

一种基于时空位置预测的空间众包任务分配方法

徐天承¹ 乔少杰¹ 武俊² 韩楠³ 岳昆⁴ 易玉根⁵ 黄发良⁶ 元昌安⁷

¹(成都信息工程大学软件工程学院 成都 610225)

²(西南财经大学证券与财经学院 成都 610074)

³(成都信息工程大学管理学院 成都 610225)

⁴(云南大学信息学院 昆明 650500)

⁵(江西师范大学软件学院 南昌 330022)

⁶(南宁师范大学计算机与信息工程学院 南宁 530023)

⁷(广西教育学院 南宁 530023)

(ticxu2019@qq.com)

A Spatial Crowdsourcing Task Assignment Approach Based on Spatio-Temporal Location Prediction

Xu Tiancheng¹, Qiao Shaojie¹, Wu Jun², Han Nan³, Yue Kun⁴, Yi Yugen⁵, Huang Faliang⁶, and Yuan Chang'an⁷

¹(School of Software Engineering, Chengdu University of Information Technology, Chengdu 610225)

²(School of Securities and Futures, Southwestern University of Finance and Economics, Chengdu 610074)

³(School of Management, Chengdu University of Information Technology, Chengdu 610225)

⁴(School of Information Science and Engineering, Yunnan University, Kunming 650500)

⁵(School of Software, Jiangxi Normal University, Nanchang 330022)

⁶(School of Computer and Information Engineering, Nanning Normal University, Nanning 530023)

⁷(Guangxi College of Education, Nanning 530023)

Abstract Space crowdsourcing technology has a varying type of application scenarios in the real physical world, which has been widely concerned by academia and industry. Task assignment is one of the important research issues in space crowdsourcing, that is, distributing workers to appropriate tasks. However, most existing task assignment methods assume that the location and time of crowdsourcing workers and space tasks are known, ignoring the dynamic change of crowdsourcing workers and space tasks in real crowdsourcing platforms. Due to the requirement of strong timeliness

收稿日期:2021-08-26;修回日期:2021-11-17

基金项目:国家自然科学基金项目(61772091, 61802035, 61962006, 61962038, U1802271, U2001212, 62072311);四川省科技计划项目(2021JDJQ0021, 22ZDYF2680, 2021YZD0009, 2021ZYD0033);成都市技术创新研发项目(2021-YF05-00491-SN);成都市重大科技建设项目(2021-YF08-00156-GX);四川音乐学院数字媒体艺术四川省重点实验室资助项目(21DMAKL02);CCF-华为数据库创新研究计划项目(CCF-HuaweiDBIR2020004A);广西自然科学基金项目(2018GXNSFDA138005);成都市“揭榜挂帅”科技项目(2021-JB00-00025-GX);四川省科技创新苗子工程项目(2021006)

This work was supported by the National Natural Science Foundation of China (61772091, 61802035, 61962006, 61962038, U1802271, U2001212, 62072311), Sichuan Science and Technology Program (2021JDJQ0021, 22ZDYF2680, 2021YZD0009, 2021ZYD0033), Chengdu Technology Innovation and Research and Development Project (2021-YF05-00491-SN), Chengdu Major Science and Technology Innovation Project (2021-YF08-00156-GX), Digital Media Art, Key Laboratory of Sichuan Province, Sichuan Conservatory of Music, Chengdu, China (21DMAKL02), CCF-Huawei Database System Innovation Research Plan (CCF-HuaweiDBIR2020004A), Natural Science Foundation of Guangxi (2018GXNSFDA138005), Chengdu “Take the lead” Science and Technology Project (2021-JB00-00025-GX), and Science and Technology Innovation Seedling Project of Sichuan Province (2021006).

通信作者:乔少杰(sjqiao@cuit.edu.cn)

in space crowdsourcing platforms, in this situation, the designed assignment method can only get the local optimal results. A new problem of task assignment with maximum value and the minimum cost is proposed, which aims to distribute current and future workers and obtains the maximum assignment value by using the minimum traveling cost. To handle this problem, a trajectory based model is proposed to predict the distribution of tasks. In addition, a kernel density estimation based model is proposed to predict the distribution of workers. A task allocation algorithm based on location prediction is designed to calculate the relative optimal assignment strategy between crowdsourcing workers and spatial tasks. The proposed location prediction method uses graph convolution neural network and ConvLSTM model to predict, which is more accurate and stable than the traditional grid-based location distribution prediction approaches. The heuristic assignment algorithm based on location prediction can complete task allocation in a linear time by combining the predicted location information, which is consistent with the strong timeliness of the space crowdsourcing platform. Extensive experiments are conducted on real datasets to prove the effectiveness of the proposed methods. Compared with the grid-based prediction method, the accuracy of task/worker location prediction is increased by 15.7% and 18.8%, respectively.

Key words spatial crowdsourcing; online task assignment; spatial data intelligence; location prediction; minimum cost

摘要 空间众包技术在现实物理世界中有着丰富的应用场景,得到学术界和工业界的广泛关注。任务分配是空间众包的主要研究问题之一,即把工人分配给合适的任务。但是现有的任务分配方法大多假设众包工人和空间任务出现的位置和时间是已知的,忽略了真实的众包平台中众包工人和空间任务的动态变化,由于空间众包平台的强时效性,这种情况下设计的分配方式只能得到局部最优分配结果。提出最大价值最小成本任务分配的新问题,目标是对当前和未来的工人进行分配,使用最小的移动成本获得最大的分配价值。为解决这一问题,提出了基于轨迹的任务分布预测方法及基于核密度估计的工人分布预测方法,设计基于位置预测的任务分配算法来计算众包工人和空间任务的相对最优分配策略。所提位置预测方法利用图卷积神经网络和 ConvLSTM 模型进行预测,相较传统基于网格的位置分布预测更加精确和稳定。基于位置预测的启发式分配算法可以在线性时间内结合预测得到的位置信息完成任务分配,更加契合空间众包平台的强时效性。在真实数据集上进行大量实验来证明所提方法的有效性,相比于基于网格的预测方法,任务/工人位置预测准确率分别提高了 15.7% 和 18.8%。

关键词 空间众包;在线任务分配;空间数据智能;位置预测;最小成本

中图法分类号 TP311

众包的概念源于“外包”、“承包”,外包与承包强调的是专业化团队长期稳定的合作,而众包则不同,将复杂的任务拆分为细粒度的任务,调动个体用户积极参与,寻求跨专业领域知识为工作成果带来创新^[1]。当前移动和可穿戴设备的普及不仅丰富了人们的日常生活,同时开辟了一种新的工作模式:空间众包(spatial crowdsourcing)。空间众包相较传统众包,需要众包工作者在真实世界中,物理移动到任务所标注的特定地点,完成任务要求,这一模式在生活中存在广泛的应用场景(例如:滴滴打车^[2]、美团众包),它可以帮公司企业节省大量成本。

空间众包框架中任务请求人可以通过众包平台发布空间任务,例如观察咖啡店是否拥挤、监控某条路段的交通状况等,并交付给真实世界中的众包工作者来完成空间任务。服务器需要基于空间任务和众包工人的位置以及其他约束对其进行分配,这一过程被称为众包任务分配问题。在真实的空间众包平台中任务和工人都是动态出现的,如果不考虑任务/工人的动态性,则无法获得最佳的分配结果。而现实生活中众包任务的发布和众包工人的登录往往具有非常复杂的动态规律,使得传统的机器学习和统计学方法难以对任务/工人的发布分布进行准确预测。

通过分析滴滴出行分享的 GAIA 数据集^[2], 可以发现每天同一地点的任务发布数量均是随时间呈周期变化的. 如图 1 所示, 该图统计了以各兴趣点 (point of interest, POI) 为中心, 半径 500 m 内每日不同时间发生的任务数量. 横轴为按小时划分的一天, 纵轴为任务发布的数量. 每个种类的线均有 7 条, 代表一周内的 7 天, 每种线后的阴影区域代表该 POI 处发布任务数量随时间变化的趋势. 从图 1 中

可以发现, 景点 2 附近在 19 点时大概率会出现任务数量的下降, 而景点 1 却没有这种现象; 火车站每日的任务数量高峰期在 16~18 点产生, 而景点 1 一般在 19~23 时达到高峰. 这表明, 同一区域每日任务发布的数量具有相似的变化走势. 通过上述分析, 可以得出结论: 进行未来时刻任务时空分布预测时需要依据过去的任务分布, 将时间和空间作为预测所需的特征学习.

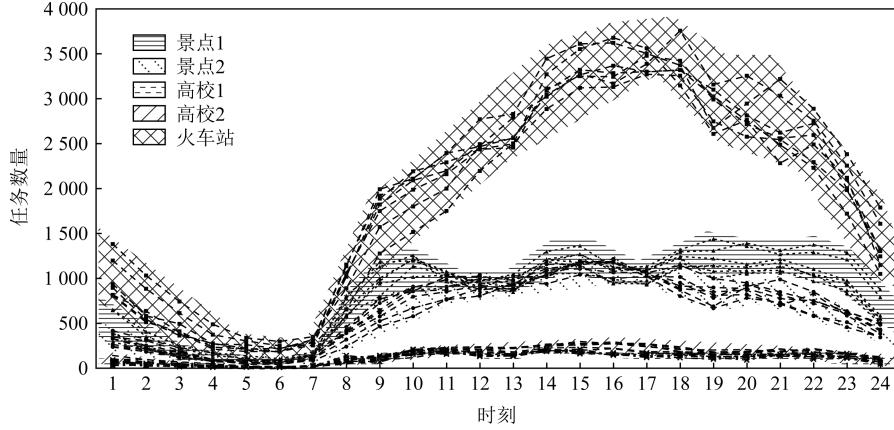


Fig. 1 Quantity trend of different released POI tasks

图 1 不同 POI 任务发布数量趋势图

目前已有许多关于空间众包任务分配的研究, 但现有的空间众包任务分配方法通常存在 2 个方面的局限, 有待进一步完善.

1) 传统的任务/工人分布预测方法很少使用工人和任务的时空特征, 且一般基于网格进行预测, 但网格的不同划分会影响到预测结果的准确性, 并且很难找到最优的网格划分方法. 本文提出基于轨迹的任务分布预测方法和基于二维核密度估计的工人分布预测方法, 学习历史数据中任务/工人的时空特征, 实现更精确的任务分配.

2) 空间众包需要较快的任务分配效率, 面对大规模的时空数据, 计算分配的时间代价过大导致结果没有应用价值. 本文提出了一种基于任务/工人预测进行任务分配的方法, 以最大化目标函数为目的, 可以在线性时间内得到有效的任务分配结果.

针对上述不足, 本文对在线情景下空间众包任务分配问题进行了研究, 设计了一种基于位置预测的空间众包任务分配方法, 可以有效地计算全局最优的任务分配问题. 本文的主要贡献有 3 个方面:

1) 提出了基于轨迹的任务分布预测模型. 不同于网格划分的整体分布预测方法, 本文基于拐点划分轨迹, 使用轨迹聚类算法获得高流量的热点轨迹,

基于轨迹使用图卷积来捕获任务分布的空间特征, 并使用核密度估计预测任务在轨迹上发布的具体位置, 给出了计算预测所得任务价值的方法, 方便后续任务分配时寻找最优解.

2) 提出基于核密度估计的工人分布预测模型. 先通过二维核密度估计来表示空间任务的分布, 使 ConvLSTM 能更好地提取任务分布的空间特征, 对于预测得到的空间分布核密度估计矩阵, 提出了一种基于消减数量的方法来还原真实的任务分布.

3) 提出了一种基于位置预测的任务分配方法. 依据任务和工人间的位置关系, 计算每个任务的可用工人集, 以此来创建二分图, 设定搜索增广路径上限次数, 优先选择权值高的任务进行分配, 并针对预测任务设置了单独的权值比较方法, 能在多项式时间复杂度内寻找尽可能好的任务分配组合.

1 相关工作

与传统众包技术不同, 空间众包需要基于位置信息进行工人任务间的分配^[3]. 按照任务复杂度可将任务分为原子任务和复杂任务 2 种^[4], 原子任务可视为能由一位工人完成的简单任务, 复杂任务

可以划分为多个原子任务,需要多种特定技能才能完成,本文将任务设定为原子任务.按照任务发布模式,可分为服务器分配任务(server assigned tasks)模式和工人选择任务(worker selected tasks)模式2种,现有的空间众包系统大多使用服务器分配任务模式,在该模式中由服务器将任务分配给适当的工人,这种方法相较后者可以尽可能地减少无人接受的任务,并实现一些系统优化目标,如:技能覆盖且成本最小^[5].按任务的物理位置,可以分为特定点、特定区域和特定路线3种^[4],这3种任务位置种类分别要求:到达一个特定的坐标点,到达特定的建筑区域,从起始点提取特定物品交付给目标客户.为了简化问题,现有的研究大多是基于特定点的众包任务研究.按照服务器分配策略中数据的状态,可以分为离线场景和在线场景^[6],离线场景下服务器在一开始就可以获取完整的信息,在线场景下工人和任务的登录信息是动态的,算法将以数据流的形式接受工人和任务的登录信息.现有的大部分研究都是基于离线场景的假设,没有考虑到空间众包任务分配的时空特性,不能得到全局最优分配结果.

文献[7]首先提出了空间众包平台上的任务分配问题,设定服务器统一管理所有的任务和工人位置信息,以最大化任务分配数量为目的利用贪心策略将任务分配给各个工人.文献[8-11]仅考虑了当前系统中存在的工人和任务,忽略了动态变化的任务和工人,无法得到最优的分配策略.文献[12]最大化工人的收入为目标,并针对任务的动态性设计了一种平衡影响因素的路线推荐算法,但是该算法没有考虑动态加入系统的工人.文献[13]提出了一种基于位置熵最小优先级的候选工人算法,通过计算曼哈顿距离获得候选工人,再基于决策树进行任务分配;文献[14]考虑到工人是动态移动的,设计了贪心算法和极值算法2种近似算法,并提出了一种基于反馈的合作机制进行任务分配;但文献[13-14]均忽略了动态加入系统的任务.文献[15-16]以在线算法(online algorithm)的方式来解决新加入系统的空间众包任务分配问题.文献[15]考虑工人—项目—任务的三色效用,设计了基于skyline查询的解决方案;文献[16]提出了一种基于空间感知多智能体的动态优化算法,并通过基于网格的空间优化方法来分解众包问题,以便解决大规模数据;但这些方法的时间复杂度较高,难以应用于真实的众包平台.文献[17]通过预测的方式预估工人和任务在未来时刻的空间分布,基于预测得到的数据以最大化分配质量

分数为目标,基于贪心算法进行分配.文献[18]提出了三阶段的启发式算法求解最佳任务分配,通过预估每个工人的可用时间预测工人未来时刻的轨迹,尽可能地最大化工人完成任务的数量,使得工人的移动距离均匀.但上述基于预测的方法没有考虑到任务的分布与时间变化的关系,且在分布预测中均使用基于网格的预测方法.本文将以最大化分配价值,最小化工人移动距离为目标设计任务分配方法.

2 问题描述

本文要解决的问题中包含2种实体对象,在一个时间实例集 $P = \{1, 2, \dots, p, \dots\}$ 中,实体定义如下:

定义1. 空间任务 t . 表示为 $t = \langle t.s, t.l, t.e, t.v \rangle$ 是一个在 $t.s$ 时刻发生 $t.l$ 位置的任务,该任务在 $t.e$ 时刻前完成被认为是有效的.完成该任务后会产生 $t.v$ 的价值,其中 $t.l$ 为真实空间中的坐标 (x, y) ,实验中将其视作二维数据空间的一个点.

定义2. 众包工人 w . 表示为 $w = \langle w.l_p, w.v \rangle$,表示一个在时刻 p 位置在 $w.l_p$ 处的众包工人,其移动速度为 $w.v$.为简单且不失一般性,本文假设每个空间任务只需要分配一个众包工人,任务需要的处理时间为0,即众包工人只要到达任务处即完成任务.

定义3. 最大价值最小成本任务分配问题. T_p 为时刻 p 系统中现有的空间任务和预测得到的空间任务组成的集合; W_p 为时刻 p 系统中现有的众包工人和预测得到的众包工人组成的集合; I_p 为时刻 p 的任务分配集,由一组有效的 $\langle t, w \rangle$ 分配对组成,其中 $t \in T_p, w \in W_p$,分配对 $\langle t, w \rangle$ 需要满足工人能在任务截止时间前到达任务所在的位置,且同一时间一个工人只能被分配到一个任务.基于任务和工人相关定义,最大价值最小成本任务分配问题可以定义为:在给定的时间实例集 P 中,最大价值最小成本任务分配问题的目标为

$$\text{maximize} \sum_{p \in P} \sum_{\langle w_i, t_j \rangle \in I_p} t_j.v, \quad (1)$$

$$\text{minimize} \left(\sum_{p \in P} \sum_{\langle w_i, t_j \rangle \in I_p} \text{dist}(t_j.l, w_i.l_p) \right) / |I_p|, \quad (2)$$

其中, $\text{dist}(t.l, w.l_p)$ 表示任务 t 到工人 w 的欧氏距离,真实情景下的轨迹分布难以类比为“棋盘格”状,利用曼哈顿距离计算得工人距离并不准确,本文计算 $\text{dist}(t.l, w.l_p)$ 是为了对比不同分配间距离的

相对大小进行择优,使用欧氏距离计算的效果更准确,故本文选择使用欧氏距离作为不同分配对间的成本度量。 $|I_p|$ 表示分配对的个数,最大价值最小成本任务分配问题的目标是最大化分配集 I_p 的价值和,并最小化分配集 I_p 中众包工人到空间任务的平均距离。

3 基于轨迹的任务分布预测模型

空间众包中任务的出现受物理世界中时间和空间因素影响,其中任务请求人的行为模式在任务分配中发挥关键作用。空间众包平台中,众包工人接收任务后需要移动到任务标注的位置,可以认为众包任务的发布位置依附于众包工人的轨迹,分析数据集可知不同地点的任务发生数量按时间呈周期性的模式变化。本节给出一种基于图卷积神经网络的模型,该模型利用轨迹聚类捕捉任务分布时间空间相关性,可以高效地估计未来的任务分布及数量。

3.1 基于轨迹的位置信息提取

受到 Lee 等人^[19]提出的分段及归组框架通过轨迹挖掘得到路网信息的启发,本节通过基于特征点的街角提取(characteristic point-based road corner extraction, CP-RCE)选取工人轨迹中的拐点,使用拐点划分轨迹,并设定轨迹间的距离函数使用 DBSCAN 算法得到处理后的轨迹图。此外,分时段选出任务发生数量高于阈值的轨迹,作为预测任务发布位置的基础,通过后续模型学习其中的空间特征进行任务分布的预测。由图 1 的分析可知每日不同 POI 中任务发生的数据随时间变化的趋势是相似的,可以认为每日相同时段有相同任务分布模式。假设前后两日拥有相似的任务分布模式,那么在第 a 日的 p 时使用的图结构是基于第 $a-1$ 日的 p 时的数据生成的。本文提出的基于轨迹的位置信息提取方法包含如下 2 个关键步骤。

1) 提取轨迹拐角。通过分析数据集中的 GPS 轨迹,使用基于特征点的街角提取方法提取轨迹中的拐角。具体的说,CP-RCE 方法以定宽的窗口来遍历 GPS 轨迹,在中心点的左右通过线性拟合的方法生成 2 条直线,通过计算轨迹点在两直线上投影间的夹角,夹角大于阈值的轨迹点就是代表拐角的特征点。

图 2(a)所示为某一时刻系统中一部分的 GPS 轨迹分布示例,通过基于特征点的街角提取方法提取轨迹中的拐角,以宽度为 w 的窗口遍历轨迹点。以轨迹点 l_p 为例,对窗口左侧轨迹点 $\{l_{p-w}, l_{p-w+1}, \dots,$

$l_p\}$ 和右侧轨迹点 $\{l_p, l_{p+1}, \dots, l_{p+w}\}$ 分别进行线性拟合,得到图内所示的 2 条虚线 L_1 和 L_2 。然后,将点 l_{p-w} 和 l_p 投影到 L_1 上, l_{p-w} 和 l_p 在 L_1 上的投影点分别为 l'_{p-w} 和 l'_p ;将点 l_p 和 l_{p+w} 投影到 L_2 上, l_p 和 l_{p+w} 在 L_2 上的投影点分别为 l''_p 和 l'_{p+w} 。最后,计算线 $l'_{p-w}l'_p$ 和线 $l''_pl'_{p+w}$ 之间的夹角,如果计算得到的夹角大于阈值,则认为该点为道路拐点。值得注意的是,如果连续多个点都被认为是道路拐点,则说明当前拐角路段较长,这种情况下选择其中夹角最大的点代表该拐角。图 2(a)经过拐角提取后得到了由虚线框标注出的 5 个拐点,可由提取的拐点将轨迹划分为 7 个路段,如图 2(b)图所示。因为大量重复的轨迹和 GPS 的误差,在对真实 GPS 数据挖掘时,会出现很多本该是相同的路段的轨迹,故本文使用线段聚类(line segment clustering)^[20]算法进行轨迹聚类得到各路段信息。为防止聚类后的轨迹拐点位置出现偏差,将位置距离在 20 m 内的点视作同一拐点。通过这种方式得到的轨迹道路和真实道路有较大偏差,生成的轨迹网络不能称为路网。

2) 构建轨迹图。由上述步骤得到的道路网络结构无法以矩阵和向量形式表达,很难以传统的神经网络模型来捕捉各个路段的空间依赖关系,故使用这些路段构建图 $G = (V, E)$,以图的形式来存储轨迹信息。图 G 中的 V 为结点集,将轨迹聚类后得到的每一条轨迹视作一个结点,故轨迹的个数就是图 G

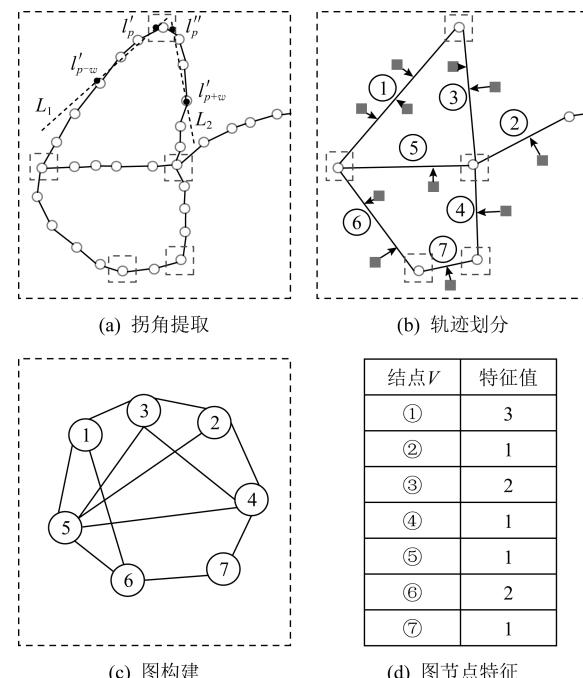


Fig. 2 The process of graph construction

图 2 图构建过程

中结点的个数; E 为边集, 如果 2 条轨迹间有连接, 则两结点间就有边.

图使用邻接矩阵和特征矩阵存储. 邻接矩阵表示各结点间的连接关系, 邻接矩阵的值为 0 或 1, 如果两结点之间有连接则为 1, 反之则为 0. 特征矩阵包含各结点代表路段的特征信息, 数据空间中众包任务视作发生在最近的轨迹路段, p 时结点的特征值为 p 时内轨迹路段上发生的任务数量. 图 2(b)图所示的轨迹被 5 个拐点划分为 7 条路段, 分别被标记为 ①~⑦, 方块代表空间众包任务. 最终生成的图结构如图 2(c)所示, 将各任务划分到距离最近的轨迹路段上, 可得到各结点的特征值如图 2(d)表格所示. 如果图有很多结点的特征长时间为 0 的结点, 会影响模型的效果. 为提高模型预测的效果, 在每一时间段筛选掉任务发布数低于阈值的轨迹路段, 将这些轨迹代表的结点从该时段中删除.

3.2 基于图卷积的任务分布预测模型

本节将基于 3.1 节得到的图结构数据通过基于图卷积的任务分布预测模型来预测路段的流量, 再

使用核密度估计的方法预测各路段任务发生的具体位置, 最后给出了预测要发生价值的方法. 具体为如下 3 个步骤:

1) 任务预测模型网络结构. 图 3 所示为任务分布预测模型的结构图, 图卷积神经网络可以有效提取图结构数据的空间特征, 结合各结点的空间特征和以时间为序列的特征值输入, 可以预测每个结点在未来时刻的任务数量. 图 3 中的输入层需要以图的特征矩阵和邻接矩阵形式输入 3.1 节构建图, 其中特征矩阵需要以时序结构进行输入. 具体地讲, 图构建完成后, 整理图结构数据的邻接矩阵 \mathbf{A} , 然后将各结点的特征值按固定窗口宽度分为时间序列数据 $\mathbf{X}_v = \{x_v^{p-s}, x_v^{p-s+1}, \dots, x_v^p\}$, \mathbf{X}_v 表示结点 v 输入的时间序列, x_v^p 表示结点 v 在时刻 p 的属性, s 表示历史数据窗口宽度. 输入层将输入数据转化为图的邻接矩阵和所有结点的 \mathbf{X}_v 组成的特征矩阵时间序列. 输入层输出的数据会经过 2 个隐藏层提取空间特征, 该过程如图 3 的隐藏层 1 和隐藏层 2 所示. 网络中各属性的形式化表达如式(3)~(7)所示:

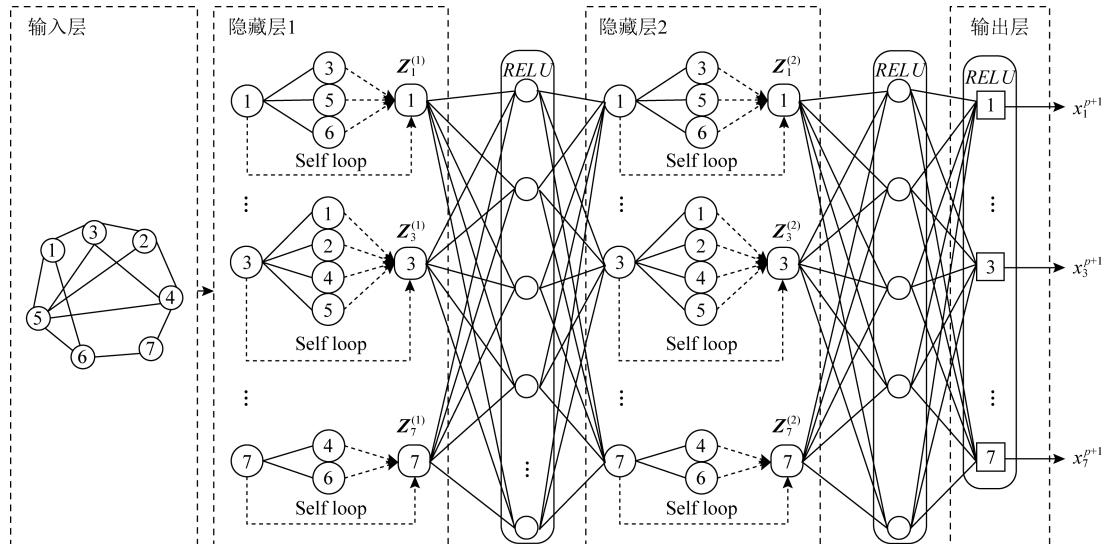


Fig. 3 Prediction model of task distribution

图 3 任务分布预测模型

$$\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}, \quad (3)$$

$$\mathbf{L} = \mathbf{D}^{-1/2} \cdot \tilde{\mathbf{A}} \cdot \mathbf{D}^{-1/2}, \quad (4)$$

$$\mathbf{Z} = \text{ReLU}(\mathbf{L} \cdot \text{ReLU}(\mathbf{L} \cdot \mathbf{X} \cdot \mathbf{W}^{(1)}) \cdot \mathbf{W}^{(2)}), \quad (5)$$

$$\mathbf{Z}_u^{(1)} = \text{ReLU} \left(\sum_{v \in N_u} \mathbf{L} \cdot \mathbf{X}_v \cdot \mathbf{W}^{(1)} \right) \quad (u = 1, 2, \dots, k), \quad (6)$$

$$\mathbf{Z}_u^{(2)} = \text{ReLU} \left(\sum_{v \in N_u} \mathbf{L} \cdot \mathbf{Z}_v^{(1)} \cdot \mathbf{W}^{(2)} \right) \quad (u = 1, 2, \dots, k), \quad (7)$$

其中, \mathbf{A} 为邻接矩阵, \mathbf{I} 为单位矩阵, $\tilde{\mathbf{A}}$ 表示添加自循环后的邻接矩阵, \mathbf{L} 表示拉普拉斯矩阵, \mathbf{D} 为 \mathbf{A} 对应的度矩阵为 $\mathbf{W}^{(l)}$ 为第 l 层的权重, N_u 为结点 u 的所有邻居(包含结点 u 自身), \mathbf{X} 为结点在输出层的特征矩阵, \mathbf{X}_v 为结点 u 的邻居结点的特征, \mathbf{Z} 为输出层的结点特征矩阵, $\mathbf{Z}_u^{(1)}$ 为结点 u 在隐藏层 1 输出的特征, $\mathbf{Z}_v^{(1)}$ 为结点 u 在隐藏层 2 中邻居结点的特征, $\mathbf{Z}_u^{(2)}$ 为结点 u 在隐藏层 2 输出的特征, k 为图结构的结点数. 输入层输入特征矩阵和邻接矩阵

后,隐藏层1中会通过邻接矩阵 \mathbf{A} 计算图的拉普拉斯矩阵 \mathbf{L} ,然后通过图卷积网络提取了图结构的结点特征 $\mathbf{Z}_u^{(1)}$.通过隐藏层2中的第2层图卷积网络可以提取最终的结点特征 \mathbf{Z} .最后,通过输出层的全连接层,结合隐藏层提取的图结构结点特征和时序特征矩阵,可以梯度下降学习并预测各结点的下一时刻发布的任务数量 x_u^{p+1} .根据文献[21],2层图卷积神经网络的效果一般是最优的,这也是本文选择2层图卷积结构的原因.

2) 预测任务具体位置.任务分布预测模型可以得到未来时刻各轨迹路段发布的任务数量,但轨迹路段的长度让其不能被视作一个坐标点,任务发布在轨迹路段的不同位置(例如左端和右端)对后续任务分配的影响非常巨大.核密度估计(kernel density estimation, KDE)常用于根据已知的数据估计位置的密度函数,使用KDE可通过部分数据对估计量的发生概率作出判断,相较于认定任务发生在历史任务发生位置的做法可以减小预测误差,故选择KDE通过各轨迹路段的历史任务位置来估计未来时刻任务在轨迹路段上各位置发布的概率.具体地说,任务分布预测模型得到下一时刻路段发布的任务数为 m ,假设该路段历史有 n 个任务发生,历史任务的位置集合为 $\{l_1, l_2, \dots, l_n\}$ (其中 l 表示任务坐标),核密度估计函数为 $f_t(l) = 1/(nh) \times \sum_{i=1}^n K((l - l_i)/h)$,可以使用核密度估计方法计算该路段任务发生位置的概率密度,其中 $K(\cdot)$ 是核函数, n 表示历史样本点个数, $h > 0$ 为一个平滑参数,表示核密度估计过程中的带宽,核密度估计的带宽 h 使用平均积分平方误差(MISE)最小反推其取值,最后推得的式(8)^[22]为

$$h = \{R(K)\}^{(1/5)} / \{m_2(K)\}^{(2/5)} R(f'')^{(1/5)} n^{(1/5)}, \quad (8)$$

其中, $R(K), m_2, f''$ 均由选择的核函数确定,依据中心极限定理,复杂综合的概率分布呈正态分布,而高斯函数是正态分布的密度函数,故此处选择高斯核函数 $K(l) = (2\pi)^{-(1/2)} \exp(-(1/2)l^T l)$,在选择高斯核函数的情况下 $h \approx 1.06\sigma n^{-(1/5)}$.

截选图3(b)中的轨迹路段①如图4(a)所示,轨迹①在历史上有3个任务发生,将这些任务投影到路段①上,可以得到3个投影点的坐标为 $l_{t_1}, l_{t_2}, l_{t_3}$.使用核密度估计函数 f_t 计算可以得到任务发生位置的概率密度函数 $f(x)$ 如图4(b)所示.假设预测得到下一时刻任务发布的数量为 m ,路段从上至下的两侧的端点坐标为 $(x_e, y_e), (x_s, y_s)$,概率密度函数 $f(x)$ 前 m 高的横坐标集合 $X = \{x_1, x_2, \dots, x_m\}$,则预测得到的 m 个任务发布坐标 (x_p, y_p) 可通过式(10)得到:

$$d = \sqrt{(x_s - x_e)^2 + (y_s - y_e)^2}, \quad (9)$$

$$(x_p, y_p)_i = \left(\frac{x_i |x_s - x_e|}{d}, \frac{y_i |y_s - y_e|}{d} \right), \quad (10)$$

其中, d 为轨迹路段的长度, (x_s, y_s) 和 (x_e, y_e) 分别表示轨迹路段左端点和右端点的坐标, d 由轨迹路段两端的坐标求欧氏距离计算; $(x_p, y_p)_i$ 为轨迹路段上预测得到的第 i 个点的坐标, i 的取值为1到 m .

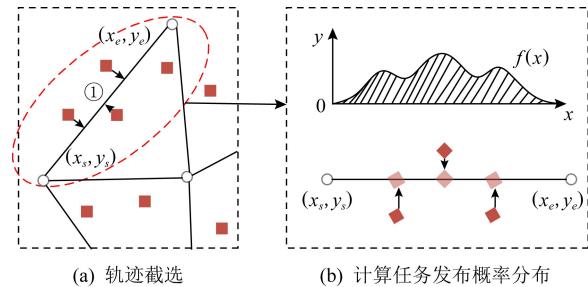


Fig. 4 Diagram of specific location prediction of task publishing

图4 任务发布具体位置预测示意图

3) 预测任务价值.在本文描述问题中,任务拥有价值属性 $t.v$,后续的任务分配过程会基于各任务的价值做出更优的分配,故预测的任务也需要价值属性,本节提出一种预估预测任务价值的方法.首先设 $T_{r.v} = \{t_{1.v}, t_{2.v}, \dots, t_{m.v}\}$ 为轨迹路段 r 历史任务的价值集合,本文假设目标轨迹 r 中任务价值服从高斯分布 $N(\mu_r, \sigma_r^1)$ 模型,可以推得轨迹 r 中任务价值的均值服从高斯分布 $N(\mu_r, \sigma_r^2 / |T_{r.v}|)$,其中 μ_r 是轨迹路段 r 历史任务价值的均值, σ_r 是轨迹路段 r 历史任务价值标准差, $|T_{r.v}|$ 表示轨迹 r 历史任务的数量. μ_r, σ_r 可由极大似然估计计算,如式(11)、式(12)所示,故 $N(\mu_r, \sigma_r^2 / |T_{r.v}|)$ 的累计分布函数(probability density function, PDF)如式(13)所示.

$$\mu_r = \frac{1}{|T_{r.v}|} \sum_{i=1}^m t_{i.v}, \quad (11)$$

$$\sigma_r = \sqrt{\frac{1}{|T_{r.v}|} \sum_{i=1}^m (t_{i.v} - \mu_r)^2}, \quad (12)$$

$$\begin{aligned} PDF\left(x, N\left(\mu_r, \frac{\sigma_r^2}{|T_{r.v}|}\right)\right) = \\ \frac{1}{\sqrt{2\pi\sigma_r^2 / |T_{r.v}|}} \exp\left(-\frac{(x - \mu_r)^2}{2\sigma_r^2 / |T_{r.v}|}\right). \end{aligned} \quad (13)$$

任务价值的预估可分为如下 3 种情况:

1) 同一轨迹上预测任务的价值比较.如果任务 t_p 为轨迹 r 中预测要发生的任务,任务 t'_p 为轨迹 r 中预测要发生的另一个任务,认为两任务价值相当,如式(14)所示:

$$t_p.v = t'_p.v. \quad (14)$$

2) 已发布任务和预测任务价值比较.任务 t_i 为已发布的任务,任务 t_p 为轨迹 r 中预测要发生的任务.本文将高斯分布的置信度大于 80% 设定为可信,则 $\int_{-\infty}^{t_i.v} PDF(t_i.v, N_r(\mu_r, \sigma_r^2 / |T_r.v|)) dt_i.v < 0.2$ 的情况下认为任务 t_i 的价值要小于任务 t_p 的价值, $\int_{-\infty}^{t_i.v} PDF(t_i.v, N_r(\mu_r, \sigma_r^2 / |T_r.v|)) dt_i.v > 0.8$ 的情况下认为任务 t_i 的价值要大于任务 t_p 的价值,通过查询正态分布表可得到两任务价值大小情况如式(15)所示:

$$\begin{cases} t_i.v > t_p.v, t_i.v \geq \mu_r + 0.85\sigma_r / \sqrt{|T_r.v|}, \\ t_i.v = t_p.v, \text{ 其他,} \\ t_i.v < t_p.v, t_i.v \leq \mu_r - 0.85\sigma_r / \sqrt{|T_r.v|}. \end{cases} \quad (15)$$

3) 不同路段上预测任务的价值比较.任务 t_p 为轨迹 r_1 中预测要发生的任务,任务 t''_p 为轨迹 r_2 中预测要发生的任务.假设 2 条轨迹上任务价值均值服从的高斯分布模型分别为 $N(\mu_{r_1}, \sigma_{r_1}^2 / |T_{r_1}.v|)$ 和 $N(\mu_{r_2}, \sigma_{r_2}^2 / |T_{r_2}.v|)$, 可计算得分布概率密度函数的期望 E 如式(16)所示,因为价值 $v > 0$,所以期望 $E(N(\mu_r, \sigma_r^2 / n))$ 的积分是从 0 而不是 $-\infty$ 开始.通过比较两分布概率密度函数的期望 E 可得到该情况下两任务价值的大小情况如式(17)所示.

$$E\left(N\left(\mu_r, \frac{\sigma_r^2}{|T_r.v|}\right)\right) = \int_0^{+\infty} x \times PDF\left(x, N\left(\mu_r, \frac{\sigma_r^2}{|T_r.v|}\right)\right) dx, \quad (16)$$

$$\begin{cases} t_p.v = t''_p.v, E\left(N\left(\mu_{r_1}, \frac{\sigma_{r_1}^2}{|T_{r_1}.v|}\right)\right) = E\left(N\left(\mu_{r_2}, \frac{\sigma_{r_2}^2}{|T_{r_2}.v|}\right)\right), \\ t_p.v > t''_p.v, E\left(N\left(\mu_{r_1}, \frac{\sigma_{r_1}^2}{|T_{r_1}.v|}\right)\right) > E\left(N\left(\mu_{r_2}, \frac{\sigma_{r_2}^2}{|T_{r_2}.v|}\right)\right), \\ t_p.v < t''_p.v, E\left(N\left(\mu_{r_1}, \frac{\sigma_{r_1}^2}{|T_{r_1}.v|}\right)\right) < E\left(N\left(\mu_{r_2}, \frac{\sigma_{r_2}^2}{|T_{r_2}.v|}\right)\right). \end{cases} \quad (17)$$

若 $E(N(\mu_{r_1}, \sigma_{r_1}^2 / n)) = E(N(\mu_{r_2}, \sigma_{r_2}^2 / n))$, 说明在分布 $N(\mu_{r_1}, \sigma_{r_1}^2 / n)$ 中取值的价值期望等于分布 $N(\mu_{r_2}, \sigma_{r_2}^2 / n)$ 中取值的价值期望, 代表任务 t_p 和任务 t''_p 的预期价值相等; 当 $E(N(\mu_{r_1}, \sigma_{r_1}^2 / n)) > E(N(\mu_{r_2}, \sigma_{r_2}^2 / n))$ 时, 表示 t_p 的价值大于 t''_p 的价值; 当 $E(N(\mu_{r_1}, \sigma_{r_1}^2 / n)) < E(N(\mu_{r_2}, \sigma_{r_2}^2 / n))$ 时, 代表 t_p 的价值小于 t''_p 的价值.

4 基于核密度估计的工人分布预测模型

由于空间众包系统中工人位置信息的时效性,为了实现全局更优的分配,需要对新加入空间众包系统的工人的位置信息进行精确预测.本节将介绍一种基于 ConvLSTM 的工人签到位置预测方法,它可以有效地估计未来时刻工人的数量和位置分布.工人分布预测过程的总体结构图如图 5 所示,从左向右,首先通过二维核密度估计进行过去时间工人签到位置分布平滑化处理,使得 ConvLSTM 模块可以更容易地提取到空间分布的特征.然后,构建基于 ConvLSTM 的 2 层网络模型,在输入层按照定宽的窗口滑动划分各时刻的二维核密度估计矩阵,得到时序的核密度估计矩阵投入模型,经过隐藏层的 ConvLSTM 模块提取空间特征,接着通过输出层预测得到下一时刻的工人分布的二维核密度估计矩阵.最后,通过删减策略的数量还原过程来精确预测未来时刻工人位置的分布.

4.1 二维核密度估计位置分布

本节将任务的位置分布信息使用其二维核密度图进行表示.将真实世界的地图视作一个 $U = [0, 1]^2$ 的二维数据空间,对于每个时间实例的工人签到位置分布样本,使用二维核密度估计可以得到工人分布的概率分布曲线.对数据空间 U 中的概率分布曲线进行模拟,数据空间二维长度均为 0 到 1,如果不对估计函数归一化处理,数据空间中对应位置的函数值就是该位置可能签到的工人数.二维核密度的估计函数为

$$f_w(x) = \prod_{dim=1}^2 1/(H_{dim}) \times K((x[dim] - x_i[dim])/H_{dim}),$$

其中, $K(\cdot)$ 为核函数, dim 表示维度 ($dim=1$ 表示二维数据空间的第一维), 由于后续分配工人时会对核密度估计矩阵进行向下取整的操作, 为避免向下取整丢失较大的小数值(比如 1.9 取整为 1)此处使用均匀核函数 $K(x) = 1/2, -1 \leq x \leq 1, H_{dim}$ 为平滑

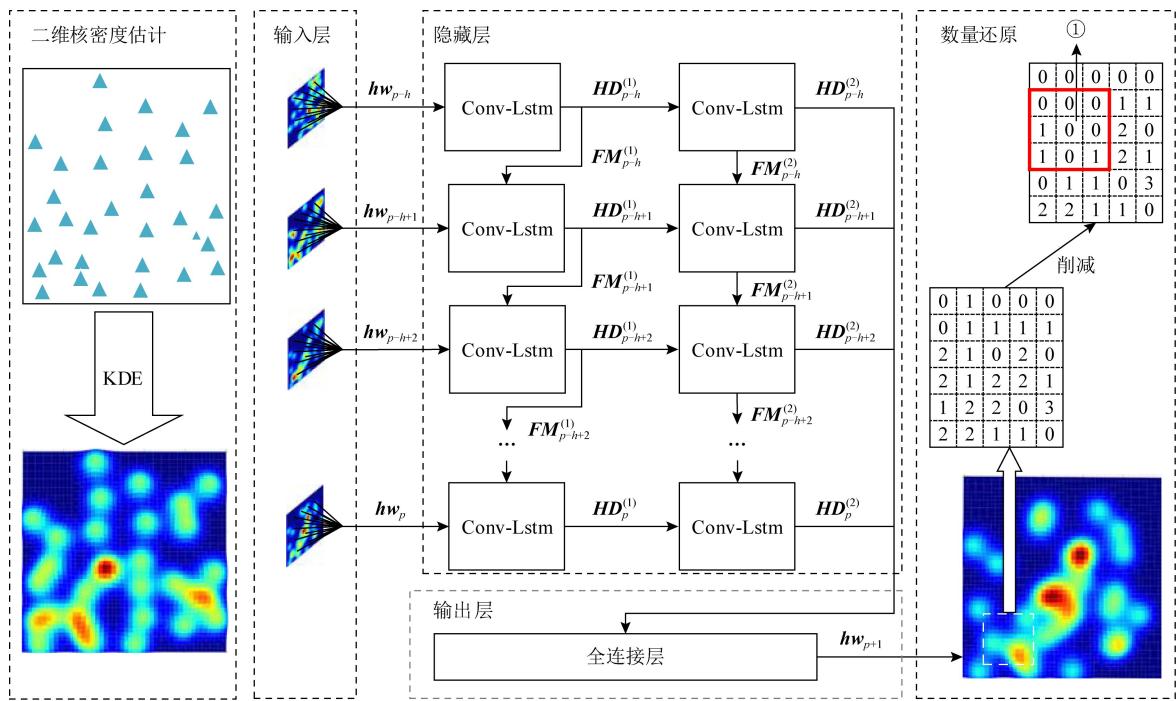


Fig. 5 Working mechanism of worker distribution prediction

图 5 工人分布预测工作原理

参数带宽,计算方法与 3.1 节中计算核密度估计滑窗的方法类似(H_1, H_2 需要分别计算).

通过如图 5 所示方法可以得到工人的分布矩阵 h_w , 该步骤如图 5 中二维核密度估计部分所示, 得到的分布矩阵类似图中的热力图, 核密度估计将稀疏的样本通过带宽 H_{dim} 扩散到相邻区域, 使得工人分布矩阵密度变高, 方便后续卷积过程更好地捕捉工人空间分布的特征. 但也是由于该原因, 工人矩阵中的数量和并不是真正的工人数量.

4.2 基于 ConvLSTM 的工人分布预测模型

本节将通过工人分布预测模型提取工人分布核密度估计矩阵的时空特征来预测下一时刻的工人分布, 并基于核密度估计原理复原真实工人数量, 具体分为如下 2 个步骤:

1) 工人预测模型网络结构. 由于 ConvLSTM 模型^[23]在时空数据特征提取上性能较佳, 本文将其应用于预测众包工人登录位置分布. 在 ConvLSTM 中, 要预测时间实例 $p+1$ 时的工人位置分布, 需要基于设定的滑窗宽度 s 输入一组过去工人分布的核密度估计矩阵的时间序列 $H_w = \{h_w_{p-s}, h_w_{p-s+1}, \dots, h_w_{p-1}, h_w_p\}$, H_w 表示核密度估计矩阵的时间序列, h_w_p 为时刻 p 的核密度估计矩阵. 时间序列 H_w 为图 5 中输入层的输入. 网络结构中各参数的形式化表达如式(18)~(23)所示:

$$I_p = \text{Sigmoid}(\text{Conv}(h_w_p; w_{xi}) + \text{Conv}(\mathbf{HD}_{p-1}; w_{hi}) + b_i), \quad (18)$$

$$F_p = \text{Sigmoid}(\text{Conv}(h_w_p; w_{xf}) + \text{Conv}(\mathbf{HD}_{p-1}; w_{hf}) + b_f), \quad (19)$$

$$O_p = \text{Sigmoid}(\text{Conv}(h_w_p; w_{xo}) + \text{Conv}(\mathbf{HD}_{p-1}; w_{ho}) + b_o), \quad (20)$$

$$NM_p = \text{Tanh}(\text{Conv}(h_w_p; w_{xg}) + \text{Conv}(\mathbf{HD}_{p-1}; w_{hg}) + b_g), \quad (21)$$

$$FM_p = F_p \odot FM_{p-1} + I_p \odot NM_p, \quad (22)$$

$$HD_p = O_p \odot \text{Tanh}(FM_p), \quad (23)$$

其中, \odot 符号表示矩阵的哈达玛乘积, I_p, F_p, O_p 分别表示输入门、遗忘门、输出门 3 个门的程度参数, NM_p 是常规循环神经网络(recurrent neural network, RNN)步骤得到的特征, 为短时记忆, FM 表示长时记忆, $HD^{(1)}/HD^{(2)}$ 分别为输出给下一单元/最后输出的特征. 在隐藏层网络中, 如图 5 隐藏层部分, 各 ConvLSTM 单元的长期记忆 FM 随着输入序列时间方向传播, 各单元的输出 HD 随网络的深度方向传播. 经过 2 层网络提取 H_w 的时空特征, 输出结果经图 5 中输出层的全连接网络后, 可以预测得到 $p+1$ 时刻的工人分布核密度估计矩阵 h_w_{p+1} .

2) 复原真实工人数量. 通过上述步骤预测得的工人分布核密度估计矩阵 h_w_{p+1} 是 $p+1$ 时刻工人分布的核密度估计矩阵, 分配工人前需要先对矩阵

进行取整操作.根据核密度估计的原理,核密度估计矩阵是通过核函数 $K(\cdot)$ 以带宽 H_{dim} 对每一个样本进行观察,并拟合样本点附近的概率,将所有的概率密度函数合并得到的,故每个样本影响概率分布的范围为 $\pm H_{dim}$.所以,核密度估计矩阵中对应位置的值并不是真实工人数量,若要得到 hw_{p+1} 中的真实工人数量,需要在减少矩阵某位置的数量时,同时减少该位置 $\pm H_{dim}$ 范围的值.

复原真实工人数量方法如下:假设核密度估计时使用的 H_{dim} 为矩阵的 len 个宽度,如果分配矩阵中 (i, j) 位置的一个工人,需要将 (i, j) 位置和 $\{i - len, \dots, i, \dots, i + len\} \otimes \{j - len, \dots, j, \dots, j + len\}$ (\otimes 表示笛卡尔积) 中元素位置中工人数大于零的部分全部减少 1.以图 5 中数量复原部分使用虚线框截取矩阵中的一部分为例,箭头连接的矩阵为该部分工人的分布矩阵,矩阵中的数字表示对应位置的工人数量.通过模拟分配 2 个工人的过程来展示消减数量策略的过程,如图 6 所示:

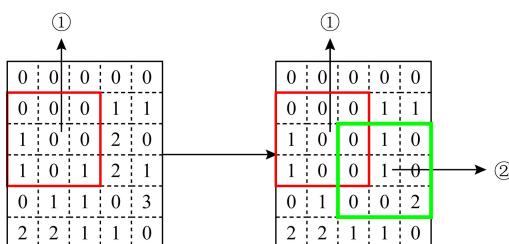


Fig. 6 Example of quantity reduction

图 6 数量削减过程示例

首先,将图 6 中标注①的(3,2)位置处的 1 个工人进行分配,假设进行核密度估计时计算出的 H_{dim} 在核密度估计矩阵中宽度为 1,则以集合 $\{2,3,4\} \otimes \{1,2,3\} = \{(2,1), (2,2), (2,3), (3,1), (3,2), (3,3), (4,1), (4,2), (4,3)\}$ 的九宫格内,数量大于零的位置工人数量均要减少 1;将标注②的(4,4)位置处的 1 个工人进行分配,矩阵中 $\{3,4,5\} \otimes \{3,4,5\} = \{(3,3), (3,4), (3,5), (4,3), (4,4), (4,5), (5,3), (5,4), (5,5)\}$ 位置的网格中人数均减 1.分配可以持续到矩阵所有位置的数量都为 0.

5 基于位置预测的任务分配算法

传统的任务分配问题一般可视为最大流问题,本文提出的最大价值最小成本任务分配问题可以分类为一种包含位置预测数据,任务为结点和工人为权值的二分图最大流问题,并且结点和边均带权值.

解决最大流问题的传统算法有匈牙利算法和 Kuhn-Munkras 算法,若图中结点有 $|V|$ 个,边有 $|E|$ 个,则前者的时间复杂度为 $O(|V| \times |E|)$,但不能解决带权值的二分图最大流问题,后者可以解决带权值的二分图最大流问题,但时间复杂度为 $O(|V|^3)$,难以应用于现实场景.传统的方法没有完全适配本文问题的求解方法.故本节提出一种基于贪心算法和匈牙利算法的启发式算法,通过搜索任务的可用工人集,构造二分图,尽可能寻找符合目标函数式(1)(2)的最优分配.

定义 4. 任务可用工人集.在理想的众包系统中,每个空间任务的有效工人集由任务需要的技能和工人拥有的技能、工人的可信度等几个因素决定,但现有的数据集中缺少这几种属性,故本实验中仅考虑工人是否可以在任务结束前到达任务位置(不考虑工人执行任务的时间).在 p 时,若已存在系统中和预测将要加入系统的空间任务和众包工人分别有 M 和 N 个,空间任务 t 的可用工人集表示为 AW_t ,且应该遵循以下条件:

- $\forall w \in AW_t$,
- 1) $dist(t, w) \leq (t.e - p) \times w.v$;
 - 2) $\sum_{i=1}^M ind(w, t_i) \leq 1$.

其中, $ind(w, t)$ 表示一个指示器,若工人 w 被分配给任务 t ,则 $ind(w, t) = 1$,否则 $ind(w, t) = 0$; $dist(t_i, w_j)$ 为任务位置 $t_i.l$ 到工人位置 $w_j.l$ 之间的欧氏距离,若任务 t_i 为预测得到的任务,则任务位置 $t_i.l$ 为式(4)计算得到的坐标 $(x_p, y_p)_i$,若工人 w_j 为预测得到的任务,则任务 j 的位置 $w_j.l$ 为任务 j 所在核密度矩阵位置对应网格处的中心坐标.

5.1 基于预测数据的任务分配算法

假设时间序列 P 中系统中一共存在任务及工人分别有 M 和 N 个,算法目的为在多项式时间复杂度内将 N 个工人分配给 M 个任务,并使分配对间的平均移动距离尽可能低,分配对的总价值尽可能高.

算法 1 给出基于预测数据的任务分配算法的伪代码.

算法 1. 基于预测数据的任务分配算法.

输入:时间序列 $P = \{1, 2, \dots, p, \dots\}$ 、时刻 p 任务集 T_p 、时刻 p 工人集 W_p ;

输出:任务分配集 I .

$$\textcircled{1} \quad I = \emptyset, T_n = \emptyset, W_n = \emptyset, I'_1 \leftarrow \emptyset;$$

```

② for  $p=1$  to  $n$  then
③    $T_p \leftarrow T_p + T_n, W_p \leftarrow W_p + W_n, T_n \leftarrow \emptyset,$ 
       $W_n \leftarrow \emptyset;$ 
④    $I_p \leftarrow \emptyset, W_u \leftarrow \emptyset, T_c \leftarrow \emptyset, C \leftarrow \emptyset;$ 
⑤   for each pairs  $\langle t, w \rangle_p \in I'_p$ 
⑥     if 分配对  $\langle t, w \rangle_p$  在时刻  $p$  存在 then
⑦        $I_p \leftarrow I_p + \langle t, w \rangle_p;$ 
⑧     else 将已有的对象加入  $T_p$  或  $W_p$  ;
⑨     end if
⑩   end for
⑪   for each  $t_i \in T_p$  do
⑫     使用定义 2 生成  $AW_{t_i}$  ;
⑬     使用  $AW_{t_i}$  生成二分图矩阵  $C$  ;
⑭   end for
⑮   基于式(14)(15)(17)将  $C$  降序排序,
       $i \leftarrow -1$  ;
⑯   while  $T_p \neq \emptyset$  and  $W_p \neq \emptyset$  do
⑰      $i \leftarrow i + 1$  ;
⑯     将  $C[i]$  升序排序;
⑯     for  $j = 0$  to  $\text{len}(C[i])$  do
⑯       if  $w_j \notin W_u$  then
⑯          $I_p \leftarrow I_p + \langle t_i, w_j \rangle, W_u \leftarrow W_u - w_j$  ;
⑯          $T_p \leftarrow T_p - t_i, W_p \leftarrow W_p - w_j$  ;
⑯         break;
⑯       end if
⑯       if  $w_j \in W_u$  then
⑯         continue;
⑯       end if
⑯       if 任务  $t_i$  没能被分配 then
⑯         将  $T_c$  按价值升序排序;
⑯         for each  $t'_i \in T_c$  do
⑯           if  $w \notin W_u \& \& w \in AW_{t'_i}$  then
⑯             选择出距离最小的  $w'_j$ , 使用
⑯              $\langle t'_i, w'_j \rangle$  替换  $I_p$  中的  $\langle t'_i, w''_j \rangle$  ;
⑯              $I_p \leftarrow I_p + \langle t_i, w'_j \rangle$  ;
⑯              $T_p \leftarrow T_p - t_i$  ,
⑯              $W_p \leftarrow W_p - w'_j + w''_j$  ;
⑯             break;
⑯           end if
⑯         end for
⑯       end if
⑯       if 任务  $t_i$  无法被分配 then
⑯          $T_n \leftarrow T_n + t_i$  ;
⑯          $T_p \leftarrow T_p - t_i$  ;

```

```

⑪       end if
⑫       end if
⑬     end while
⑭      $W_n \leftarrow W_p$  ;
⑮      $I_p \leftarrow I_p - \{\langle t, w \rangle_p\}$  ;
⑯      $I'_{p+1} \leftarrow \{\langle t, w \rangle_p\}$  ;
⑰      $I = I + I_p$  ;
⑱   end for
⑲   return  $I$ .

```

算法 1 的输入为一个时间实例 P , 及当前时刻 p 的任务集 T_p 和工人集 W_p . 首先, 初始化一个空的总分配集合 I , 上时刻未分配的任务集和工人集 T_n 和 W_n , 时刻 1 预测任务分配集合 I'_1 (行①). 开始按时间顺序遍历时间序列 P , 每时刻初始化一个空的时刻 p 工人/任务分配集合 I_p , 一个空的已分配工人集 W_u , 一个冲突任务集 T_c , 并把上一时刻没被分配的工人/任务 T_n 和 W_n 加入 T_p/W_p , 继续参与分配(行②~④). 遍历上一时刻预测任务分配集 I'_p , 判断含预测对象的分配对中预测工人/任务是否真的出现(预测位置是否有工人/任务), 如果是, 将该分配对加入当前时刻的分配实例 I_p , 如果没有, 则将分配对中已存在的任务/工人加入当前时刻的 T_p 和 W_p 集中, 继续参与分配(行⑤~⑩). 然后, 遍历任务集 T_p , 根据定义 2 计算每一个任务的可用工人集 AW_{t_i} , 并根据 AW_{t_i} 创建一个代表二分图的矩阵 C , 该矩阵维数为 2, 第 1 维长度为 $|T_p| = M$, 第 2 维长度为 $|W_p| = N$, 遍历任务集 T , 对于任务 t_i , 若可用工人集 AW_{t_i} 之中存在工人 w_j 则 $C[i][j] = dist(t_i, w_j)$, 即代表任务 t_i 和工人 w_j 之间存在边, 且边的权值为任务 t_i 和工人 w_j 之间的欧氏距离 $dist(t_i, w_j)$ (行⑪~⑭).

再者, 遍历任务集开始分配. 按照 3.1 节预估任务价值式(14)(15)(17)比较任务间价值的高低, 按照任务价值由高到低的顺序将矩阵 C 的第 1 维排序(行⑮). 遍历矩阵 C , 对于遍历到的任务 t_i , 将 $C[i]$ 按第 2 维数据从小到大进行排序, 遍历排序后工人集合选择工人 w_j 与任务 t_i 配对; 若该工人不在 W_u 中, 即代表工人没有被分配, 就将分配对 $\langle t_i, w_j \rangle$ 加入分配实例集合 I_p , 工人 w_j 加入 W_u 中, 将 t_i, w_j 从 T_p 和 W_p 中删除, 然后继续分配下一个任务(行⑯~⑯). 如果选择的工人存在于 W_u 中, 代表工人 w_j 已经被分配, 则继续遍历下一个工人(行⑯~⑯). 如果该任务的可用工人集中所有工人均已被分配, 则遍历 I_p 的可用工人集, 将这些工人分配

的任务视作冲突任务集 T_c ;按分配对间距离从小到大的顺序遍历 T_c ,访问分配对中的冲突任务 t'_i (行 28~30);遍历 t'_i 的可用工人集 $AW_{t'_i}$,如果 t'_i 有其他不存在于 W_u 中的工人,则选择距离最小的工人 w'_j ,将 I_p 中已有分配对 $\langle t'_i, w'_j \rangle$ 替换为 $\langle t'_i, w_j \rangle$ (行 31~32),并把 $\langle t_i, w_j \rangle$ 加入 I_p ,接着把任务 t_i 和 w'_j 移出时刻 p 的任务/工人集 T_p/W_p ,把 w'_j 重新加入 W_p (行 33~34),并跳出循环(行 35).如果任务 t_i 的 T_c 集中所有的任务均没有可以替换的工人,则代表 t_i 在当前时刻就不能被分配,在这种情况下,若任务还有存续时间,则将任务 t_i 从 T_p 中移除并加入 T_n ,参加下一时刻的分配(行 38~41).在这时刻的分配结束后(T_p 或 W_p 被清空),若还有工人没有被分配,则将剩余的工人加入 W_n 参加下时刻的分配;将 I_p 中含有预测任务/工人的分配对加入 I'_{p+1} 中,并从 I_p 中删除,在下一时刻判断预测的对象是否发生(行 44~46).将时刻 p 的分配实例集 I_p 加入总分配实例集 I (行 47).遍历时间序列重复上述操作,直到时间序列结束,算法返回总的分配实例集 I (行 48).

5.2 算法复杂性分析

本文算法中生成所有 AW_i 的复杂度约为 $O(N \times M)$,分配步骤的复杂度约为 $O(M \times \log M + M \times N \times \log N)$,故从整体的角度分析其平均时间复杂度约为 $O(N \times M \times \log M)$,此处 N 表示工人的数量, M 为任务的数量.

例 1. 如图 7 所示,共有 4 个任务($t_1 \sim t_4$)和 5 个工人($w_1 \sim w_5$),任务的价值和工人到各任务的距离如表 1 和表 2 所示.首先分配价值较高的空间任务 t_2 ,在 t_2 的可用工人集中,距离最近的是工人 w_3 ,故首先将 w_3 分配给 t_2 .然后分配价值为 4 的空间任务 t_1 ,在 t_1 的可用工人集中,距离最近的是工人 w_1 ,并且 w_1 没有分配给别的任务,故把 w_1 分配

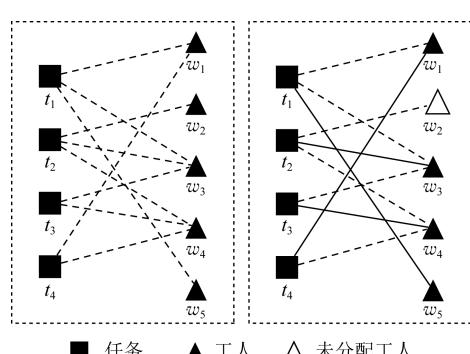


Fig. 7 Example of task assignment

图 7 任务分配示例

给任务 t_1 .接着分配任务 t_3, t_4 可用的工人中距离最近的工人是 w_4 ,故将 w_4 分配给任务 t_3 .最后分配 t_4, t_4 的可用工人 w_1 和 w_4 均已经被分配,故可以得出冲突任务集 T_c 为 $\{\langle t_1, w_1 \rangle, \langle t_3, w_3 \rangle\}$,两冲突分配对的距离分别为 2 和 4,首先尝试替换 t_3 的分配,但 t_3 没有其他可用工人,故选择替换 $\langle t_1, w_1 \rangle$ 为 $\langle t_1, w_5 \rangle$.最终得到的分配 $I_p = \{\langle t_1, w_5 \rangle, \langle t_2, w_3 \rangle, \langle t_3, w_4 \rangle, \langle t_4, w_1 \rangle\}$,而剩余无法被分配的 w_2 会留到下一时刻继续参与分配.经过本文方法分配后,得到的总价值为 14,平均距离为 2.75;而从价值高的任务开始遍历,每次选取距离最近工人的贪心法取得的总价值 12,平均距离为 3, $14 > 12$ 且 $2.75 < 3$,本文的方法明显更优.

Table 1 Task Value

表 1 任务价值

任务	价值
t_1	4
t_2	5
t_3	3
t_4	2

Table 2 Distance of Task and Worker

表 2 任务工人间距离

任务	w_1	w_2	w_3	w_4	w_5
t_1	2		4		3
t_2		7	3	5	
t_3			4	3	
t_4		2			4

6 实验

6.1 实验设置

本文实验部分使用滴滴盖亚开放数据计划提供的真实数据集,具体在滴滴提供的 KDD CUP 2020 网约车数据集^[2]上测试提出的算法.该数据集中包含 30 天的用户订单数据和司机轨迹数据,共有 7 065 937 个订单数据,来自 1181 180 个司机的 6 105 003 条轨迹数据,数据均来自滴滴公司的网约车服务.本实验以该数据集范围内的区域(经度 $104.042102^\circ \sim 104.129591^\circ$ 和纬度 $30.652828^\circ \sim 30.727818^\circ$ 范围内的区域)作为实验的背景地区.实验将订单数据和司机轨迹数据分别视作最大价值最小成本任务分配问题中的空间任务和众包工人,具体地说,订单的发布时间视作空间任务的发布时间,订单的发布位置

视作空间任务的发布位置,订单价值视作空间任务的价值;将每条轨迹的首个时间戳和首个轨迹点坐标视作一个众包工人的登录时间戳和登录位置。经过处理得到的数据集包含 7 065 937 个众包任务和 6 105 003 个众包工人。

本实验在该数据集上通过 3.1 节的轨迹聚类得到 315 个拐角和 384 条轨迹,经过图构建过程建立出一个 384 个结点,806 条边的图(根据各路段间的连接情况得到)。为验证所提方法的性能,对最大价值最小成本任务分配问题中需要但数据集里不存在的属性进行合成,本实验使用高斯分布来模拟每个任务的持续时间和工人的速度,此处任务持续时间指的是任务发布后的有效时间,也就是 $t.e - p$,工人速度指的是工人的移动速度,也就是 $w.v$,如表 3 所示,任务持续时间和工人移动速度在 $N((p^- + p^+)/2, (p^+ - p^-)^2)$ 和 $N((v^- + v^+)/2, (v^+ - v^-)^2)$ 中随机取值。实验中选择前 20 天的数据作为训练集,第 21 天的数据作为验证集,22~30 天的数据作为测试集。

Table 3 Experimental Parameters

表 3 实验参数

属性设置	值
滑窗宽度 s	4,6,8,10,12
训练集大小	30%,40%,60%,80%, 100%
测试集大小/ d	1,3,5,7, 9
任务持续时间 $[p^-, p^+]/\text{min}$	30~45,45~60, 60~75 , 75~90,90~105
工人速度 $[v^-, v^+]/(\text{km} \cdot \text{h}^{-1})$	5~6,6~7, 7~8 ,8~9,9~10

注:黑体数字为默认值。

6.2 评价标准及对比算法

本实验根据预测结果的准确率以及误差来评估本文预测方法的有效性;以式(1)中定义的最大化总分配价值和式(2)最小化分配中工人到空间任务的平均距离目标,衡量分配策略的质量,并以最小化计算所消耗的 CPU 时间为目标,来衡量分配策略的效率。表 3 给出本文的实验参数设置,其中默认值以黑体显示。在实验过程中,每组实验都会改变一个参数,其他参数均会设置为默认值。

在本文的实验中,任务预测方法的有效性验证采取 2 种对比方法:1)单层图卷积网络(GCN- α)。该方法使用本文 3.2 节中去掉隐藏层 2 的任务位置预测模型进行预测;2)基于网格的任务预测方法(grid-based task prediction, GTP)^[17]。使用基于网格划分的方法划分数据空间,并在各单元格内使用线性回

归的方法进行任务数量预测,并假设任务发生在单元格的中心位置。工人位置预测方法的有效性验证采取 2 种对比方法:1)不使用核密度估计处理工人分布图的 ConvLSTM 模型(ConvLSTM- α)。该方法为本文的基于 ConvLSTM 的工人分布预测模型,去掉 4.1 节和 4.2 节(2)中二维核密度估计预处理和还原步骤的工人位置预测模型;2)基于网格的任务预测法(grid-based worker prediction, GWP)^[17]。与 GWP 相似,网格划分数据空间,使用线性回归法进行工人数量预测。

分配方法的有效性和效率采用了 2 种基础的对比算法:1)贪心分配算法(greedy)。以任务价值从高到低的顺序选择任务,每个任务均在其可用工人集中选择距离最短的工人进行分配,若没有可用工人集,则该任务当前时间实例无法分配。与本文基于预测数据的任务分配算法相比较,贪心分配算法把其中分配部分替换为贪心算法,平均时间复杂度约为 $O(2M \times N + M \times \log M)$ 。2)随机分配算法(random)。随机选择一个任务,在其可用工人集中随机选择工人进行分配,若没有可用工人集,则该任务当前时间实例无法分配,该分配算法的平均复杂度约为 $O(M \times N + M)$,其中 M 为任务数量, N 为工人数量。

本文所有实验均是在 AMD Ryzen 5 3600 CPU @3.6 GHz,16 GB 内存以及 Nvidia GTX 1660s 硬件条件下使用 Python 语言实现实验中算法。

6.3 工人和任务预测效果分析

本节实验对工人预测效果表现进行评估,并评估其对后续任务分配步骤的影响。使用前 20 天的数据作为训练集,后 9 天为测试集,并通过改变滑窗宽度和训练集大小,来测试模型准确度的变化,同时设定模型的其他参数如表 3 中默认值(加黑)所示。

1) 滑窗宽度的影响。

首先通过改变滑窗宽度来评估工人和任务分布预测的准确性,实验的 epoch 设置均为 100,分别使用平均绝对误差(mean absolute error, MAE)和均方根误差(root mean square error, RSME)2 种误差来衡量模型的准确度情况,滑动窗口带宽的取值为 4,6,8,10,12。如图 8(a)~(b)所示,对于任务分布预测模型,可以观察到随着滑窗宽度从 4 增大到 8 的过程中,预测的误差略有下降,但从 8 增大到 12 的这个过程中,预测的误差基本上停止变化。这是因为在 3.2 节对任务预测的模型设定中,滑窗宽度代表投入模型的历史数据量大小。3.1 节中基于时间划

分不同模型一定程度上消除任务位置分布随时间变化的影响,对于任务位置预测模型来说不需要太多的历史数据进行学习.图 8(c)~(d)中工人分布预测中滑窗宽度从 8 增大到 12 的过程中,误差反而有所增加,这是因为工人的分布随时间变化较为迅速,

使用较多的历史数据反而会导致模型不能专注于近期的变化,产生较大的误差.总的来说,2 个模型对于带宽的敏感程度都不大,且预测效果均较好(MAE 和 RMSE 均小于 1),根据实验结果选择默认的滑动窗口大小为 8.

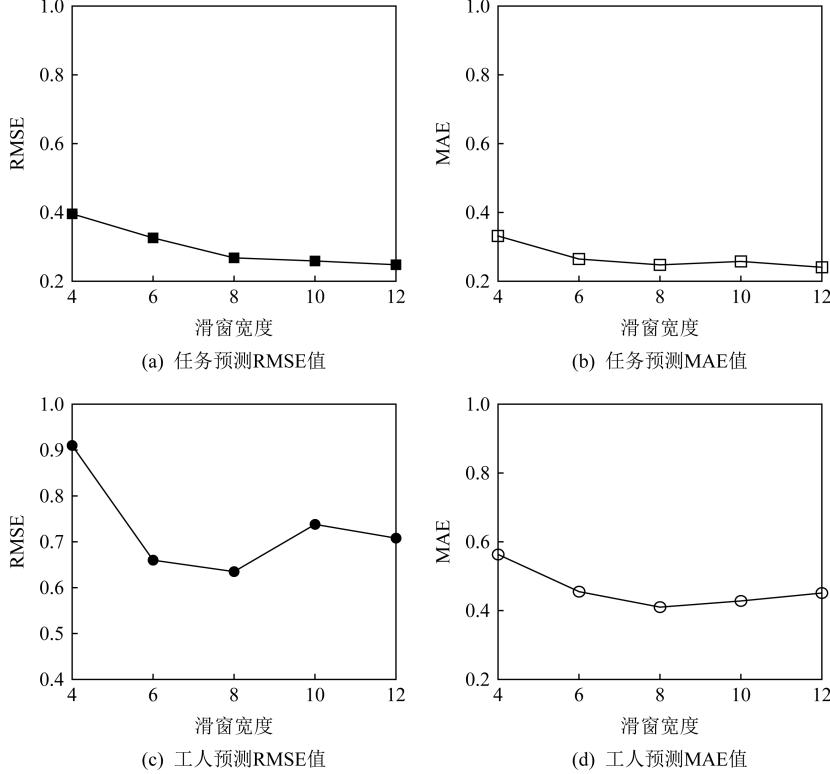


Fig. 8 Task/worker prediction model effect

图 8 任务/工人预测模型效果

2) 训练集大小的影响.

① 任务分布预测分析.通过改变训练集大小来评估工人位置预测算法的性能和该部分对后续任务分配的影响.数据集的前 20 天设定为训练集,训练集的大小取值为 30%, 40%, 60%, 80%, 100%.对于任务位置分布预测,使用了 2 种对比方法:基于网格的任务预测方法(GTP),单层图卷积网络(GCN- α).为衡量模型性能,使用一种基于网格的准确率:将二维数据空间分为等大小的网格,通过比较单元中工作者/任务的估计数量 q' 和实际数 q 得出准确率 $ACC = |q' - q| / q \times 100\%$.为了测试位置预测效果对分配结果的影响,使用本文所提基于位置预测的任务分配算法,基于位置预测进行了任务分配,并计算了分配的总价值.

在准确率评估实验中,在图 9(a)中可以观察到,随着训练集变大,3 种方法均有一定程度的准确率

提升.本文方法的准确率从 64.4% 提升到了 89.1%,比排名第 2 的 GCN- α 最终的 82.1% 高出了 7 个百分点,而 GTP 方法的准确率最后稳定在 73.5% 左右.在图 9(b)中可以发现分配得到的价值随着 3 种方法准确率的增加均有明显上升.GCN- α 和 GTP 方法在训练集从 40% 增加到 60% 的步骤中有较大的价值提升,这是因为训练集大小在 30%~40% 左右时,GCN- α 和 GTP 的准确率仅有 50% 左右,导致产生了大量无效分配,在准确率提高后分配效果才有所提升.而本文方法的准确稳定在 64% 以上,故总的分配价值比较稳定.本文方法在 3 种方法中表现最好,原因在于本文方法使用轨迹路段作为预测任务分布的基础,相较 GTP 和 GCN- α 预测方法更加精细,更加符合工人接受任务的位置.

② 工人分布预测分析.对于工人的位置分布预测,对比本文方法选择了 2 种对比方法:基于网格的

工人预测方法(GWP),不进行二维核密度估计的本文方法(ConvLSTM- α)。为了衡量模型性能,同样使用 $ACC = |q' - q|/q \times 100\%$ 做为准确率评价标准,与任务预测算法效果实验中相似,使用基于位置预测的任务分配算法进行任务分配,并比较分配的总价值。

准确率评估实验如图 10(a)所示,随着训练集增大,3 种方法工人预测的准确率均有增加。本文方法准确率最高,其次是 GWP 和 ConvLSTM- α 。原因在于本文方法通过核密度估计把数据空间划分为比 GWP 方法更精细的子区域,使用核密度估计弥补了工人分布矩阵的稀疏性,相较 ConvLSTM- α 方

法中直接使用 ConvLSTM 提取时空特征的做法,可以更好的提取工人分布的空间特征。本文方法工人预测准确率最高可达 73.6%,相比 GWP 方法高出 18.8 个百分点。ConvLSTM- α 方法的准确率最终仅达到 52.3%,这是由于没有经过预处理的工人分布图过于稀疏,模型不能提取到空间特征,导致预测结果不理想。图 10(b)中,在训练集大小从 30% 增加到 40% 的过程中可以观察到基于 ConvLSTM- α 方法预测的分配结果反而有所下降,这是由于任务分配很大程度上依赖位置预测的结果,而 ConvLSTM- α 预测结果平均准确度仅有 21%,较差的预测准确率会使分配得到的价值不稳定。

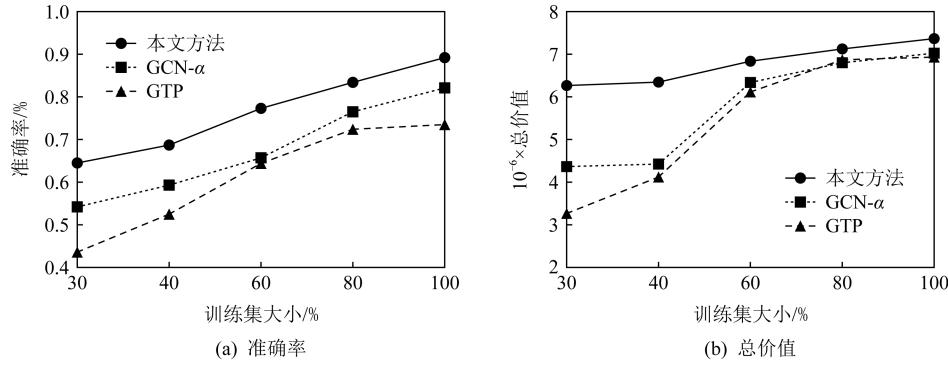


Fig. 9 Performance comparison of task prediction under different training sets

图 9 不同训练集下任务预测性能对比

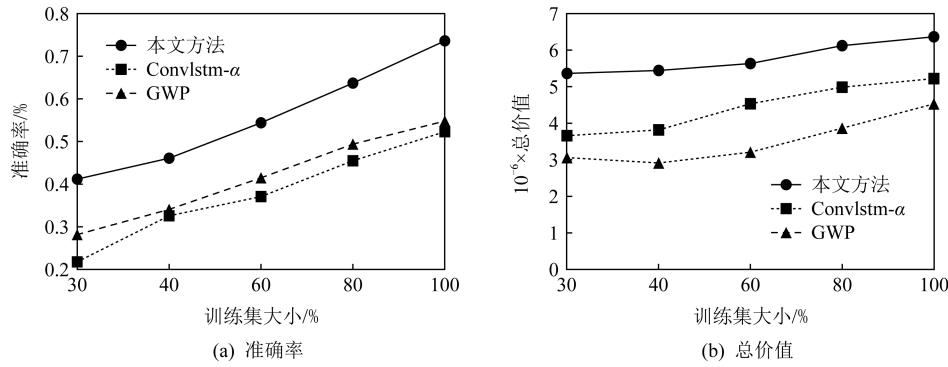


Fig. 10 Performance comparison of worker prediction under different training sets

图 10 不同训练集下工人预测性能对比

6.4 任务分配效果分析

本节实验评估本文提出的基于位置预测的任务分配算法的有效性,本节使用任务分配的总价值和分配对任务/工人间的平均距离评价分配策略的效果,使用计算分配所需的 CPU 时间评价分配策略的效率。对比算法选择贪心分配算法(Greedy)和随机分配算法(Random)。除了 2 种基础的对比算法,

还实现了 3 种朴素对比算法,即 3 种预测算法不使用位置预测数据,在仅考虑当前时刻任务/工人分布的情况下进行分配。对比算法分别为:基于位置预测的任务分配算法(本文方法)、贪心分配算法(Greedy)、随机分配算法(Random),和不加位置预测的任务分配算法(本文方法- α)、不加预测的贪心分配算法(Greedy- α)、不加预测的随机分配算法(Random- α)。

1) 测试集大小的影响.实验选择 22~30 天的数据作为测试集,训练集大小取值分别为 1 天,3 天,5 天,7 天,9 天.首先改变测试集的大小来研究分配算法的性能和时间效率,也即通过改变参加分配的任务和工人数量来测试算法.价值评估实验如图 11(a)所示,所有算法分配得的总价值随着测试集大小的增加而增长;同时可以观察到 Random 算法和 Random- α 算法计算得到的分配总价值最低,而本文提出任务分配算法得到的分配总价值最高,随后是本文方法- α 、Greedy 和 Greedy- α ,这是由于本文方法在有任务无法被分配时,会考虑替换已有的分配实现更好的分配组合.总体上讲,所有使用了位置预测算法效果均要比没有使用预测的算法好,这是由于基于位置预测的分配算法从全局的角度进行了分配.而 Random- α 在一些情况下优于 Random 算法是由于随机分配算法有较强的随机性,实验结果均符合预期.图 11(b)所示为各算法时间对比,该部分统计的 CPU 时间是分配所消耗的总时间.随着训练集的增大,各算法计算所用时间均呈线性增长,因为 3 种算法的时间复杂度均与任务/工人数量相关,任务/工人数量的增加导致耗时增加.可以观察到所有使用位置预测的算法时间消耗均高于不使用预测的算法,这是因为预测数据本身需要消耗时间,且预测得到的任务/工人也参与分配增加了 3 种算法要分配的任务数量.本文方法的耗时最高,这是寻找尽可能多的分配对需要消耗时间.图 11(c)为各分配对间的任务和工人间的平均距离变化情况.随着训练集大小的增长,Greedy 算法和本文提出的任务分配算法的变化较为稳定,基于位置预测的分配平均距离与不考虑位置预测的平均距离接近,同样稳定.

但 Random 算法的平均距离并不稳定,这是因为随机分配有较强的随机性,这种分配结果显然效果较差,真实的空间众包平台中工人的移动距离必然是越短越稳定越好.Greedy 算法和本文方法的平均距离接近,但 Greedy 算法的效果略优,这是因为 Greedy 算法为每个任务都分配了最近的工人,而本文方法为了分配更多的任务,选择将一些工人分配给了较远的任务,这也是本文方法总分配价值远高于 Greedy 算法的原因.

2) 任务持续时间的影响.接下来通过改变任务持续时间测试分配方法的效果和效率,任务持续时间为高斯分布 $[p^-, p^+]$ 区间内随机取值.如图 12(a)所示,随着任务持续时间的增长,所有方法的分配总价值均有增长,这是由于任务有更长的持续时间意味着原本无法被分配任务可以在更多的时间实例参与分配,使该任务更可能得到分配.本文分配方法同样取得了最高的总价值,所有含位置预测的分配方法均要优于不含位置预测的分配算法,可以证明本文方法的可行性,和预测方法的有效性.图 12(b)中统计的耗时为进行一个时刻任务分配消耗的 CPU 时间.可以观察到,随着任务持续时间的增加,所有方法的计算分配所需的时间都有所增长,这是因为任务的持续时间增长,导致任务的可用工人集变大,而分配算法的时间复杂度与可用工人集大小也有关,与预期结果相符.在最极端的情况下本文方法平均耗时也仅高于 Random 算法 4.20 s,属于可以接受的范围.从图 12(c)中可以观察到除 Random 算法外各算法的平均距离均较为稳定,同样证明了本文方法的可行性.

3) 工人移动速度的影响.实验通过改变工人移动

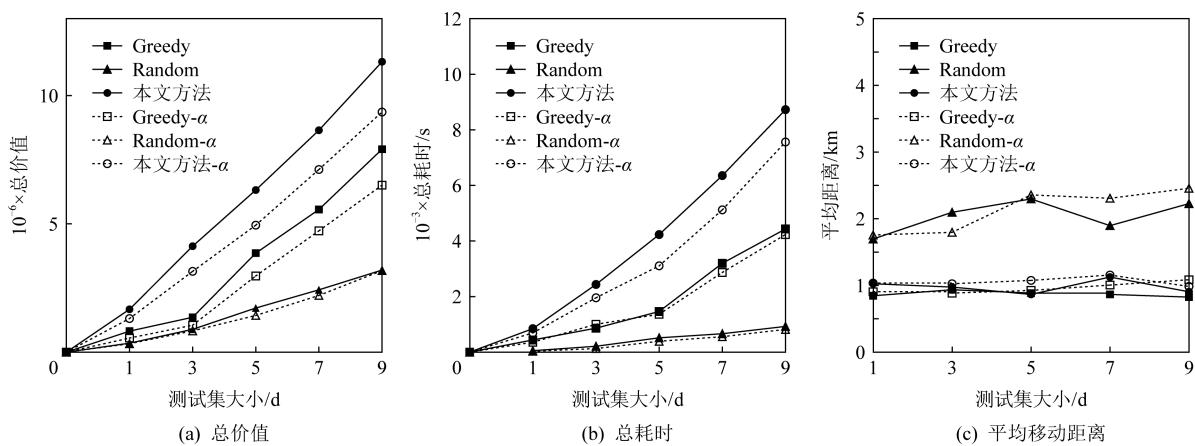


Fig. 11 Comparison of task assignment performance under different datasets

图 11 不同测试集下任务分配性能对比

速度测试分配方法的效果和效率,工人移动速度为高斯分布 $[v^-, v^+]$ 区间内随机取值.图13(a)中,随着工人移动速度增加,所有方法得到的分配总价值均有增长,原因与任务持续时间增长带来的影响类似.工人移动速度增加相等于增加了每个任务的可用工人集,而所有的分配算法均是在可用工人集上进行的,更多的可用工人可能让原本无法被分配的任

务更可能得到分配,同样这增加了需要计算的任务/工人数量,导致了图13(b)中Random算法外所有算法的耗时均有明显增加.图13(c)所示本文分配算法计算得的工人平均移动距离依旧稳定.本文方法仅比Random算法平均多耗时仅为4.89 s的情况下,取得了优于Greedy算法32.7%的分配价值和稳定的工人移动距离,可以证明本文方法的有效性.

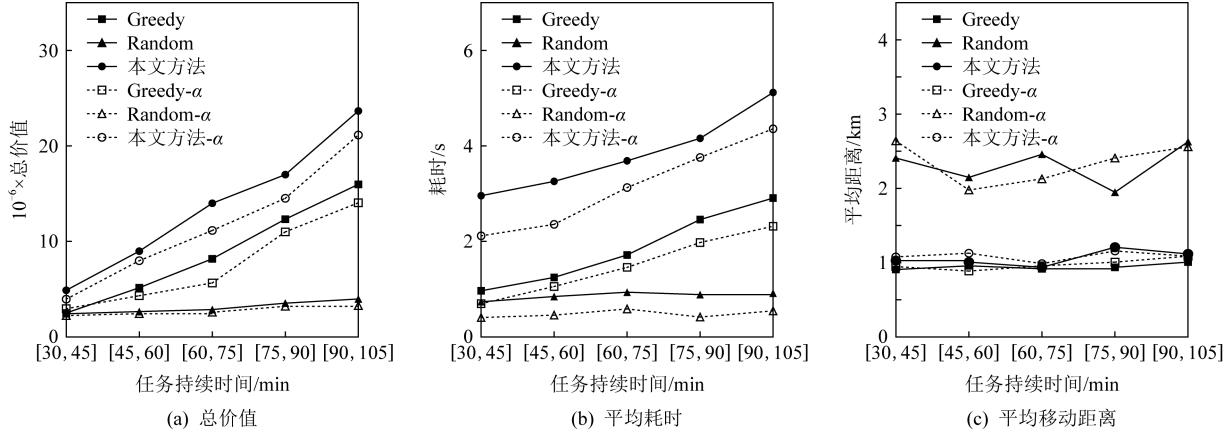


Fig. 12 Comparison of task assignment performance under different time of duration

图12 不同持续时间下任务分配性能对比

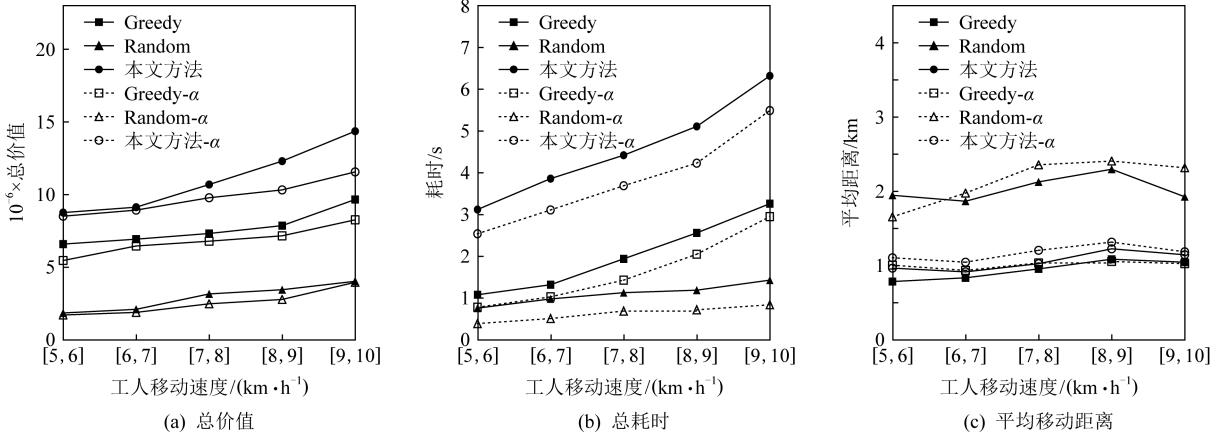


Fig. 13 Comparison of task assignment performance under different movement speed of workers

图13 不同工人移动速度下任务分配性能对比

7 结束语

本文提出一种解决空间众包任务分配的方法,使用基于轨迹的任务位置预测模型和基于核密度估计的工人位置预测模型预测了未来时刻任务/工人的分布,设计了一种基于位置预测的任务分配方法来寻找尽可能好的分配.本文通过大量实验证实

了所提方法的效率和有效性,任务预测和工人预测的准确率优于其他算法,且分配方法能取得远优于基础算法的分配效果.本文通过大量实验证实了所提方法的效率和有效性,任务预测和工人预测的准确率优于其他算法,且分配方法能取得远优于基础算法的分配效果.但在工人预测和分配工作中,没能计算出预测工人的出现概率,只是在预测工人没有出现的情况下尽可能早地把配对的任务重新分配,

导致工人预测不精准时分配效果变差。在未来进一步研究中会针对该问题寻找更优的解决办法,研究更复杂情景中的任务分配方法,实现空间众包中复杂任务(多个工人分配一个任务),及特定路线任务(根据工人移动路线分配多个任务)的分配方法。此外,将应用文献[24-27]中位置预测算法提高众包任务分配的准确性。

致谢:本文实验部分使用的数据来自滴滴出行“盖亚”数据开放计划(Didi Chuxing GAIA Initiative)。

作者贡献声明:徐天承参与研究和文章撰写,包括算法研究、设计论文框架、起草文章和实验等工作;乔少杰参与研究和文章撰写,包括算法提出、设计研究方案、修订论文和指导性支持等工作;武俊参与研究,包括采集整理数据统计分析和实施研究过程等工作;韩楠参与研究,包括采集整理数据、算法设计等工作;岳昆参与工作支持,包括理论基础和数据支持;易玉根参与研究,包括数据采集与处理;黄发良参与工作支持,包括数据和实验支持;元昌安参与研究,包括算法设计及修订研究方案。

参 考 文 献

- [1] Feng Jianhong, Li Guoliang, Feng Jianhua. A Survey on Crowdsourcing[J]. Chinese Journal of Computers, 2015, 38(9): 1713-1726 (in chinese)
(冯剑红, 李国良, 冯建华. 众包技术研究综述[J]. 计算机学报, 2015, 38(9): 1713-1726)
- [2] Didi ChuXing. Didi Chuxing GAIA Initiative: GAIA Open Dataset [EB/OL]. Beijing: Didi ChuXingTechnology Co, Ltd., 2017 [2021-00-00]. <https://gaia.didichuxing.com> (in Chinese)
(滴滴出行. 滴滴出行盖亚数据开放计划: 盖亚开放数据集 [EB/OL]. 北京: 滴滴出行科技有限公司, 2017 [2021-00-00]. <https://gaia.didichuxing.com>)
- [3] Tong Yongxin, Yuan Ye, Cheng Yurong, et al. Survey on spatiotemporal crowdsourced data management techniques[J]. Journal of Software, 2017, 28(1): 35-58 (in Chinese)
(童咏昕, 袁野, 成雨蓉, 等. 时空众包数据管理技术研究综述 [J]. 软件学报, 2017, 28(1): 35-58)
- [4] Gummidi S R B, Xie Xike, Pedersen T B. A Survey of Spatial Crowdsourcing[J]. ACM Transactions on Database Systems, 2019, 44(2): 1-46
- [5] Wang Ning, Wu Jie. Cost-efficient heterogeneous worker recruitment under coverage requirement in spatial crowdsourcing [J]. IEEE Transactions on Big Data, 2021, 7(2): 407-420
- [6] Tong Yongxin, Zhou Zimu, Zeng Yuanxiang, et al. Spatial crowdsourcing: A survey[J]. The VLDB Journal, 2020, 29(1): 217-250
- [7] Kazemi L, Shahabi C. Geocrowd: Enabling query answering with Sspatial crowdsourcing[C] //Proc of the 20th Int Conf on Advances in Geographic Information Systems. New York: ACM, 2012: 189-198
- [8] Mitsopoulou E, Litou I, Kalogeraki V. Multi-objective online task allocation in spatial crowdsourcing systems[C] //Proc of the 40th IEEE Int Conf on Distributed Computing Systems (ICDCS). Piscataway, NJ: IEEE, 2020: 1123-1133
- [9] Alababdi A A, Abulkhair M F. Multi-objective task scheduling optimization in spatial crowdsourcing [J]. Algorithms, 2021, 14(3): 77-97
- [10] Jiang Yun, Cui Lizhen, Cao Yiming, et al. Spatial crowdsourcing task assignment based on the quality of workers[C] //Proc of the 3rd Int Conf on Crowd Science and Engineering. New York, NY: ACM, 2018: 1-6
- [11] Zhao Yongjian, Han Qi. Offline worker selection for real-time spatial crowdsourcing multi-worker tasks[C] //Proc of the 20th IEEE Int Conf on Mobile Data Management (MDM). Piscataway, NJ: IEEE, 2019: 545-550
- [12] Sun Dezhi, Xu Ke, Cheng Hao, et al. Online delivery route recommendation in spatial crowdsourcing [J]. World Wide Web, 2019, 22(5): 2083-2104
- [13] Yu Dunhui, Zhang Xiaoxiao, Zhang Xingsheng, et al. Multitask assignment algorithm based on decision tree in spatial crowdsourcing environment[C] //Proc of Int Conf on Algorithms and Architectures for Parallel Processing. Berlin: Springer, 2019: 300-314
- [14] Tang Feilong, Zhang Heteng. Spatial task assignment based on information gain in crowdsourcing[J]. IEEE Transactions on Network Science and Engineering, 2019, 7(1): 139-152
- [15] Zheng Bolong, Huang Chenze, Jensen C S, et al. Online trichromatic pickup and delivery scheduling in spatial crowdsourcing[C] //Proc of the 36th IEEE Int Conf on Data Engineering (ICDE). Piscataway, NJ: IEEE, 2020: 973-984
- [16] Sun Yong, Wang Jun, Tan Wenan. DynamicWorker-and-Task assignment on uncertain spatial crowdsourcing[C] //Proc of the 22nd IEEE Int Conf on Computer Supported Cooperative Work in Design. Piscataway, NJ: IEEE, 2018: 755-760
- [17] Cheng Peng, Lian Xiang, Chen Lei, et al. Prediction-based task assignment in spatial crowdsourcing[C] //Proc of the 33rd IEEE Int Conf on Data Engineering (ICDE). Piscataway, NJ: IEEE, 2017: 997-1008
- [18] Zhao Yan, Zheng Kai, Cui Yue, et al. Predictive task assignment in spatial crowdsourcing: A data-driven approach [C] //Proc of the 36th IEEE Int Conf on Data Engineering (ICDE). Piscataway, NJ: IEEE, 2020: 13-24

- [19] Lee J G, Han Jiawei, Whang K Y. Trajectory clustering: A partition-and-group framework[C] //Proc of the 2007 ACM SIGMOD Int Conf on Management of data. New York: ACM, 2007: 593–604
- [20] Wang Tianben, Zhang Daqing, Zhou Xingshe, et al. Mining Personal Frequent Routes Via Road Corner Detection[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2015, 46(4): 445–458
- [21] Li Guohao, Muller M, Thabet A, et al. Deepgcns: CanGCNS Go as deep as CNNs? [C] //Proc of the IEEE/CVF Int Conf on Computer Vision. Piscataway, NJ: IEEE, 2019: 9267–9276
- [22] Silverman B W. Density estimation for statistics and data analysis chapman and hall[J]. Journal of the American Statistical Association, 1987, 150(4): 403–404
- [23] Shi Xingjian, Chen Zhourong, Wang Hao, et al. Convolutional LSTM network: A machine learning approach for precipitation nowcasting[C] //Proc of Advances in Neural Information Processing Systems. Cambridge, MA: MIT Press, 2015: 802–810
- [24] Qiao Shaojie, Shen Dayong, Wang Xiaotian, et al. A self-adaptive parameter selection trajectory prediction approach via hidden markov models [J]. IEEE Transactions on Intelligent Transportation Systems, 2015, 16(1): 284–296
- [25] Qiao Shaojie, Han Nan, Zhu W, et al. TraPlan: An effective three-in-one trajectory-prediction model in transportation networks [J]. IEEE Transactions on Intelligent Transportation Systems, 2015, 16(3): 1188–1198
- [26] Qiao Shaojie, Han Nan, Wang Junfeng, et al. Predicting long-term trajectories of connected vehicles via prefix-projection technique[J]. IEEE Transactions on Intelligent Transportation Systems, 2018, 19(7): 2305–2315
- [27] Jing Yao, Guo Bin, Chen Huihui, et al. CrowdTracker: Object tracking using mobile crowd sensing[J]. Journal of Computer Research and Development, 2019, 56(2): 328–337 (in Chinese)
(景瑶, 郭斌, 陈荟慧, 等. CrowdTracker: 一种基于移动群智感知的目标跟踪方法[J]. 计算机研究与发展, 2019, 56(2): 328–337)



Xu Tiancheng, born in 1997. Master candidate. Member of CCF. His main research interest is space crowdsourcing.
徐天承, 1997 年生. 硕士研究生. CCF 会员. 主要研究方向为空间众包.



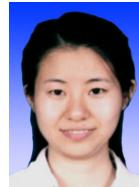
Qiao Shaojie, born in 1981. Postdoc, professor. Distinguished member of CCF. His main research interests include spatio-temporal databases, urban computing and artificial intelligence.

乔少杰, 1981 年生. 博士后, 教授, CCF 杰出会员. 主要研究方向为时空数据库、城市计算、人工智能.



Wu Jun, born in 1983. Master. His main research interest is artificial intelligence.

武俊, 1983 年生. 硕士. 主要研究方向为人工智能.



Han Nan, born in 1984. PhD, associate professor. Her main research interest is spatio-temporal databases.

韩楠, 1984 年生. 博士, 副教授. 主要研究方向为时空数据库.



Yue Kun, born in 1979. PhD, professor, PhD supervisor. His main research interest is artificial intelligence.

岳昆, 1979 年生. 博士, 教授, 博士生导师. 主要研究方向为人工智能.



Yi Yugen, born in 1986. PhD, associate professor. His main research interests include machine learning, deep learning.

易玉根, 1986 年生. 博士, 副教授. 主要研究方向为机器学习、深度学习.



Huang Faliang, born in 1981. PhD, associate professor. His main research interest is data mining.

黄发良, 1981 年生. 博士, 副教授. 主要研究方向为数据挖掘.



Yuan Chang'an, born in 1964. PhD, professor, PhD supervisor. His main research interest is databases.

元昌安, 1964 年生. 博士, 教授, 博士生导师. 主要研究方向为数据库.