \in R^{|G_i|\times|W_i|}$, 用来表示工人与网格节点之间的链接强度, 具体来说, 如果短期内工人 w 完成了网格集合 $G_f^w \in G_i$ 中所有网格的任务, 则视为 w 链接到了 G_f^w 中的每一个网格, 如果短期内网格 g 中有工人集合 $W_f^g \in W_i$ 完成了任务, 则视为 g 链接到了 W_f^g 中的每一个工人, M_W 与 M_G 设置如下:

$$M_W(w,g') = \begin{cases} \dfrac{1}{out_w}, & g' \in G_f^w \\ 0, & \text{otherwise} \end{cases}, \quad M_G(g,w') = \begin{cases} \dfrac{1}{out_g}, & w' \in W_f^g \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

其中, out_w 为工人节点 w 的出度, 即工人 w 所完成任务的网格数量总和, out_g 为网格节点 g 的出度, 即在网格 g 中完成任务的工人数量总和.

算法 2 给出 ST-HITS 的流程. 首先初始化迭代轮数与迭代差值, 同时初始化工人节点的 *active* 值以及网格节点的 *popularity* 值为该节点出入度与节点集合的范数比值, 表示初始状态下每个节点所具有的活跃度与热门程度 (第 1–7 行), 接着不断迭代计算每名工人的 *active* 值与 *popularity* 值. 当工人节点的 *active* 值与网格节点的 *popularity* 值的范数之和小于终止阈值时, 算法迭代结束 (第 8–18 行), 最终得到网格集合在某个时间段 t_i 的热门程度 $G_i.popularity$.

算法 2. ST-HITS algorithm.

输入: 时间段 t_i, 工人集 W_i, 地理网格集合 G_i, 节点关系矩阵 M_W 和 M_G, 终止阈值 ε;
输出: 网格在时间段 t_i 的热门程度 $G_i.popularity$.

1. 初始化迭代轮数 k=1, 迭代差值 *diff=Infinity*
2. **For** (W_i 中的工人 w)
3. $w.active = out_w / norm(out_{W_i})$
4. **Endfor**
5. **For** (G_i 中的网格 g)

6. $g.popularity = out_g / norm(out_{G_i})$
7. **Endfor**
8. **While** ($diff > \varepsilon$)
9. **For** (W_i 中的工人 w)
10. $w.active\ (k) = \sum_{g' \in G_f^w} (M_W(w, g') \times g'.popularity(k-1))$
11. **Endfor**
12. **For** (G_i 中的网格 g)
13. $g.popularity\ (k) = \sum_{w' \in W_f^g} (M_G(g, w') \times w'.active(k))$
14. **Endfor**
15. $diff = \|W_i.active(k) - W_i.active(k-1)\| + \|G_i.popularity(k) - G_i.popularity(k-1)\|$
16. $k = k + 1$
17. 归一化 $W_i.active$ 和 $G_i.popularity$
18. **Endwhile**
19. **Return** $G_i.popularity$

最后, 工人的短期偏好由他的短期活跃范围以及任务位置的热门程度根据公式 (5) 计算而得.

4.3 长短期时空偏好

给定工人 w 对类别为 c 的任务 s 的长期时空偏好 $LP_{(w,s)}$ 以及短期时空偏好 $SP_{(w,s)}$, 工人的长短期时空偏好为:

$$P_{(w,s)} = \rho \times LP_{(w,s)} + (1-\rho) \times SP_{(w,s)} \tag{9}$$

其中, ρ 为长短期权重参数, 决定了长期偏好与短期偏好在总偏好中的占比.

5 任务分配

给定某时间段 t_i 的工人集 $W_i = \{w_1, w_2, \ldots, w_{|W_i|}\}$ 和任务集 $S_i = \{s_1, s_2, \ldots, s_{|S_i|}\}$, 基于第 4 节偏好模型得到的时间偏好张量 γ_T^{rec}、空间偏好张量 γ_G^{rec}、地理网格热门程度 $G_i.popularity$ 与工人短期完成任务记录 L_W, 通过公式 (9) 计算出工人集中工人 w 对任务集中任务 s 的偏好 $pre_{(w,s)}$, 存入工人任务偏好表 P 中用于任务分配. 我们提出基于时空偏好的贪心与 KM 算法完成任务分配.

5.1 基于长短期时空偏好的贪心算法

为了解决 LSPTA 问题, 基于时空偏好的贪心算法优先分配报酬最大的任务, 同时将该任务分配给对其偏好最大的工人, 定义任务 s 的可选工人集合为 $AW(s)$, $AW(s)$ 需要满足 LSPTA 问题定义给出的时空约束. 如算法 3 所示, 初始化任务分配集合为空后, 依据任务收益对任务进行降序排列, 保证优先分配收益最大的任务 (第 1, 2 行), 接着为每个任务生成他的可选工人集 $AW(s)$, 参考偏好表 P, 按照长短期时空偏好降序排列 $AW(s)$ 中的工人, 每次都安排偏好最大的工人完成任务, 实现任务收益与工人偏好最大化的效果. 最终返回时间段 t 的任务分配集合 A_t (第 3–8 行).

算法 3. Preference-aware greedy algorithm.

输入: 时间段 t_i, 工人集 W_i, 任务集 S_i, 工人任务偏好表 P;
输出: 任务分配结果 A_i.

1. 初始化 $A_i = \varnothing$
2. $S_{i\text{sorted}} \leftarrow$ 根据任务收益对 S_i 的任务降序排列

3. **For** ($S_{t\text{sorted}}$ 中的任务 s)
4. 生成 s 的可选工人集 $AW(s)$
5. $AW(s)_{\text{sorted}}$ ←参考 P, 根据 $AW(s)$ 中工人对 s 的偏好降序排列
6. $A_i \cup [(s, AW(s)_{\text{sorted}}[0])]$
7. **Endfor**
8. **Return** A_i

5.2 基于长短期时空偏好的 KM 算法

为了避免贪心算法造成的局部最优, 我们将 LSPTA 问题转化为二分图最大权重匹配问题. 二分图表示为 $G=(V, E)$, V 为图中工人和任务的顶点集, 分为工人顶点 V_W 和任务顶点 V_S, $V_W \cap V_S = \varnothing$, v_i^w 表示工人 w_i 在二分图中对应的节点, v_j^s 表示任务 s_j 在二分图中对应的节点, E 为工人与任务之间的连线边集. 如果工人 w_i 满足某个任务 s_j 的时空约束, 用 (v_i^w, v_j^s) 表示他们之间的可达连线关系. 为了在 KM 算法中综合考虑工人偏好与总报酬, 参考工人任务偏好表 P, 将节点连线权重 $weight(v_i^w, v_j^s)$ 设为工人 w_i 对任务 s_j 的长短期时空偏好 $pre_{(w_i, s_j)}$ 与任务报酬 $s_j.r$ 的带权和, 即 $weight(v_i^w, v_j^s) = w_r \times s_j.r + w_p \times pre_{(w_i, s_j)}$, w_r 与 w_p 分别为报酬权重参数与偏好权重参数, 表示工人偏好与任务收益在任务分配时具有的重要性. 当工人与任务的顶标之和与他们的连线权重相等时, 该分配能同时最大化任务收益以及工人对分配任务的偏好值, 从而在给工人分配最感兴趣任务的同时, 兼顾工人所获的任务报酬. 我们首先提出广度优先任务搜索算法 BFTS (breadth-first task search algorithm), 为每个工人找到可用的任务, 如算法 4 所示.

算法 4. Breadth-first task search algorithm.

输入: 工人 w, 任务集 S, 工人顶标 ex_{worker}, 任务顶标 ex_{task}, 分配表 A;
输出: 是否找到可用任务布尔值.

1. 初始化 $q = deque([w])$, $vis_{\text{task}} = [\text{False}] \times len(S)$, $vis_{\text{worker}}[w] = \text{True}$, $prev = [-1] \times len(S)$
2. **While** (q 非空)
3. $w_{\text{top}} = q.popleft()$
4. **For** (w_{top} 邻接的任务 s)
5. **If** ($vis_{\text{task}}[s]$ 为 False)
6. **If** ($ex_{\text{worker}}[w_{\text{top}}] + ex_{\text{task}}[s] = weight\ (v_{w_{\text{top}}}^W, v_s^S)$)
7. $vis_{\text{task}}[s]$ 置为 True
8. **If** ($A[s]$ 为 False)
9. **While** (True)
10. **If** (w_{top} 为−1)
11. break
12. **Endif**
13. $A[s], w_{\text{top}} = w_{\text{top}}, prev[w_{\text{top}}]$
14. **Endwhile**
15. **Return** True
16. **Endif**
17. $prev[match[s]] = w_{\text{top}}$
18. $q.append(match[s])$
19. **Else**

20. $slack[s] = \min(slack[s], ex_{\text{worker}}[w_{\text{top}}] + ex_{\text{task}}[s] - weight(v^W_{w_{\text{top}}}, v^S_s))$
21. **Endif**
22. **Endif**
23. **Endfor**
24. **Endwhile**
25. **Return** False

首先初始化 vis_{task} 数组用于标记任务节点是否被访问过, 当前搜索起始点 w 置为已访问, 初始化 *prev* 数组与队列 q 分别记录前驱节点与遍历过的节点, 用于广度搜索时回溯增广路径 (第 1 行). 搜索过程将忽略已遍历过的任务节点, 弹出节点队列首元素 w_{top}, 并计算其与尚未遍历的任务节点 s 的顶标和, 判断该值是否等于两点连线的权重, 如果相等且任务 s 还没有被匹配, 则寻找到了一条增广路径, 将 s 分配给工人 w_{top}, 并更新前驱节点, 寻找任务成功 (第 2–15 行), 若任务 s 已被匹配, 则更新任务 s 已匹配节点的前驱节点并将其加入队列进行下一轮广度搜索 (第 17, 18 行). 若顶标和不等于两点连线权重, 则调整顶标差值 *slack*, 用于后续扩大子图范围, 降低匹配难度 (第 20 行). 若未能找到一条增广路径, 则任务分配失败 (第 25 行).

基于长短期时空偏好的 KM 算法如算法 5 所示, 输入为二分图 G、工人集 W_i、任务集 S_i. 初始时, 任务分配结果 A_i 为空集, 每个任务的顶标 ex_{task} 为 0, 顶标差值 *slack* 为无限大, 工人顶标为其连线权重的最大值 (第 1 行). 调用算法 4 广度优先寻找与工人 w 匹配的任务 (第 4 行), 将匹配结果添加至 A_i 中, 并修改 *slack*, 如果任务分配失败, 就要调整 w 和相关任务的顶标, 降低匹配难度 (第 5–24 行). 算法最终会返回任务分配集合 A_i (第 25 行), 其中的工人任务匹配结果全面考虑了工人偏好与任务收益.

算法 5. Preference-aware KM algorithm.

输入: 二分图 G, 工人集 W_i, 任务集 S_i;
输出: 任务分配结果 A_i.

1. 初始化 $A_i = \varnothing$, 任务顶标 ex_{task} 为 0, 顶标差值 *slack* 为 *Infinity*, 工人顶标 $ex_{\text{worker}} = \max(weight(v^W_w, v^S_s))$
2. **For** (工人集 W_i 的每个工人 w)
3. 已访问工人 vis_{worker} 与已访问任务 vis_{task} 置为 False
4. **While** (*BFTS*(w, S_i, ex_{worker}, ex_{task}, A) 为 False)
5. *diff* = *Infinity*
6. **For** (任务集 S_i 的每个任务 s)
7. **If** ($vis_{\text{task}}[s]$ 为 False)
8. *diff* = min(*diff*, *slack*[s])
9. **Endif**
10. **Endfor**
11. **For** (工人集 W_i 的每个工人 w)
12. **If** ($vis_{\text{worker}}[w]$ 为 True)
13. $ex_{\text{worker}}[w]$ − = *diff*
14. **Endif**
15. **Endfor**
16. **For** (任务集 S_i 的每个任务 s)
17. **If** ($vis_{\text{task}}[s]$ 为 True)
18. $ex_{\text{task}}[s]$ + = *diff*

19. **Else**
20. $slack[s] -= diff$
21. **Endif**
22. **Endfor**
23. **Endwhile**
24. **Endfor**
25. **Return** A_i

6 实 验

6.1 实验设置

由于签到数据集中的数据提供了高精度的时空关系, 时空众包任务分配研究通常使用签到数据集来模拟众包平台的实体数据[5−7]. 我们使用两个真实签到数据集来模拟众包平台的任务分配, 分别是 FourSquare 的 NewYork City 数据集 (NYC) 和 Tokyo City 数据集 (TKY)[21]. NewYork City 数据集包含了纽约在 2012 年 4 月 12 日至 2013 年 2 月 16 日期间共 227 428 条签到记录, Tokyo City 数据集包含了东京在 2012 年 4 月 3 日至 2012 年 8 月 22 日期间共 330 216 条签到记录. 我们假设数据集中的用户为众包工人, 将 2 h 内签到 5 次以上的用户, 视为有效用户, 将签到地点相距不超过 2 个邻接网格, 且最近一次签到在其短期活跃范围内的有效用户视为有效的工人参与任务分配. 工人的位置设置为最近一次签到的地点. 同时, 我们将签到地点视为任务, 签到时间设置为任务发布时间, 地点的类别设置为任务类别. 除此之外, 我们设置每个时间段为 15 min, 在当前时间段签到的用户和地点被视为当前时间段的在线工人和新发任务, 进行任务分配. 考虑到工人位置分布与城市邮编区分布相似, 因此在空间层面, 我们按照邮编区对区域进行网格划分. 工人与任务的距离使用欧几里得距离进行计算. 最后, 考虑到现实中高额收益和低额收益的任务总是少数, 因此我们设置每个任务的收益服从 $N(1.5, 0.25)$ 高斯分布. 表 1 给出了数据集的相关统计信息.

表 1 数据集统计信息

数据集	工人数量	任务数量	任务类别数量	网格数量
NYC	5 184	38 333	251	175
TKY	10 693	61 595	240	62

实验的参数在表 2 中列出, 默认值加粗标明. 所有算法都是在 Lenovo R9000X AMD Ryzen 7@2.90 GHz with 16 GB RAM 配置下运行的.

表 2 实验参数

参数	范围
任务数目$\|S\|$	1 000, 2 000, **3 000**, 4 000, 5 000
工人数目$\|W\|$	1 000, 2 000, **3 000**, 4 000, 5 000
工人可达距离d (km)	0.1, 0.5, **1.0**, 3.0, 5.0
任务持续时间$s.e–s.p$ (h)	0.5, 1.0, **1.5**, 2.0, 2.5
分片填补比例σ (%)	0, 0.05, 0.1, 0.2, **0.3**

6.2 评估结果及分析

6.2.1 时空偏好预测模型的性能

将过去 1 周和 4 周的历史数据用于工人长期偏好预测, 参与任务分配的当天作为短期时间. 公式 (7) 的参数

λ_1、λ_2、λ_3均设置为 0.01, 采用 MAE、RMSE 和 MAPE 评估偏好的预测性能. 我们随机移除时空张量中 20% 的非零值作为张量分解的测试集, 剩下 80% 的非零值用作训练集. 实验采用的 5 个对比方法如下.

(1) Random filling (RF): 用 [0, 1] 之间的随机值填充张量中的空缺值, 作为工人偏好预测值.

(2) Average filling (AF): 用当前时间段或地理网格的非零平均值填充空缺值, 作为工人偏好预测值.

(3) Tensor decomposition (TD)[9]: 仅根据原始张量中的非零值进行分解, 不考虑历史记录与上下文辅助矩阵.

(4) HCTD[9]: 基于历史记录和上下文信息 (X_T 与 X_G) 的张量分解.

(5) SICTD: 本文提出的基于分片填补的张量分解, 使用 X_T、X_G 和 X_C 作为辅助矩阵, 共同完成张量分解.

● 时空偏好预测准确性

表 3 和表 4 分别展示了两个数据集上 5 种方法在 1 周和 4 周的历史数据上对时间和空间偏好的预测性能. 可以看出, 我们提出的 SICTD 算法由于采用分片填补解决数据稀疏问题, 并辅以任务类别关联等上下文辅助矩阵, 预测准确性明显高于其他 4 种基线方法. 在 NYC 数据集和 TKY 数据集上, 相比于目前最先进的 HCTD 算法, SICTD 的时间偏好预测的 MAE 分别平均减小了 29.39% 和 22.37%, RMSE 分别平均减小了 21.62% 和 23.51%, MAPE 分别平均减小了 22.29% 和 20.33%; SICTD 的空间偏好预测指标 MAE 分别平均减小了 28.63% 和 28.18%, RMSE 分别平均减小了 27.96% 和 22.25%, MAPE 分别平均减小了 20.33% 和 20.77%.

表 3　时间偏好预测准确性

数据集	对比算法	时间跨度					
		1 week			4 weeks		
		MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)
NYC	RF	0.5809	0.6687	98.32	0.2969	0.4012	96.83
	AF	0.5231	0.6130	91.80	0.2949	0.3981	97.38
	TD	0.4678	0.5903	61.33	0.4379	0.5903	61.33
	HCTD	0.2160	0.3023	58.84	0.1458	0.2097	42.00
	SICTD	**0.1563**	**0.2332**	**43.87**	**0.1004**	**0.1671**	**33.96**
TKY	RF	0.5596	0.6505	98.76	0.3230	0.4395	98.69
	AF	0.5017	0.5945	99.78	0.3255	0.4343	94.61
	TD	0.4585	0.5881	67.77	0.4585	0.5881	67.77
	HCTD	0.2234	0.3404	59.71%	0.1463	0.2223	44.64
	SICTD	**0.1587**	**0.2601**	**44.83**	**0.1232**	**0.1702**	**37.61**

表 4　空间偏好预测准确性

数据集	对比算法	时间跨度					
		1 week			4 weeks		
		MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)
NYC	RF	0.5694	0.6511	125.32	0.4099	0.4929	121.83
	AF	0.5554	0.6710	93.80	0.3949	0.4714	79.38
	TD	0.5074	0.5654	70.25	0.5074	0.5654	70.25
	HCTD	0.2952	0.4063	68.84	0.1477	0.2269	50.33
	SICTD	**0.1951**	**0.2862**	**48.87**	**0.1132**	**0.1756**	**38.91**
TKY	RF	0.5856	0.6746	128.76	0.4156	0.4816	108.63
	AF	0.5715	0.6567	95.73	0.3952	0.4807	90.63
	TD	0.5279	0.5912	73.18	0.5279	0.5912	73.18
	HCTD	0.3306	0.4212	68.54	0.1657	0.2395	53.62
	SICTD	**0.2065**	**0.3231**	**54.15**	**0.1345**	**0.1887**	**42.61**

● 分片填补比例对 SICTD 预测准确性的影响

设置 SICTD 中分片填补比例为 0 至 0.3%, 预测结果如表 5 所示. 分片填补比例增加, 预测指标都下降, 尤其当分片填补比例增加至 0.3% 时, 与未做填补相比, 时间偏好预测的 MAE、RMSE 和 MAPE 分别平均下降

60.37%, 66.62% 和 60.03%, 空间偏好预测的 MAE、RMSE 和 MAPE 分别平均下降 56.75%, 52.94% 和 47.08%. 说明分片填补比例的增加能显著地提升偏好预测的准确性. 从实验结果可以看出, 设置 0.2% 的分片填补比例能平衡预测性能与时间开销, 继续增加也不会过多提升预测性能, 反而增加运行开销.

表 5　分片填补比例对时空偏好预测的影响

数据集	填补比例 (%)	时间偏好			空间偏好		
		MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)
NYC	0	0.1649	0.2567	27.30	0.2153	0.2941	50.46
	0.05	0.1264	0.1848	21.33	0.1634	0.2346	43.82
	0.1	0.0889	0.1213	15.05	0.1364	0.1912	34.35
	0.2	0.0757	0.0960	12.86	0.1085	0.1568	28.69
	0.3	**0.0734**	**0.0928**	**11.90**	**0.0921**	**0.1296**	**27.04**
TKY	0	0.1954	0.2966	32.84	0.2752	0.3638	52.23
	0.05	0.1379	0.2143	22.85	0.2464	0.3272	43.76
	0.1	0.1113	0.1616	18.43	0.1995	0.2617	35.17
	0.2	0.0726	0.0953	12.80	0.1334	0.2197	29.12
	0.3	**0.0679**	**0.0908**	**11.94**	**0.1203**	**0.1821**	**27.29**

6.2.2　基于长短期时空偏好的任务分配性能

我们在两个数据集上对比了 6 个算法, 分别为: 不考虑偏好的 Greedy 和 KM、基于 HCTD 的 Greedy+Pre(HCTD) 和 KM+Pre(HCTD)、基于 SICTD 的 Greedy+Pre(SICTD) 和 KM+Pre(SICTD), 采用以下 5 个实验指标评估任务分配性能: (1) CPU time: 任务分配的运行时间; (2) Reward: 完成任务总收益; (3) Preference score (Prescore): 工人完成任务的平均偏好值; (4) Average travel distance (ATD): 工人平均旅行距离; (5) Assignment success rate (ASR): 任务分配成功率. 同时, 我们还设计了两组消融实验, 进一步评估从长期和短期两个角度以及时间和空间两个维度全面考虑工人偏好进行时空众包任务分配的好处. 实验中, KM 算法的收益和偏好的权重参数 w_r 与 w_p 均设置为 0.5, 长短期偏好的权重参数 ρ 设置为 0.5.

● 偏好预测对任务分配性能的影响.

表 6 给出不同的偏好预测方法对任务分配性能的影响. 无论在 NYC 还是 TKY 上, 基于 SICTD 的算法 (Greedy+Pre(SICTD) 和 KM+Pre(SICTD)) 在工人收益、平均偏好、移动距离和任务分配成功率方面均优于基于 HTCD 的相应算法, 尤以 KM+Pre(SICTD) 表现最佳. 在 NYC 数据集上, 相比于 KM+Pre(HCTD), KM+Pre(SICTD) 因为需要进行张量分片填补, 时间开销增大了约 19 s, 但换来了更优秀的任务分配性能, 工人收益提升了 15.62%, 平均偏好提升了 18.74%, 旅行距离下降了 12.19%, 任务分配成功率提升了 17.75%. 实验结果显示, 依据 SICTD 提供的更加准确的偏好预测进行任务分配, 任务分配质量明显提高, 任务的完成率和执行效率均得到提升.

● 任务分配参数对分配性能的影响.

随机选取 1 000–5 000 个工人参与任务分配, 图 4 给出工人数目对任务分配性能的影响. 可以看出, 工人所获收益、工人平均偏好值以及任务分配成功率均随着工人数目增加而增加, 而工人平均旅行距离随着工人增加而减小, 因为增加工人后, 成功分配任务的可能性增大, 并且任务有更大几率分配给较近的工人. 总体而言, 我们提出的 KM+Pre(SICTD) 相较于对比算法, 各项指标表现最为出色.

随机选取 1 000–5 000 个任务参与任务分配, 图 5 给出任务数目对任务分配性能的影响. KM+Pre(SICTD) 凭借更准确的偏好预测和全局最大匹配的分配方案, 所有指标均优于对比算法. 随着任务数目增多, 工人平均旅行距离减小, 工人平均偏好值增大, 因为任务数目增多后, 工人有更大几率分配到离其位置更近的任务, 增加短期偏好的同时减小旅行开销; 任务分配成功率随任务数目增加而下降, 因为受工人数目以及时空约束的限制, 会有部分新增的任务无法找到合适的工人被放弃. 无论任务数目如何, KM+Pre(SICTD) 仍能凭借出色的偏好预测能力保持最高的任务分配性能.

表 6　偏好预测对任务分配性能的影响

数据集	对比算法	CPU time (s)	Reward	Prescore	ATD	ASR
NYC	Greedy	6.473	1 428	0.295	0.832	0.238
	Greedy+Pre(HCTD)	30.012	1 728	0.403	0.731	0.291
	Greedy+Pre(SICTD)	50.568	**1 964**	**0.478**	**0.626**	**0.313**
	KM	17.853	1 680	0.328	0.823	0.280
	KM+Pre(HCTD)	41.857	2 298	0.512	0.689	0.383
	KM+Pre(SICTD)	60.951	**2 657**	**0.616**	**0.605**	**0.451**
TKY	Greedy	10.234	1 290	0.240	0.942	0.215
	Greedy+Pre(HCTD)	45.083	1 583	0.331	0.757	0.273
	Greedy+Pre(SICTD)	70.152	**1 764**	**0.477**	**0.731**	**0.294**
	KM	26.846	1 488	0.316	0.923	0.248
	KM+Pre(HCTD)	65.094	2 250	0.421	0.722	0.375
	KM+Pre(SICTD)	107.321	**2 583**	**0.553**	**0.639**	**0.432**

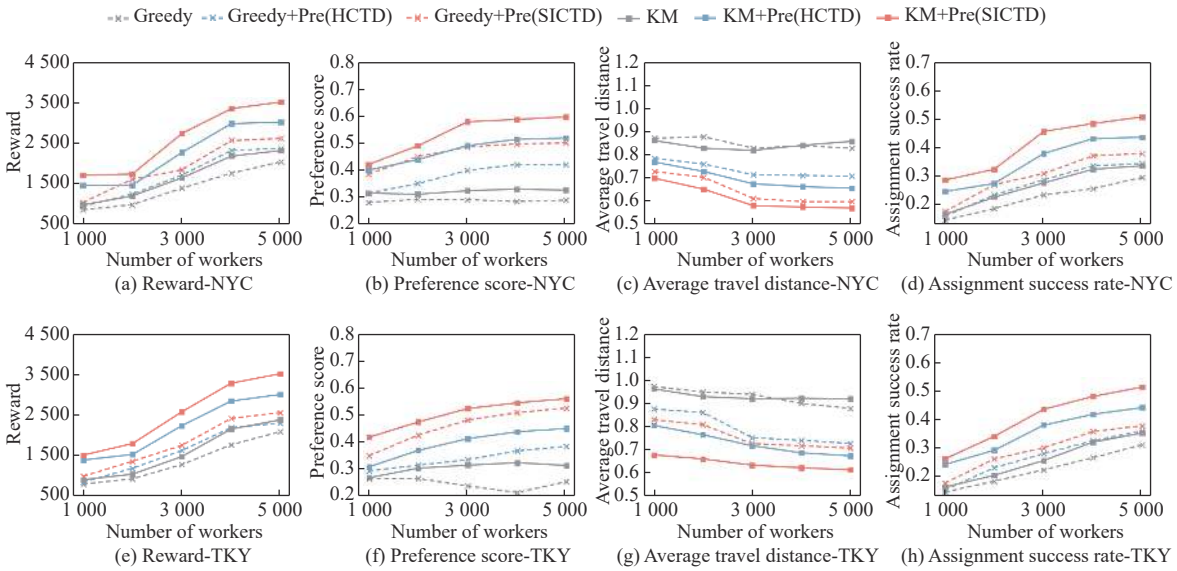

图 4　工人数目对任务分配性能的影响

设置任务持续时间为 0.5–2.5 h, 图 6 给出任务持续时间对任务分配性能的影响. 可以看出, 任务持续时间长于 1.5 h, 两个数据集上工人收益和任务分配成功率趋于稳定, 原因是工人有足够的时间前往任务地点完成任务, 任务持续时间增加不会产生大量新的任务分配结果; 少部分任务因时间约束变宽松, 工人有机会去稍远的地方完成自己更喜欢的任务, 使得平均旅行距离和偏好都增加. 总的来看, 我们提出的 KM+Pre(SICTD) 算法在指标上都优于 KM+Pre(HCTD), 说明 KM+Pre(SICTD) 确实能改善任务分配质量.

设置工人可达距离为 0.1–5 km, 图 7 给出工人可达距离对任务分配性能的影响. 可以发现, 工人可达距离增大时, 工人有更大几率分配到自己喜欢的工作, 使平均偏好指标值增加; 与此同时, 更多工人满足任务的时空约束使任务分配成功率和工人收益增加, 但平均旅行距离会增大. 还可以看出, 工人可达距离大于 3 km 时, 指标趋于稳定, 原因是工人已能够完成远距离的任务, 继续增大可达距离不会产生大量新的任务分配结果. 我们提出的 KM+Pre(SICTD) 始终保持最小的工人平均旅行距离和最大的任务分配成功率, 体现算法最佳的任务分配性能.

● 消融实验

消融实验分两组, 主要验证从长期和短期两个角度以及时间和空间两个维度考虑偏好的必要性. 我们选取偏好预测效果更好的 KM+Pre(SICTD) 进行本实验, 简写为 KM+Pre. 首先设置历史数据时间跨度为 4 周, 分别比较

不考虑偏好 (KM)、不考虑长期偏好 (KM+Pre(w/o long))、不考虑短期偏好 (KM+Pre(w/o short)) 以及综合考虑长短期偏好 (KM+Pre) 对任务分配的影响, 表 7 给出比较结果. 可以看出, KM+Pre(w/o short) 和 KM+Pre 的工人收益和任务分配成功率都大于 KM 和 KM+Pre(w/o long), 原因是凭借长期偏好, 工人可以被分配到更愿意完成的任务. KM+Pre 因为同时考虑了工人长短期偏好, 工人收益和任务分配成功率最大而平均旅行距离最小, 具有最佳的任务分配质量.

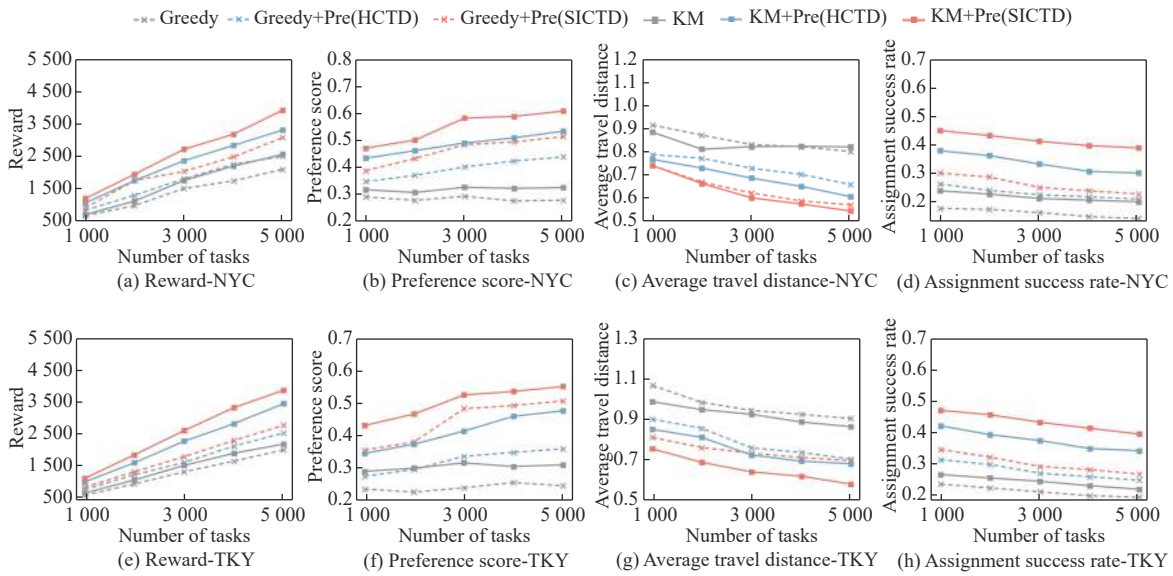

图 5 任务数目对任务分配性能的影响

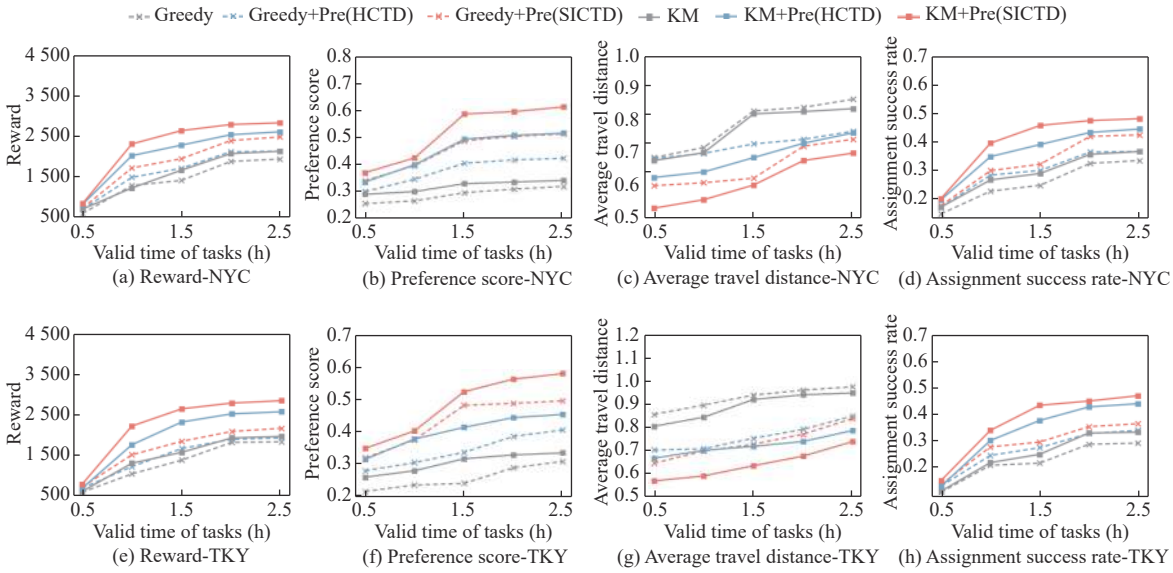

图 6 任务持续时间对任务分配性能的影响

在时空维度上, 分别比较不考虑偏好 (KM)、不考虑时间偏好 (KM+Pre(w/o temporal))、不考虑空间偏好 (KM+Pre(w/o spatial)) 以及综合考虑时空偏好 (KM+Pre) 对任务分配的影响, 结果如表 8 所示. KM+Pre 综合考虑时空偏好, 在两个数据集上的所有指标都优于其他算法, 展现了最佳的任务分配性能.

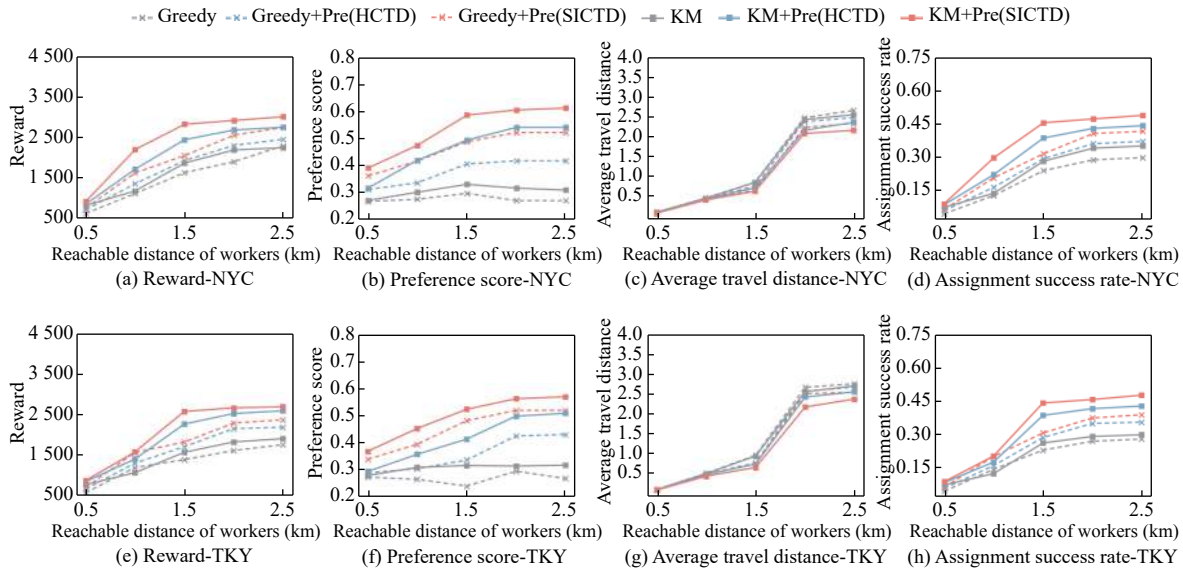

图 7　工人可达距离对任务分配性能的影响

表 7　长短期偏好对任务分配的影响

数据集	对比算法	Reward	Prescore	ATD	ASR
NYC	KM	1 680	0.328	0.823	0.280
	KM+Pre(w/o long)	2 237	0.534	0.687	0.372
	KM+Pre(w/o short)	2 547	0.556	0.755	0.428
	KM+Pre	**2 657**	**0.616**	**0.605**	**0.451**
TKY	KM	1 488	0.316	0.923	0.248
	KM+Pre(w/o long)	18 942	0.451	0.725	0.318
	KM+Pre(w/o short)	2 247	0.530	0.773	0.374
	KM+Pre	**2 583**	**0.553**	**0.639**	**0.432**

表 8　时空偏好对任务分配的影响

数据集	对比算法	Reward	Prescore	ATD	ASR
NYC	KM	1 680	0.328	0.823	0.280
	KM+Pre(w/o temporal)	1 895	0.434	0.674	0.357
	KM+Pre(w/o spatial)	1 921	0.451	0.781	0.369
	KM+Pre	**2 657**	**0.616**	**0.605**	**0.451**
TKY	KM	1 488	0.316	0.923	0.248
	KM+Pre(w/o temporal)	1 774	0.423	0.743	0.339
	KM+Pre(w/o spatial)	1 742	0.406	0.826	0.324
	KM+Pre	**2 583**	**0.553**	**0.639**	**0.432**

7 结　论

移动计算的快速发展孕育了时空众包这种新型的计算模式. 本文研究了时空众包中基于长短期时空偏好的任务分配问题, 从时间和空间两个维度以及长期和短期两个角度综合考虑工人的偏好, 提高任务分配的质量和任务完成率. 我们提出分片填补的张量分解算法 SICTD, 通过填补空缺值的方式使偏好张量更加稠密, 提升长期偏好的预测准确性; 使用 ST-HITS 算法求得地点的热门程度, 综合考虑工人的短期活跃范围, 计算工人的短期偏好. 在

计算长短期时空偏好的基础上, 我们分别采用基于偏好的 Greedy 和 KM 算法确保工人分配到他们最感兴趣的任务, 同时兼顾所获收益. 大量的实验验证了我们提出方法的有效性.

References:

[1] Tong YX, Yuan Y, Cheng YR, Chen L, Wang GR. Survey on spatiotemporal crowdsourced data management techniques. Ruan Jian Xue Bao/Journal of Software, 2017, 28(1): 35−58 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5140.htm [doi: 10.13328/j.cnki.j0s.005140]

[2] Wang DJ, Deng SG, Xu GD. Sequence-based context-aware music recommendation. Information Retrieval Journal, 2018, 21(2): 230–252. [doi: 10.1007/s10791-017-9317-7]

[3] Chen R, Chang YS, Hua QY, Gao QL, Ji X, Wang B. An enhanced social matrix factorization model for recommendation based on social networks using social interaction factors. Multimedia Tools and Applications, 2020, 79(19): 14147−14177. [doi: 10.1007/s11042-020-08620-3]

[4] Ying YK, Chen L, Chen GC. A temporal-aware POI recommendation system using context-aware tensor decomposition and weighted HITS. Neurocomputing, 2017, 242: 195–205. [doi: 10.1016/j.neucom.2017.02.067]

[5] Zhao Y, Xia JF, Liu GF, Su H, Lian DF, Shang S, Zheng K. Preference-aware task assignment in spatial crowdsourcing. Proc. of the AAAI Conf. on Artificial Intelligence, 2019, 33(1): 2629–2636. [doi: 10.1609/aaai.v33i01.33012629]

[6] Zhao Y, Zheng K, Yin HZ, Liu GF, Fang JH, Zhou XF. Preference-aware task assignment in spatial crowdsourcing: From individuals to groups. IEEE Trans. on Knowledge and Data Engineering, 2022, 34(7): 3461–3477. [doi: 10.1109/TKDE.2020.3021028]

[7] Li YC, Zhao Y, Zheng K. Preference-aware group task assignment in spatial crowdsourcing: A mutual information-based approach. In: Proc. of the 2021 IEEE Int'l Conf. on Data Mining (ICDM). Auckland: IEEE, 2021. 350–359. [doi: 10.1109/ICDM51629.2021.00046]

[8] Kazemi L, Shahabi C. GeoCrowd: Enabling query answering with spatial crowdsourcing. In: Proc. of the 20th Int'l Conf. on Advances in Geographic Information Systems. Redondo Beach: ACM, 2012. 189–198. [doi: 10.1145/2424321.2424346]

[9] Ye GY, Zhao Y, Chen XH, Zheng K. Task allocation with geographic partition in spatial crowdsourcing. In: Proc. of the 30th ACM Int'l Conf. on Information & Knowledge Management. Queensland: ACM, 2021. 2404–2413. [doi: 10.1145/3459637.3482300]

[10] Wang ZW, Zhao Y, Chen XH, Zheng K. Task assignment with worker churn prediction in spatial crowdsourcing. In: Proc. of the 30th ACM Int'l Conf. on Information & Knowledge Management. Queensland: ACM, 2021. 2070–2079. [doi: 10.1145/3459637.3482301]

[11] Xia JF, Zhao Y, Liu GF, Xu JJ, Zhang M, Zheng K. Profit-driven task assignment in spatial crowdsourcing. In: Proc. of the 28th Int'l Joint Conf. on Artificial Intelligence. Macao: AAAI Press, 2019. 1914–1920. [doi: 10.24963/ijcai.2019/265]

[12] Cheng P, Lian X, Chen Z, Fu R, Chen L, Han JS, Zhao JZ. Reliable diversity-based spatial crowdsourcing by moving workers. Proc. of the VLDB Endowment, 2015, 8(10): 1022–1033. [doi: 10.14778/2794367.2794372]

[13] Wang Y, Zhao CX, Xu SS. Method for spatial crowdsourcing task assignment based on integrating of genetic algorithm and ant colony optimization. IEEE Access, 2020, 8: 68311–68319. [doi: 10.1109/ACCESS.2020.2983744]

[14] Zhao Y, Li Y, Wang Y, Su H, Zheng K. Destination-aware task assignment in spatial crowdsourcing. In: Proc. of the 2017 ACM on Conf. on Information and Knowledge Management. Singapore: ACM, 2017. 297–306. [doi: 10.1145/3132847.3132894]

[15] Zhao Y, ZHENG K, Li Y, Su H, Liu JJ, Zhou XF. Destination-aware task assignment in spatial crowdsourcing: A worker decomposition approach. IEEE Trans. on Knowledge and Data Engineering, 2020, 32(12): 2336–2350. [doi: 10.1109/TKDE.2019.2922604]

[16] Cheng P, Chen L, Ye JP. Cooperation-aware task assignment in spatial crowdsourcing. In: Proc. of the 35th IEEE Int'l Conf. on Data Engineering (ICDE). Macao: IEEE, 2019. 1442–1453. [doi: 10.1109/ICDE.2019.00130]

[17] Zhao Y, Guo JN, Chen XH, Hao JY, Zhou XF, Zheng K. Coalition-based task assignment in spatial crowdsourcing. In: Proc. of the 37th IEEE Int'l Conf. on Data Engineering (ICDE). Chania: IEEE, 2021. 241–252. [doi: 10.1109/ICDE51399.2021.00028]

[18] Li X, Zhao Y, Zhou XF, Zheng K. Consensus-based group task assignment with social impact in spatial crowdsourcing. Data Science and Engineering, 2020, 5(4): 375–390. [doi: 10.1007/s41019-020-00142-0]

[19] Zhou X, Liang ST, Li KL, Gao YJ, Li KQ. Bilateral preference-aware task assignment in spatial crowdsourcing. In: Proc. of the 38th IEEE Int'l Conf. on Data Engineering (ICDE). Kuala Lumpur: IEEE, 2022. 1687–1699. [doi: 10.1109/ICDE53745.2022.00172]

[20] Ma YQ, Fu Y. Manifold Learning Theory and Applications. Boca Raton: CRC Press, 2012. 314. [doi: 10.1007/978-3-030-03243-2_824-1]

[21] FourSquare—NYC and Tokyo Check-ins. 2016. https://www.kaggle.com/datasets/chetanism/foursquare-nyc-and-tokyo-checkin-dataset

附中文参考文献:

[1] 童咏昕, 袁野, 成雨蓉, 陈雷, 王国仁. 时空众包数据管理技术研究综述. 软件学报, 2017, 28(1): 35−58. http://www.jos.org.cn/1000-9825/5140.htm [doi: 10.13328/j.cnki.j0s.005140]

王府鑫(1998－), 男, 硕士, CCF 学生会员, 主要研究领域为机器学习, 时空众包数据分析.

王宁(1967－), 女, 博士, 教授, 博士生导师, CCF 专业会员, 主要研究领域为人工智能赋能的数据管理, 大数据与群智计算, 数据挖掘.

曾奇雄(1996－), 男, 博士生, 主要研究领域为数据库查询优化, 人工智能赋能的数据管理, 时空众包.