(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 113806028 A
(43)申请公布日 2021.12.17

(21)申请号 202010545869.X

(22)申请日 2020.06.16

(71)申请人 阿里巴巴集团控股有限公司
地址 英属开曼群岛大开曼资本大厦一座四层847号邮箱

(72)发明人 曾凯 侯震宇 关涛

(74)专利代理机构 北京安信方达知识产权代理有限公司 11262
代理人 陶丽 栗若木

(51)Int.Cl.
G06F 9/48(2006.01)
G06N 3/04(2006.01)
G06N 3/08(2006.01)

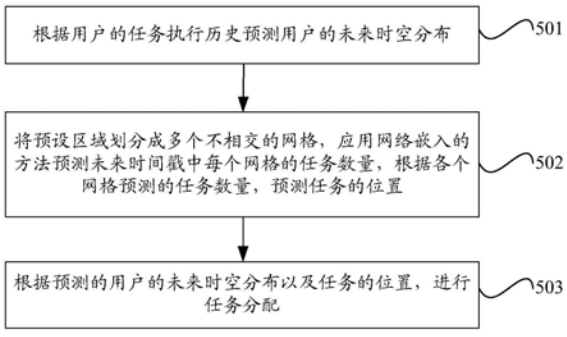
权利要求书5页 说明书28页 附图14页

(54)发明名称

一种空间众包任务分配方法及系统、计算机可读存储介质

(57)摘要

本申请公开了一种空间众包任务分配方法及系统、计算机可读存储介质，所述空间众包任务分配方法包括：根据用户的任务执行历史预测用户的未来时空分布；将预设区域划分成多个不相交的网格，应用网络嵌入的方法预测未来时间戳中每个网格的任务数量，根据各个网格预测的任务数量，预测任务的位置；根据预测的用户的未来时空分布以及任务的位置，进行任务分配。本申请通过预测用户的未来时空分布、任务的数量和位置，能够在任务分配过程中达到最佳分配。



CN 113806028 A

1.一种空间众包任务分配方法,其特征在于,包括:

根据用户的任务执行历史预测用户的未来时空分布;

将预设区域划分成多个不相交的网格,应用网络嵌入的方法预测未来时间戳中每个网格的任务数量,根据各个网格预测的任务数量,预测任务的位置;

根据预测的用户的未来时空分布以及任务的位置,进行任务分配。

2.根据权利要求1所述的空间众包任务分配方法,其特征在于,所述根据用户的任务执行历史预测用户的未来时空分布,包括:

将用户的任务执行历史作为时间序列数据,利用序列模式挖掘方法从所述时间序列数据中挖掘出现频率超过最小支持阈值的时间实例集合,并将每个连续的时间实例集合作为用户的可用时间;

在所述可用时间开始时,基于时空递归神经网络模型预测每个未来用户的空间分布。

3.根据权利要求2所述的空间众包任务分配方法,其特征在于,所述每个未来用户的空间分布包括每个未来用户的位置;

所述时空递归神经网络模型包括输入层、隐藏层和输出层,所述输入层包含用户w在时间t_i到达位置的潜在向量$q_{l_{t_i}^w} \in \mathbb{R}^d$,所述隐藏层包含用户w在当前时间实例t的向量表示$h_{t,l_t^w}^w$,且

$$h_{t,l_t^w}^w = f\Big(\sum_{l_{t_i}^w \in L^w, t-\hat{v}<t_i<t} S_{l_t^w - l_{t_i}^w} T_{t-t_i}\, q_{l_{t_i}^w} + Ch_{t-\hat{v},l_{t-\hat{v}}^w}^w\Big)$$

其中,W和L分别为一组潜在的用户集合及位置集合,$p_w \in \mathbb{R}^d$和$q_l \in \mathbb{R}^d$分别表示用户w和位置l的潜在向量,每个位置和坐标相关,每个用户w有一系列刚刚经过的历史位置的集合,即$L^w = \{l_{t_1}^w, l_{t_2}^w, \ldots\}$,v表示时间窗口的宽度,使用近似值$\hat{v}$作为局部窗口宽度,所述近似值$\hat{v}$最接近于v且$l_{t-\hat{v}}^w \in L^w$,$S_{l_t^w - l_{t_i}^w}$表示$l_t^w$和$l_{t_i}^w$之间地理距离的转移矩阵,$T_{t-t_i}$表示时间间隔$t-t_i$的转移矩阵,C是先前状态传播顺序信号的循环连接,激活函数f(x)使用Sigmod函数$f(x)=\exp(1/1+e^{-x})$;

所述输出层包含用户w在时间t到达位置l的概率$O_{w,t,l}$,且

$$O_{w,t,l} = (h_{t,l}^w + p_w)^T q_l$$

其中,$h_{t,l}^w$在特定的时空情境中捕捉动态兴趣,p_w是表明其兴趣和活动范围的永久表示;

所述在所述可用时间开始时,基于时空递归神经网络模型预测每个未来用户的空间分布,包括:

将时间间隔和地理距离分别划分到离散仓,学习对应的离散仓的上下边界的转移矩阵并且通过线性插值的方法计算其他时间间隔和地理距离的转移矩阵,以估计用户可用时间开始时的位置;

使用贝叶斯个性化排名和基于时间的反向传播方法训练所述时空递归神经网络模型;

将当前用户的历史位置数据输入所述时空递归神经网络模型;

计算输出层中的位置表示和用户的内积,预测用户w在时间t最大概率出现的位置。

4.根据权利要求2所述的空间众包任务分配方法，其特征在于，所述每个未来用户的空间分布包括每个未来用户的路线，所述基于时空递归神经网络模型预测每个未来用户的空间分布，包括：

使用基于特征点的道路拐角提取方法从历史轨迹中提取道路拐角；

使用基于道路拐角的轨迹映射方法表示历史轨迹数据和将被预测的查询轨迹，生成以道路拐角为中心的路线；

使用前缀投射顺序模式挖掘算法从历史轨迹中挖掘数量大于最小支持度阈值的轨迹，作为满足最小支持度阈值的频繁移动模式；

基于被发现的频繁移动模式，使用模式匹配方法以预测未来路线；

当无模式匹配时，使用所述时空递归神经网络模型预测下一个行驶中的道路拐角。

5.根据权利要求4所述的空间众包任务分配方法，其特征在于，

所述使用基于特征点的道路拐角提取方法从历史轨迹中提取道路拐角，包括：通过线性拟合的方法生成一组特征点，所述特征点为轨迹方向发生显著变化的坐标点，使用基于噪音的多密度空间聚类方法检测道路拐角；

所述使用模式匹配方法以预测未来路线，包括：执行模式匹配过程以找到待匹配轨迹，通过计算待匹配轨迹与频繁移动模式的匹配覆盖率，找到最长的轨迹模式作为待匹配轨迹的预测路线。

6.根据权利要求1所述的空间众包任务分配方法，其特征在于，所述应用网络嵌入的方法预测未来时间戳中每个网格的任务数量，包括：

基于任务的时空信息建立基于空间任务的网络，所述网络的节点包括基于时间单元的节点$C_i^{t_j}$和基于空间任务的节点B_m，所述网络的边包括空间邻近边$e\left(C_i^{t_j}, C_k^{t_g}\right)$和任务相关边$e\left(C_i^{t_j}, B_m\right)$，所述基于时间单元的节点$C_i^{t_j}$表示在时间戳$t_j$处网格的特定空间单元$C_i$，所述基于空间任务的节点$B_m$表示特定任务类别m的节点，所述空间邻近边$e\left(C_i^{t_j}, C_k^{t_g}\right)$表示两个基于时间单元的节点的空间邻近关系，任务相关边$e\left(C_i^{t_j}, B_m\right)$表示有特定类别m的任务在时间戳$t_j$发布在空间单元$C_i$中；

建立空间路径以捕捉空间信息，建立任务相关路径以捕捉与任务相关的信息，建立时间排序的任务相关路径以捕捉时间信息，所述空间路径包括基于时间单元的节点和空间邻近节点，所述任务相关路径包括基于时间单元的节点、基于空间任务的节点和任务相关边，所述时间排序的任务相关路径包括基于时间单元的节点、基于空间任务的节点和任务相关边，且所述基于时间单元的节点按照一个时间单位的增长速率根据时间以升序排序；

沿着所述空间路径、任务相关路径以及时间排序的任务相关路径，使用λ长度的随机游走方法，以基于空间任务的网络G作为输入，获得每个节点多个λ长度的随机游走序列；

通过跳字SkipGram神经网络模型学习每个节点的向量表示，基于历史数据和其他节点数据，使用回归算法估计每个网格中的任务数量。

7.根据权利要求6所述的空间众包任务分配方法，其特征在于，所述每个网格中的任务数量，通过如下公式估计：

$$N_i^{t_{j+1}} = \alpha N_i^{t_j} + (1-\alpha)\sum_{C_{i'}^{t_j} \in \mathbb{C}^{t_j}, i' \neq i} \frac{\mathrm{sim}\left(C_i^{t_j}, C_{i'}^{t_j}\right)}{\sum_{C_{i'}^{t_j} \in \mathbb{C}^{t_j}, i' \neq i} \mathrm{sim}\left(C_i^{t_j}, C_{i'}^{t_j}\right)} N_{i'}^{t_j} + \beta;$$

其中，$N_i^{t_j}$表示节点$C_i^{t_j}$的任务数量，$\mathbb{C}^{t_j}$表示在时间t_j处的所有节点，$\mathrm{sim}\left(C_i^{t_j}, C_{i'}^{t_j}\right)$表示节点$C_i^{t_j}$和$C_{i'}^{t_j}$之间的相关性，所述相关性由两者向量的点积计算；$\sum_{C_{i'}^{t_j} \in \mathbb{C}^{t_j}, i' \neq i} \frac{\mathrm{sim}\left(C_i^{t_j}, C_{i'}^{t_j}\right)}{\sum_{C_{i'}^{t_j} \in \mathbb{C}^{t_j}, i' \neq i} \mathrm{sim}\left(C_i^{t_j}, C_{i'}^{t_j}\right)} N_{i'}^{t_j}$表示两个节点之间的传播效果；α为用于控制历史数据和其他节点贡献的第一参数，β为用于避免过拟合的第二参数；α和β通过随机梯度下降算法进行训练并且使用均方误差作为损失函数。

8.根据权利要求1所述的空间众包任务分配方法，其特征在于，所述根据预测的用户的未来时空分布以及任务的位置，进行任务分配，包括：

基于用户可达距离和任务失效时间的约束，建立在给定时间段中每个用户能够执行的可达任务集合；

通过迭代地扩展可达任务集合，计算每个用户的所有极大有效任务集的集合，极大有效任务集的计算公式如下：

$$\mathrm{opt}(Q, s_j) = \begin{cases} 1, & |Q| = 1 \\ \max\limits_{s_i \in Q, s_i \neq s_j} \mathrm{opt}\left(Q - \{s_j\}, s_i\right) + \delta_{ij}, & \text{其他} \end{cases}$$

$$\delta_{ij} = \begin{cases} 1, & t(s_j.l) \leq s_j.e, t(s_j.l) \leq t+n \\ 0, & \text{其他} \end{cases}$$

其中，opt(Q,s_j)为在用户的可达范围内通过调度Q中的所有任务最终被完成的最大任务数量，其中，Q中的任务从w.l到s_j.l，R为实现这一最大任务数量的有序任务序列，s_i表示R中到达任务s_j之前的前一个任务，R′表示opt(Q-{s_j},s_i))中对应的任务序列，δ_{ij}表示在给定时间段T＝{t,t+1,...,t+n}内在将s_j附加在R′后可以完成s_j的概率；

根据每个用户的所有极大有效任务集的集合，进行任务分配。

9.根据权利要求8所述的空间众包任务分配方法，其特征在于，所述根据每个用户的所有极大有效任务集的集合，进行任务分配，包括：

将用户集合和任务集合作为输入，初始化一个空的任务分配；

开始迭代，在每次迭代中，从剩下的将分配的用户中随机选择一个用户并且从未分配的任务中为选择的用户找到最大的极大有效任务集，并将找到的最大的极大有效任务集加入到当前任务分配中；

迭代结束，在所有迭代中找到最大任务分配。

10.根据权利要求8所述的空间众包任务分配方法，其特征在于，所述根据每个用户的所有极大有效任务集的集合，进行任务分配，包括：

对于用户集合和任务集合，根据用户之间的依赖关系构造用户依赖图，所述依赖关系

包括互相独立或互相依赖,当两个用户没有共同的可达任务时,两个用户互相独立;当两个用户有共同的可达任务时,两个用户互相依赖;

将所述用户依赖图划分成多个节点集,以分解用户的依赖关系;

根据划分的多个节点集,建立平衡树,所述平衡树中的兄弟节点互相独立;

通过深度优先搜索算法为树节点中的每个用户计算有效任务集合,以找到最优分配。

11.根据权利要求10所述的空间众包任务分配方法,其特征在于,所述将所述用户依赖图划分成多个节点集,以分解用户的依赖关系,包括:

按照节点度数的升序依次对用户依赖图中的节点进行简约处理,直至满足下列条件之一:(1)原图被简约成一个简单图或空图;(2)不存在度小于或等于k的节点;所述简约处理包括:对于指定的度i,i为自然数且0≤i≤k,找到度为i的节点v,并检查节点v的所有邻居是否构成一个团,所述团指一个两两之间有边的节点集合,若无法构成一个团,通过添加边来构造团;将节点v和它的邻居节点构成的团存储于一个堆栈中,在原图中删除节点v及其对应的边;

堆栈中的所有团和未被所述简约处理删除的团构成划分成的多个节点集。

12.根据权利要求10所述的空间众包任务分配方法,其特征在于,所述根据划分的多个节点集,建立平衡树,包括:

从用户依赖图中删除每个节点集的节点,将用户依赖图划分为多个子图,其中,最大的子图记录为Gmax,选择能生成最小|Gmax|值的节点集作为根节点集Xmin,其中,|Gmax|是图Gmax的节点数量;

对删除根节点集Xmin后的各个子图,采用k度图简约算法继续划分节点集,并对继续划分的节点集,递归地寻找根节点集。

13.一种空间众包任务分配方法,其特征在于,包括:

云端服务器接收客户端的任务分配请求,获取用户的任务执行历史以及已发布的任务的时空信息;

云端服务器根据用户的任务执行历史预测用户的未来时空分布,并将预设区域划分成多个不相交的网格,应用网络嵌入的方法预测未来时间戳中每个网格的任务数量,根据各个网格预测的任务数量,预测任务的位置;

云端服务器根据预测的用户的未来时空分布以及任务的位置,进行任务分配,将任务分配的结果发送至客户端。

14.一种空间众包任务分配系统,其特征在于,包括处理器和存储器,所述处理器用于执行存储器中存储的计算机程序以实现如权利要求1至13任意一项所述的空间众包任务分配方法的步骤。

15.一种计算机可读存储介质,其特征在于,所述计算机可读存储介质存储有计算机程序,其特征在于,所述计算机程序被处理器执行时实现如权利要求1至13任意一项所述的空间众包任务分配方法的步骤。

16.一种空间众包任务分配系统,其特征在于,包括第一预测模块、第二预测模块和任务分配模块,其中:

所述第一预测模块,用于根据用户的任务执行历史预测用户的未来时空分布;

所述第二预测模块,用于将预设区域划分成多个不相交的网格,应用网络嵌入的方法

预测未来时间戳中每个网格的任务数量，根据各个网格预测的任务数量，预测任务的位置；

所述任务分配模块，用于根据预测的用户的未来时空分布以及任务的位置，进行任务分配。

一种空间众包任务分配方法及系统、计算机可读存储介质

技术领域

[0001] 本申请涉及但不限于计算机技术领域，尤其涉及一种空间众包任务分配方法及系统、计算机可读存储介。

背景技术

[0002] 随着配备全球卫星定位系统(Global Positioning System,GPS)的智能设备的普及以及无线网络的高可用性，一种新型的众包服务使人们能够以多模式传感器的方式移动，从而即时收集和共享各种类型的高保真时空数据，即空间众包(Spatial Crowd Sourcing,SC)。具体来说，通过空间众包，请求者可以动态的向SC服务器发布空间任务，例如，拍照、录像以及报告当地的热点等，服务器根据用户的地点和其他约束将服务器上的任务分配给用户，该过程称为任务分配(Task Assignment)。

[0003] 现有对于空间众包的研究大多集中在任务分配。现有的任务分配研究往往侧重于静态离线场景，即用户和任务的位置已经事先给出。但是在实际情况中，空间众包是一个实时平台，用户和任务可以动态上线并且其地点未知。一些任务分配研究还探索了SC中的在线分配方法，主要是基于当前的任务分配将新到达的任务分配给合适的用户，但是这些研究没有考虑到未进入系统的未来用户/任务，只是尝试最大化当前分配，即获得局部最优解而不是全局最优解。由于即将到来用户/任务的动态性，现有的任务分配方法无法在在线场景中实现最佳解决方案。

发明内容

[0004] 本申请提供了一种空间众包任务分配方法及系统、计算机可读存储介质，能够在任务分配过程中达到最佳分配。

[0005] 本申请实施例提供了一种空间众包任务分配方法，包括：根据用户的任务执行历史预测用户的未来时空分布；将预设区域划分成多个不相交的网格，应用网络嵌入的方法预测未来时间戳中每个网格的任务数量，根据各个网格预测的任务数量，预测任务的位置；根据预测的用户的未来时空分布以及任务的位置，进行任务分配。

[0006] 在一些可能的实现方式中，所述根据用户的任务执行历史预测用户的未来时空分布，包括：将用户的任务执行历史作为时间序列数据，利用序列模式挖掘方法从所述时间序列数据中挖掘出现频率超过最小支持阈值的时间实例集合，并将每个连续的时间实例集合作为用户的可用时间；在所述可用时间开始时，基于时空递归神经网络模型预测每个未来用户的空间分布。

[0007] 在一些可能的实现方式中，所述每个未来用户的空间分布包括每个未来用户的位置。

[0008] 所述时空递归神经网络模型包括输入层、隐藏层和输出层，所述输入层包含用户w在时间t_i到达位置的潜在向量$q_{l_{t_i}^w} \in \mathbb{R}^d$，所述隐藏层包含用户w在当前时间实例t的向量

表示$h^{w}_{t,l^{w}_{t}}$，且

[0009] $$h^{w}_{t,l^{w}_{t}} = f\Big(\sum_{l^{w}_{t_i}\in L^{w},\, t-\hat{v}<t_i<t} S_{l^{w}_{t}-l^{w}_{t_i}} T_{t-t_i}\, q_{l^{w}_{t_i}} + Ch^{w}_{t-\hat{v},l^{w}_{t-\hat{v}}}\Big)$$

[0010] 其中，W和L分别为一组潜在的用户集合及位置集合，$p_w \in \mathbb{R}^d$和$q_l \in \mathbb{R}^d$分别表示用户w和位置l的潜在向量，每个位置和坐标相关，每个用户w有一系列刚刚经过的历史位置的集合，即$L^w = \{l^w_{t_1}, l^w_{t_2}, \ldots\}$，v表示时间窗口的宽度，使用近似值$\hat{v}$作为局部窗口宽度，所述近似值$\hat{v}$最接近于v且$l^w_{t-\hat{v}} \in L^w$，$S_{l^w_t - l^w_{t_i}}$表示$l^w_t$和$l^w_{t_i}$之间地理距离的转移矩阵，$T_{t-t_i}$表示时间间隔$t-t_i$的转移矩阵，C是先前状态传播顺序信号的循环连接，激活函数f(x)使用Sigmod函数$f(x)=\exp(1/1+e^{-x})$。

[0011] 所述输出层包含用户w在时间t到达位置l的概率$O_{w,t,l}$，且

[0012] $$O_{w,t,l} = (h^{w}_{t,l} + p_w)^{T} q_l$$

[0013] 其中，$h^{w}_{t,l}$在特定的时空情境中捕捉动态兴趣，p_w是表明其兴趣和活动范围的永久表示。

[0014] 所述在所述可用时间开始时，基于时空递归神经网络模型预测每个未来用户的空间分布，包括：将时间间隔和地理距离分别划分到离散仓，学习对应的离散仓的上下边界的转移矩阵并且通过线性插值的方法计算其他时间间隔和地理距离的转移矩阵，以估计用户可用时间开始时的位置；使用贝叶斯个性化排名和基于时间的反向传播方法训练所述时空递归神经网络模型；将当前用户的历史位置数据输入所述时空递归神经网络模型；计算输出层中的位置表示和用户的内积，预测用户w在时间t最大概率出现的位置。

[0015] 在一些可能的实现方式中，所述每个未来用户的空间分布包括每个未来用户的路线，所述基于时空递归神经网络模型预测每个未来用户的空间分布，包括：使用基于特征点的道路拐角提取方法从历史轨迹中提取道路拐角；使用基于道路拐角的轨迹映射方法表示历史轨迹数据和将被预测的查询轨迹，生成以道路拐角为中心的路线；使用前缀投射顺序模式挖掘算法从历史轨迹中挖掘数量大于最小支持度阈值的轨迹，作为满足最小支持度阈值的频繁移动模式；基于被发现的频繁移动模式，使用模式匹配方法以预测未来路线；当无模式匹配时，使用所述时空递归神经网络模型预测下一个行驶中的道路拐角。

[0016] 在一些可能的实现方式中，所述使用基于特征点的道路拐角提取方法从历史轨迹中提取道路拐角，包括：通过线性拟合的方法生成一组特征点，所述特征点为轨迹方向发生显著变化的坐标点，使用基于噪音的多密度空间聚类方法检测道路拐角。

[0017] 在一些可能的实现方式中，所述使用模式匹配方法以预测未来路线，包括：执行模式匹配过程以找到待匹配轨迹，通过计算待匹配轨迹与频繁移动模式的匹配覆盖率，找到最长的轨迹模式作为待匹配轨迹的预测路线。

[0018] 在一些可能的实现方式中，所述应用网络嵌入的方法预测未来时间戳中每个网格的任务数量，包括：

[0019] 基于任务的时空信息建立基于空间任务的网络，所述网络的节点包括基于时间单

元的节点$C_i^{t_j}$和基于空间任务的节点B_m，所述网络的边包括空间邻近边$e\left(C_i^{t_j}, C_k^{t_g}\right)$和任务相关边$e\left(C_i^{t_j}, B_m\right)$，所述基于时间单元的节点$C_i^{t_j}$表示在时间戳$t_j$处网格的特定空间单元$C_i$，所述基于空间任务的节点$B_m$表示特定任务类别m的节点，所述空间邻近边$e\left(C_i^{t_j}, C_k^{t_g}\right)$表示两个基于时间单元的节点的空间邻近关系，任务相关边$e\left(C_i^{t_j}, B_m\right)$表示有特定类别m的任务在时间戳$t_j$发布在空间单元$C_i$中；建立空间路径以捕捉空间信息，建立任务相关路径以捕捉与任务相关的信息，建立时间排序的任务相关路径以捕捉时间信息，所述空间路径包括基于时间单元的节点和空间邻近节点，所述任务相关路径包括基于时间单元的节点、基于空间任务的节点和任务相关边，所述时间排序的任务相关路径包括基于时间单元的节点、基于空间任务的节点和任务相关边，且所述基于时间单元的节点按照一个时间单位的增长速率根据时间以升序排序；沿着所述空间路径、任务相关路径以及时间排序的任务相关路径，使用λ长度的随机游走方法，以基于空间任务的网络G作为输入，获得每个节点多个λ长度的随机游走序列；通过跳字SkipGram神经网络模型学习每个节点的向量表示，基于历史数据和其他节点数据，使用回归算法估计每个网格中的任务数量。

[0020] 在一些可能的实现方式中，所述每个网格中的任务数量，通过如下公式估计：

[0021] $$N_i^{t_{j+1}} = \alpha N_i^{t_j} + (1-\alpha)\sum_{C_{i'}^{t_j}\in\mathbb{C}^{t_j}, i'\neq i} \frac{\mathrm{sim}\left(C_i^{t_j}, C_{i'}^{t_j}\right)}{\sum_{C_{i'}^{t_j}\in\mathbb{C}^{t_j}, i'\neq i}\mathrm{sim}\left(C_i^{t_j}, C_{i'}^{t_j}\right)} N_{i'}^{t_j} + \beta;$$

[0022] 其中，$N_i^{t_j}$表示节点$C_i^{t_j}$的任务数量，$\mathbb{C}^{t_j}$表示在时间t_j处的所有节点，$\mathrm{sim}\left(C_i^{t_j}, C_{i'}^{t_j}\right)$表示节点$C_i^{t_j}$和$C_{i'}^{t_j}$之间的相关性，所述相关性由两者向量的点积计算；$\sum_{C_{i'}^{t_j}\in\mathbb{C}^{t_j}, i'\neq i} \frac{\mathrm{sim}\left(C_i^{t_j}, C_{i'}^{t_j}\right)}{\sum_{C_{i'}^{t_j}\in\mathbb{C}^{t_j}, i'\neq i}\mathrm{sim}\left(C_i^{t_j}, C_{i'}^{t_j}\right)} N_{i'}^{t_j}$表示两个节点之间的传播效果；α为用于控制历史数据和其他节点贡献的第一参数，β为用于避免过拟合的第二参数；α和β通过随机梯度下降算法进行训练并且使用均方误差作为损失函数。

[0023] 在一些可能的实现方式中，所述根据预测的用户的未来时空分布以及任务的位置，进行任务分配，包括：基于用户可达距离和任务失效时间的约束，建立在给定时间段中每个用户能够执行的可达任务集合；通过迭代地扩展可达任务集合，计算每个用户的所有极大有效任务集的集合，极大有效任务集的计算公式如下：

[0024] $$\mathrm{opt}(Q, s_j) = \begin{cases} 1, & |Q| = 1 \\ \max\limits_{s_i\in Q, s_i\neq s_j} \mathrm{opt}\left(Q - \{s_j\}, s_i\right) + \delta_{ij}, & \text{其他} \end{cases}$$

[0025] $$\delta_{ij}=\begin{cases}1, & t(s_j.l)\le s_j.e, t(s_j.l)\le t+n\\ 0, & \text{其他}\end{cases}$$

[0026] 其中，opt(Q,s_j)为在用户的可达范围内通过调度Q中的所有任务最终被完成的最大任务数量，其中，Q中的任务从w.l到s_j.l，R为实现这一最大任务数量的有序任务序列，s_i表示R中到达任务s_j之前的前一个任务，R′表示opt(Q-{s_j},s_i))中对应的任务序列，δ_{ij}表示在给定时间段T＝{t,t+1,...,t+n}内在将s_j附加在R′后可以完成s_j的概率；

[0027] 根据每个用户的所有极大有效任务集的集合，进行任务分配。

[0028] 在一些可能的实现方式中，所述根据每个用户的所有极大有效任务集的集合，进行任务分配，包括：将用户集合和任务集合作为输入，初始化一个空的任务分配；开始迭代，在每次迭代中，从剩下的将分配的用户中随机选择一个用户并且从未分配的任务中为选择的用户找到最大的极大有效任务集，并将找到的最大的极大有效任务集加入到当前任务分配中；迭代结束，在所有迭代中找到最大任务分配。

[0029] 在一些可能的实现方式中，所述根据每个用户的所有极大有效任务集的集合，进行任务分配，包括：对于用户集合和任务集合，根据用户之间的依赖关系构造用户依赖图，所述依赖关系包括互相独立或互相依赖，当两个用户没有共同的可达任务时，两个用户互相独立；当两个用户有共同的可达任务时，两个用户互相依赖；将所述用户依赖图划分成多个节点集，以分解用户的依赖关系；根据划分的多个节点集，建立平衡树，所述平衡树中的兄弟节点互相独立；通过深度优先搜索算法为树节点中的每个用户计算有效任务集合，以找到最优分配。

[0030] 在一些可能的实现方式中，所述将所述用户依赖图划分成多个节点集，以分解用户的依赖关系，包括：按照节点度数的升序依次对用户依赖图中的节点进行简约处理，直至满足下列条件之一：(1)原图被简约成一个简单图或空图；(2)不存在度小于或等于k的节点；所述简约处理包括：对于指定的度i，i为自然数且0≤i≤k，找到度为i的节点v，并检查节点v的所有邻居是否构成一个团，所述团指一个两两之间有边的节点集合，若无法构成一个团，通过添加边来构造团；将节点v和它的邻居节点构成的团存储于一个堆栈中，在原图中删除节点v及其对应的边；堆栈中的所有团和未被所述简约处理删除的团构成划分成的多个节点集。

[0031] 在一些可能的实现方式中，所述根据划分的多个节点集，建立平衡树，包括：从用户依赖图中删除每个节点集的节点，将用户依赖图划分为多个子图，其中，最大的子图记录为Gmax，选择能生成最小|Gmax|值的节点集作为根节点集Xmin，其中，|Gmax|是图Gmax的节点数量；对删除根节点集Xmin后的各个子图，采用k度图简约算法继续划分节点集，并对继续划分的节点集，递归地寻找根节点集。

[0032] 本申请实施例还提供了一种空间众包任务分配方法，包括：云端服务器接收客户端的任务分配请求，获取用户的任务执行历史以及已发布的任务的时空信息；云端服务器根据用户的任务执行历史预测用户的未来时空分布，并将预设区域划分成多个不相交的网格，应用网络嵌入的方法预测未来时间戳中每个网格的任务数量，根据各个网格预测的任务数量，预测任务的位置；云端服务器根据预测的用户的未来时空分布以及任务的位置，进行任务分配，将任务分配的结果发送至客户端。

[0033] 本申请实施例还提供了一种空间众包任务分配系统，包括处理器和存储器，所述处理器用于执行存储器中存储的计算机程序以实现如上所述的空间众包任务分配方法的步骤。

[0034] 本申请实施例还提供了一种计算机可读存储介质，所述计算机可读存储介质存储有计算机程序，所述计算机程序被处理器执行时实现如上所述的空间众包任务分配方法的步骤。

[0035] 本申请实施例还提供了一种空间众包任务分配系统，包括第一预测模块、第二预测模块和任务分配模块，其中：所述第一预测模块，用于根据用户的任务执行历史预测用户的未来时空分布；所述第二预测模块，用于将预设区域划分成多个不相交的网格，应用网络嵌入的方法预测未来时间戳中每个网格的任务数量，根据各个网格预测的任务数量，预测任务的位置；所述任务分配模块，用于根据预测的用户的未来时空分布以及任务的位置，进行任务分配。

[0036] 本申请的空间众包任务分配方法及系统、计算机可读存储介质，包括根据用户的任务执行历史预测用户的未来时空分布；将预设区域划分成多个不相交的网格，应用网络嵌入的方法预测未来时间戳中每个网格的任务数量，根据各个网格预测的任务数量，预测任务的位置；根据预测的用户的未来时空分布以及任务的位置，进行任务分配，保证了在任务分配过程中达到最佳分配。

[0037] 本申请的其它特征和优点将在随后的说明书中阐述，并且，部分地从说明书中变得显而易见，或者通过实施本申请而了解。本申请的其他优点可通过在说明书以及附图中所描述的方案来实现和获得。

附图说明

[0038] 附图用来提供对本申请技术方案的理解，并且构成说明书的一部分，与本申请的实施例一起用于解释本申请的技术方案，并不构成对本申请技术方案的限制。

[0039] 图1为本申请实施例的一种动态空间任务分配问题的场景示意图；

[0040] 图2为本申请实施例的一种最大化当前分配的任务分配结果示意图；

[0041] 图3为本申请实施例的一种根据位置进行分配的任务分配结果示意图；

[0042] 图4为本申请实施例的一种根据路线进行分配的任务分配结果示意图；

[0043] 图5(a)为本申请实施例的一种空间众包任务分配方法的流程示意图；

[0044] 图5(b)为本申请实施例的另一种空间众包任务分配方法的流程示意图；

[0045] 图6为本申请实施例的一种ST-RNN神经网络模型的架构示意图；

[0046] 图7为本申请实施例的一种路线预测过程的流程示意图；

[0047] 图8为本申请实施例的一种基于空间任务的网络的结构示意图；

[0048] 图9为本申请实施例的一种用户依赖图简约过程示意图；

[0049] 图10为本申请实施例的一种用户依赖图图划分结果示意图；

[0050] 图11为本申请实施例的一种用户依赖图树构建结果示意图；

[0051] 图12(a)至图12(b)为本申请实施例的训练集$|\mathcal{T}|$的大小对用户位置预测的性能影响的实验结果图；

[0052] 图13(a)至图13(b)为本申请实施例的训练集$|\mathcal{T}|$的大小对用户路线预测的性能影

响的实验结果图；

[0053] 图14(a)至图14(b)为本申请实施例的训练集$|\mathcal{T}|$的大小对任务预测的性能影响的实验结果图；

[0054] 图15(a)至图15(d)为本申请实施例的任务数量对任务分配的性能影响的实验结果图；

[0055] 图16(a)至图16(d)为本申请实施例的任务有效时间对任务分配的性能影响的实验结果图；

[0056] 图17(a)至图17(d)为本申请实施例的用户的可达半径对任务分配的性能影响的实验结果图；

[0057] 图18为本申请实施例的一种空间众包任务分配系统的结构示意图；

[0058] 图19为本申请实施例的又一种空间众包任务分配方法的流程示意图；

[0059] 图20为本申请实施例的一种空间众包任务分配方法的应用场景示意图。

具体实施方式

[0060] 本申请描述了多个实施例，但是该描述是示例性的，而不是限制性的，并且对于本领域的普通技术人员来说显而易见的是，在本申请所描述的实施例包含的范围内可以有更多的实施例和实现方案。尽管在附图中示出了许多可能的特征组合，并在具体实施方式中进行了讨论，但是所公开的特征的许多其它组合方式也是可能的。除非特意加以限制的情况以外，任何实施例的任何特征或元件可以与任何其它实施例中的任何其他特征或元件结合使用，或可以替代任何其它实施例中的任何其他特征或元件。

[0061] 本申请包括并设想了与本领域普通技术人员已知的特征和元件的组合。本申请已经公开的实施例、特征和元件也可以与任何常规特征或元件组合，以形成由权利要求限定的独特的发明方案。任何实施例的任何特征或元件也可以与来自其它发明方案的特征或元件组合，以形成另一个由权利要求限定的独特的发明方案。因此，应当理解，在本申请中示出和/或讨论的任何特征可以单独地或以任何适当的组合来实现。因此，除了根据所附权利要求及其等同替换所做的限制以外，实施例不受其它限制。此外，可以在所附权利要求的保护范围内进行各种修改和改变。

[0062] 此外，在描述具有代表性的实施例时，说明书可能已经将方法和/或过程呈现为特定的步骤序列。然而，在该方法或过程不依赖于本文所述步骤的特定顺序的程度上，该方法或过程不应限于所述的特定顺序的步骤。如本领域普通技术人员将理解的，其它的步骤顺序也是可能的。因此，说明书中阐述的步骤的特定顺序不应被解释为对权利要求的限制。此外，针对该方法和/或过程的权利要求不应限于按照所写顺序执行它们的步骤，本领域技术人员可以容易地理解，这些顺序可以变化，并且仍然保持在本申请实施例的精神和范围内。

[0063] 现有研究表明，大多数人存在重复性的旅程，例如往返工作地点，这使得根据先前的旅行历史预测用户的位置/路线成为可能。另外，通过分析用户执行空间任务的任务执行轨迹，不仅可以了解个人的流动模式，还可以获得个人任务执行行为的宝贵见解，这些可以进一步用于提升空间众包的质量。本申请实施例的空间众包任务分配方法，使用数据驱动的方法预测用户的位置/路线和任务的位置，并且根据此预测使得全局任务分配达到最优。

[0064] 图1描绘了一个动态空间任务分配问题的示例，其中三个用户路径表示为

$\mathbb{P}_{w_1}$，$\mathbb{P}_{w_2}$和$\mathbb{P}_{w_3}$，时空任务表示为{s_1,...,s_8}。每一条路径是一系列伴有时间戳的位置(即路径$\mathbb{P}_{w_2}$的位置l_8^2和l_9^3分别带有时间戳2和3)并且当前实例是3。每个用户w和其可达距离范围(例如，本申请实施例中，可达距离范围可以设为1.2)相关。此外，每个在不同时间发布和到期的空间任务都标有位置并且只能被执行一次。在线空间众包问题是在当前和未来的时间戳下将任务分配给合适的用户，从而使得分配任务的最大化。为了更好的理解用户和任务的时空分布，将图1中的三维空间中的位置点映射到图2的二维空间平面。

[0065] 在不违反用户和任务的时空约束的条件下，可以很直观的看出可以将附近的任务分配给用户，从而在每个时间实例上最大化当前分配，这被称为最大化任务分配(Maximum Task Assignment，MTA)实例问题。因此，在所举的例子中，在当前时间，将任务s_1分配给用户w_2，将s_2分配给用户w_1从而达到任务分配数量的最大化，此时的最大任务分配数量是2。类似的，在下一个时间实例(即时刻4)，将任务s_3分配给用户w_1，任务s_6分配给用户w_3，获得时间实例4上的被分配任务数量的最大化，即2。但是，由于用户无法到达该位置执行他们自己分配的任务后剩下的任务，因此，剩余的任务无法分配给用户。结果，在时间段(例如，时间实例3-6)中这样的分配策略导致分配的任务总数是4(=2+2)，如图2所示。

[0066] 但是，上述分配方法只是尝试最大化当前分配(即局部最优而不是全局最优)，而没有考虑在未来时间实例中会动态出现的未来用户和任务。当事先知道未来用户和任务时，任务分配问题可以简化为最大覆盖率问题以及其变体。然而，SC的主要挑战来自于即将到来用户和任务的动态性，使得在在线场景中无法实现最佳解决方案。

[0067] 本申请实施例提出了一种空间众包任务分配方法，考虑当前用户和任务以及事先未知位置的未来用户和任务，预测未来用户和任务的位置，在任务分配过程中达到最大分配。

[0068] 本申请实施例提出的空间众包任务分配方法，包括用户与任务的预测阶段以及任务调度与分配阶段。

[0069] 用户与任务的预测阶段旨在预测未来时间实例中用户和任务的时空分布。对于用户预测，基于两种不同的任务分配策略，例如，特定位置/路线的任务分配(Location/Route-specific Task Assignment)。本申请实施例引入时空递归神经网络(Spatial-Temporal Recurrent Neural Network，ST-RNN)以预测每个未来用户的出现位置，并且根据未来/当前用户的旅行历史为用户设计一个混合模型，以预测可能的路线。对于任务预测，本申请实施例设计一种路线约束深度游走(Path Constrained DeepWalk，PC-DeepWalk)算法估算未来任务的数量，并且通过将任务视为空间点事件，利用核密度估计(Kernel Density Estimation，KDE)方法预测未来任务的位置。

[0070] 在任务调度与分配阶段，基于当前和未来的用户与任务，首先为每一个用户计算特定位置/路线的极大有效任务集(Location/Route-specific Maximal Valid Task Sets，L/R-MaxVTSs，见下文定义6)。然后，本申请实施例在枚举出每个用户的极大有效任务集的所有可能组合并且随着用户数量的增长呈指数级增长的情况下，解决了巨大搜索空间的计算问题。为了提高效率，本申请实施例提出一种贪心算法，该算法尝试将未分配任务的最大L/R-MaxVTSs分配给用户。本申请实施例还提出基于分解的精确图划分算法，该算法根据任务分配的总数找到最优分配结果。图3通过应用涵盖6个任务的特定精确位置方法描述

任务调度和分配结果，图4通过应用涵盖7个任务的特定确切路线方法描述任务调度和分配结果。

[0071] 本申请实施例中，相关符号的定义为：s表示空间任务；s.r表示空间任务s的发布时间；s.l表示空间任务s的位置；s.e表示空间任务s的失效时间；s.c表示空间任务s的类别；t表示时间实例；T表示时间实例集合；w表示可获得用户；w.l表示用户w目前的位置；w.Φ表示用户w的可达时间；w.range表示用户w的可达范围；$w.\mathbb{P}$表示在用户w的可达时间内的用户路线；R表示一个任务序列；S_w表示用户w的一个任务集合；t(l)表示特定位置l的到达时间；VTS(w)表示用户w的一个有效任务集合；c(a,b)表示从a到b的移动时间；L-VTS(w)表示用户w的一个特定位置的有效任务集合；R-VTS(w)表示用户w的一个特定路径的有效任务集合；L-MaxVTS(w)表示用户w的一个特定位置的极大有效任务集；R-MaxVTS(w)表示用户w的一个特定路径的极大有效任务集；A表示一个空间任务分配；$\mathbb{A}$表示一个空间任务分配集合。

[0072] 在描述本申请的具体实施例之前，首先对本申请实施例中的一些名词进行定义，旨在使本申请实施例的技术方案更加清楚。

[0073] 定义1：空间任务。

[0074] 空间任务可以表示为s＝<s.r,s.l,s.e,s.c>，其中，s.r表示该空间任务s的发布时间，s.l表示s的执行地点，可以用二维空间的一个点(x,y)描述，s.e表示s的失效时间。每一个任务s都会被标记为某个类别s.c。

[0075] 为了简单起见，本申请可以进行如下假设：1)单任务分配模式，即服务器只分配一个任务给一个用户；2)每个任务的处理时间为0，这意味着一个用户可以在完成当前任务之后直接去下一个任务地点。但是，本申请实施例提供的空间众包任务分配方法，并不受限于上述假设。

[0076] 定义2：可获得用户。

[0077] 给定一系列时间实例T＝{t,t+1,...,t+n}(t是当前时刻)，一个可获得用户$w =< w.\emptyset, w.range, w.\mathbb{P} >$，其中$w.\emptyset = \{t+k, t+k+1, \ldots, t+g\}(\subset T)$表示用户w的可达时刻，w.range表示可达范围，$w.\mathbb{P} = \left\{w.l_1^{t+k}, w.l_2^{t+k+1}, \ldots, w.l_{|w.\emptyset|}^{t+g}\right\}$表示由一系列伴有时间戳的位置组成的w的旅行路线。

[0078] 定义3：任务序列。

[0079] 给定一个用户w和一系列分配给用户w的任务S_w，则S_w的任务系列$R(S_w) = (s_1, s_2, \ldots, s_{|s_w|})$表示用户w访问$S_w$中每个任务的顺序，用户w到达任务$s_i \in S_w$(即完成任务$s_i$的时间)可以按照如下公式计算：

[0080] $$t_{w,R}(s_i.l) = \begin{cases} t_{w,R}(s_{i-1}.l) + c(s_{i-1}.l, s_i.l), & i \neq 1 \\ t_{now} + c(w.l, s_1.l), & i = 1 \end{cases}$$

[0081] 其中，c(a,b)表示从位置a到位置b的旅行时间，t_{now}表示当前时间，w.l表示用户w开始接受任务分配的位置，当w和R的上下文清晰时，本申请实施例使用$t(s_i.l)$代替$t_{w,R}(s_i \cdot l)$。

[0082] 定义4:特定位置的有效任务集合。

[0083] 给定一个时间实例集合T和一个用户w的开始位置w.l,如果存在一个任务序列$R(S_w)=(s_1,s_2,\dots,s_{|s_w|})$满足以下三个条件,则任务集合$S_w$被称为用户w的一个特定位置的有效任务集合(Location-specific Valid Task Set,L-VTS)。对于$\forall s_i \in S_w$:

[0084] 1)$t(s_i.l)\leqslant s_i.e$,

[0085] 2)$t(s_i.l)\in w.\emptyset(\subset T)$,

[0086] 3)$d(w.l,s_i.l)\leqslant w.range$,其中d(a,b)是位置a和位置b的给定距离。

[0087] 定义5:特定路线的有效任务集合。

[0088] 给定一个时间实例集合T和一条用户w的路线$w.\mathbb{P}=\{w.l_1,w.l_2,\dots,w.l_{|w.\emptyset|}\}$,如果存在一个任务序列$R(S_w)=(s_1,s_2,\dots,s_{|s_w|})$满足以下三个条件,则任务集合$S_w$被称为用户w的一个特定路线的有效任务集合(Route-specific Valid Task Set,R-VTS)。对于$\forall s_i \in S_w$:

[0089] 1)$t(s_i.l)\leqslant s_i.e$,

[0090] 2)$t(s_i.l)\in w.\emptyset\ (\subset T)$,

[0091] 3)$D(s_i.l,w.\mathbb{P})\leq\frac{w.range}{2}$,其中D(a,b)是位置a和路线b的给定距离。

[0092] 定义6:特定位置/路线的有效任务集合。

[0093] 给定一个时间实例集合T,如果特定位置/路线的有效任务集合S_w的任何超集都对用户w无效,则为最大值。这叫做特定位置/路线的有效任务集合(Location/Route-specific Valid Task Set,L/R-MaxVTS)。

[0094] 定义7:空间任务分配。

[0095] 给定一系列的时间实例T＝{t,t+1,...,t+n}以及在时间T可获得的一系列用户和任务,空间任务分配用A表示,包括一组集合对＜worker,VTS>,集合对以<w_1,VTS(w_1)＞,＜w_2,VTS(w_2)＞,…形式存在。本申请使用A.S表示分配给所有用户的任务集合,即$A.S=\cup_{w\in W}S_w$,使用$\mathbb{A}$表示所有可能的分配方式。这个问题可以被如下陈述表示:

[0096] 问题定义:给定一系列时间实例T＝{t,t+1,...,t+n}(t是当前时刻),预测性任务分配(Predictive Task Assignment,PTA)问题旨在找到全局最优分配A_{opti}使得对于$\forall A_i\in\mathbb{A},|A_i\cdot S|\leqslant|A_{opti}\cdot S|$。

[0097] 本申请实施例提供的空间众包任务分配方法,在考虑用户和任务时,不仅考虑了当前时间实例,还考虑下一个连续时间实例。因此,眼前的挑战是准确估计时空维度下用户和任务的未来分布。为此,如图5(a)所示,本申请实施例提供的空间众包任务分配方法包括两个阶段:用户与任务的预测阶段以及任务调度与分配阶段。

[0098] 用户与任务的预测阶段旨在根据用户的任务执行轨迹历史和任务的释放历史预测用户/任务的未来时空分布。本申请实施例分别针对用户和任务预测提出不同的策略。具体而言,通过将用户的任务执行历史视为序列数据,本申请实施例利用序列模式挖掘(Sequential Pattern Mining,SPM)方法以挖掘频繁的时间实例(即她更可能完成任务的

时间)为可用时间。然后,本申请实施例为每个用户在其可用的时间提出两种空间分布预测策略:1)基于时空递归神经网络(Spatial-Temporal Recurrent Neural Networks,ST-RNN)的位置预测;2)基于混合模型的路线预测。对于任务预测,由于空间任务可以被视为空间点事件,因此,本申请实施例设计路线约束深度游走(PC-DeepWalk)方法以获得每个时间实例未来任务的数量,然后采用核密度估计(Kernel Density Estimation,KDE)方法预测未来时间实例的任务位置分布。

[0099] 任务调度与分配阶段通过为每个用户安排任务序列,将任务分配给合适的用户,以达到最大的任务分配。本申请实施例首先为每个用户计算所有的极大有效任务集(MaxVTSs),然后在枚举每个用户的MaxVTS的所有可能组合时,随后的任务分配必须解决巨大搜索空间中的计算问题。本申请实施例提出贪心任务分配(Greedy Task Assignment,GTA)算法试图从未分配的任务中为每个用户分配最大的MaxVTS,并且提出最优任务分配(Optimal Task Assignment,OTA)算法以获得全局最优分配。

[0100] 本申请实施例提供了一种空间众包任务分配方法,如图5(b)所示,包括以下步骤501～503:

[0101] 步骤501:根据用户的任务执行历史预测用户的未来时空分布。

[0102] 在一种示例性实施例中,步骤501包括:

[0103] 将用户的任务执行历史作为时间序列数据,利用序列模式挖掘方法从时间序列数据中挖掘出现频率超过最小支持阈值的时间实例集合,并将每个连续的时间实例集合作为用户的可用时间;

[0104] 在可用时间开始时,基于时空递归神经网络模型预测每个未来用户的空间分布。

[0105] 空间众包涉及物理世界的移动性,受一系列与位置相关因素的影响,其中,用户的行为模式在任务分配中起着重要作用。本申请实施例利用一种序列模式挖掘方法检测用户的可用时间。

[0106] 考虑到时间顺序排列的任务执行时间作为时间序列数据,本申请实施例使用序列模式挖掘方法从用户的任务执行历史中挖掘用户更可能执行任务的频繁时间实例,该方法已经在时间序列数据库中被广泛研究。将$T_k^w = \{t_1, t_2, \ldots, t_m\}$作为一系列时间顺序的时间实例,其中,用户w在第k天中执行任务,$T^w = \{T_1^w, T_2^w, \ldots, T_n^w\}$表示用户w的任务执行历史中n天中所有的时间实例。通过扫描T^w可以提取出现频率超过给定的最小支持阈值的时间实例集合并且每个连续的时间实例集合被称为用户w的可达时间或可用时间(即w.$\emptyset$)。

[0107] 一旦获得了每个用户的可用时间,本申请实施例提出两种策略预测未来用户的空间分布,一种策略是使用一个时空递归神经网络模型以查找在其可用的时间范围内倾向于执行任务的未来位置;另一种策略是使用一个混合模型(集成模式匹配策略和时空序列相关性)以查找在其可用的时间范围内倾向于执行任务的未来路线。

[0108] 在一种示例性实施例中,每个未来用户的空间分布包括每个未来用户的位置。

[0109] 时空递归神经网络模型包括输入层、隐藏层和输出层,输入层包含用户w在时间t_i到达位置的潜在向量$q_{l_{t_i}^w} \in \mathbb{R}^d$,隐藏层包含用户w在当前时间实例t的向量表示$h_{t,l_t^w}^w$,且

[0110] $$h_{t,l_t^w}^w = f\left(\sum_{l_{t_i}^w \in L^w, t-\hat{v}<t_i<t} S_{l_t^w - l_{t_i}^w} T_{t-t_i} q_{l_{t_i}^w} + Ch_{t-\hat{v},l_{t-\hat{v}}^w}^w\right)$$

[0111] 其中，W和L分别为一组潜在的用户集合及位置集合，$p_w \in \mathbb{R}^d$和$q_l \in \mathbb{R}^d$分别表示用户w和位置l的潜在向量，每个位置和坐标相关，每个用户w有一系列刚刚经过的历史位置的集合，即$L^w = \{l_{t_1}^w, l_{t_2}^w, \ldots\}$，v表示时间窗口的宽度，使用近似值$\hat{v}$作为局部窗口宽度，所述近似值$\hat{v}$最接近于v且$l_{t-\hat{v}}^w \in L^w$，$S_{l_t^w - l_{t_i}^w}$表示$l_t^w$和$l_{t_i}^w$之间地理距离的转移矩阵，$T_{t-t_i}$表示时间间隔$t-t_i$的转移矩阵，C是先前状态传播顺序信号的循环连接，激活函数f(x)使用Sigmod函数$f(x)=\exp(1/1+e^{-x})$。

[0112] 输出层包含用户w在时间t到达位置l的概率$O_{w,t,l}$，且

[0113] $$O_{w,t,l} = (h_{t,l}^w + p_w)^T q_l$$

[0114] 其中，$h_{t,l}^w$在特定的时空情境中捕捉动态兴趣，p_w是表明其兴趣和活动范围的永久表示。

[0115] 在可用时间开始时，基于时空递归神经网络模型预测每个未来用户的空间分布，包括：

[0116] 将时间间隔和地理距离分别划分到离散仓，学习对应的离散仓的上下边界的转移矩阵并且通过线性插值的方法计算其他时间间隔和地理距离的转移矩阵，以估计用户可用时间开始时的位置；

[0117] 使用贝叶斯个性化排名和基于时间的反向传播方法训练时空递归神经网络模型；

[0118] 将当前用户的历史位置数据输入时空递归神经网络模型；

[0119] 计算输出层中的位置表示和用户的内积，预测用户w在时间t最大概率出现的位置。

[0120] 本申请实施例假设每个用户有一个全球定位系统(Global Positioning System，GPS)设备(例如，具有GPS功能的移动电话)以跟踪用户的位置。

[0121] 由于时空递归神经网络模型在查找兴趣点之间的序列相关性上的成就，在用户倾向于执行任务的可用时间开始时，本申请实施例使用该方法预测每个未来用户的位置。在时空递归神经网络模型中，给定一组潜在的用户集合W以及位置集合L，$p_w \in \mathbb{R}^d$和$q_l \in \mathbb{R}^d$分别表示用户w和位置l的潜在向量。每个位置和它的坐标相关并且每个用户w有一系列其刚刚经过的历史位置的集合，即$L^w = \{l_{t_1}^w, l_{t_2}^w, \ldots\}$。时空递归神经网络模型的架构如图6所示，该模型由输入层、隐藏层和输出层组成，输入层包含用户w在时间t_i到达位置的潜在向量$q_{l_{t_i}^w} \in \mathbb{R}^d$，隐藏层是时空递归神经网络模型的关键部分，在此部分中，w在当前时间实例t的向量表示可以计算如下：

[0122] $$h^{w}_{t,l^{w}_{t}} = f\Big(\sum_{l^{w}_{t_i}\in L^{w},\, t-\hat{v}<t_i<t} S_{l^{w}_{t}-l^{w}_{t_i}} T_{t-t_i}\, q_{l^{w}_{t_i}} + Ch^{w}_{t-\hat{v},l^{w}_{t-\hat{v}}}\Big)$$

[0123] 其中，$h^{w}_{t,l^{w}_{t}}$表示用户w在时刻t的向量表示，v表示时间窗口的宽度，$S_{l^{w}_{t}-l^{w}_{t_i}}$表示$l^{w}_{t}$和$l^{w}_{t_i}$之间地理距离的空间距离的转移矩阵，$T_{t-t_i}$表示时间间隔$t-t_i$的转移矩阵，C是先前状态传播顺序信号的循环连接，激活函数f(x)使用Sigmod函数f(x)＝exp(1/1+e^{-x})。由于历史L^{w}中不存在位置l^{w}_{t-v}，因此本申请使用近似值$\hat{v}$(最接近于v)作为局部窗口宽度，以保证$l^{w}_{t-\hat{v}}$存在于历史中(即$l^{w}_{t-\hat{v}} \in L^{w}$)。通常情况下，历史数据$L^{w}$中可能不存在位置$l^{w}_{t-v}$，即用户w在时刻t-v下无空间位置记录，因此，当$l^{w}_{t-v}$不存在时，我们采用近似值$\hat{v}$(该值最接近于v)作为此处的时间窗宽度，以确保$l^{w}_{t-v}$存在于历史数据中，即$l^{w}_{t-\hat{v}} \in L^{w}$，因此，实际上，$\hat{v}$是稍高于或稍低于v。

[0124] 输出层包含用户w在时间t到达位置l的概率$O_{w,t,l}$，$O_{w,t,l}$可以计算如下：

[0125] $$O_{w,t,l} = (h^{w}_{t,l} + p_{w})^{T} q_{l}$$

[0126] 其中，$h^{w}_{t,l}$在特定的时空情境中捕捉动态兴趣，p_w是表明其兴趣和活动范围的永久表示。

[0127] 最终，通过计算输出层中的位置表示$O_{w,t,l}$(即用户w在时间t到达位置l的概率)和用户w的内积从而产生ST-RNN的预测(即找到用户w在时间t最大概率出现的位置)。

[0128] 为了精确估计用户可用时间开始时的位置，ST-RNN将时间间隔和地理距离分别划分到离散仓，学习对应离散仓的上下边界的转移矩阵并且通过线性插值的方法计算其他时间间隔/地理距离的转移矩阵。本申请使用贝叶斯个性化排名(Bayesian Personalized Ranking，BPR)和基于时间的反向传播(Back Propagation Through Time)两种方法学习参数(即S，T，C，p，q)。

[0129] 在另一种示例性实施例中，每个未来用户的空间分布包括每个未来用户的路线。

[0130] 由于大多数人的日常户外移动受到实际道路的约束，本申请实施例的目标是根据GPS对用户过去在公路网络中形成的观察预测所有用户的潜在路线。基于从历史轨迹数据中提取的路线模式以预测物体未来路线已被证明是一个有效的方法。但是，模式匹配方法(Pattern Matching Approach，PMA)存在稀疏问题，即可获得的历史轨迹远不能够覆盖所有可能的轨迹，这些轨迹可能不会返回任何的预测结果。为了解决这个问题，本申请实施例将上述时空递归神经网络模型应用到无模式匹配时的预测过程。通过这种方式，可以提高路线预测的精确度和鲁棒性。

[0131] 基于时空递归神经网络模型预测每个未来用户的空间分布，包括：

[0132] 使用基于特征点的道路拐角提取方法从历史轨迹中提取道路拐角；

[0133] 使用基于道路拐角的轨迹映射方法表示历史轨迹数据和将被预测的查询轨迹，生成以道路拐角为中心的路线；

[0134] 使用前缀投射顺序模式挖掘算法从历史轨迹中挖掘数量大于最小支持度阈值(该最小支持度阈值根据轨迹数据的情况由用户指定)的轨迹，作为满足最小支持度阈值的频

繁移动模式；

[0135] 基于被发现的频繁移动模式，使用模式匹配方法以预测未来路线；

[0136] 当无模式匹配时，使用时空递归神经网络模型预测下一个行驶中的道路拐角。

[0137] 图7描述了用户路线预测的混合模型的概述，该模型首先使用基于特征点的道路拐角提取(Characteristic Point-based Road Corner Extraction，CP-RCE)方法从大量的历史轨迹中提取道路拐角，然后使用基于道路拐角的轨迹映射方法表示历史轨迹数据和将被预测的查询轨迹以生成以道路拐角为中心的路线。在路线预测过程中，本申请实施例基于被发现的频繁移动模式使用模式匹配方法以预测未来路线。当遇到无模式匹配时，可以使用时空递归神经网络模型预测下一个行驶中的道路拐角。注意，本申请实施例仅仅使用用户的最新速度作为其未来速度以计算其在可用时间内能够行驶的距离。针对这个距离，基于混合预测模型的给定查询路线逐渐增长。

[0138] 在该实施例中，使用基于特征点的道路拐角提取方法从历史轨迹中提取道路拐角，包括：通过线性拟合的方法生成一组特征点，该组特征点为轨迹方向发生显著变化的坐标点，这里，显著变化的坐标点是指用户的坐标点和道路匹配后，用户从一条道路到另外一条道路经过的拐点；使用基于噪音的多密度空间聚类(Multiple Density Level Density-Based Spatial Clustering of Applications with Noise，MDL-DBSCAN)方法检测道路拐角；

[0139] 在该实施例中，使用模式匹配方法以预测未来路线，包括：执行模式匹配过程以找到待匹配轨迹，在待匹配轨迹中，前缀和以道路拐角为中心的路线匹配，未来路线使用最长最后匹配策略从查询轨迹中生成。最长最后匹配策略通过计算待匹配轨迹与频繁模式的匹配覆盖率，找到最长的轨迹模式作为待匹配轨迹对应的未来路线。

[0140] 道路拐角检测和轨迹预测。由于实际道路的拓扑信息(如道路拐角)被嵌入到个人GPS轨迹中，因此，本申请实施例从这些轨迹中检测道路拐角并且利用它们表示每个轨迹。特别的是，本申请实施例使用CP-RCE方法，其中通过线性拟合的方法生成一组特征点(即轨迹方向发生显著变化的坐标点)，然后使用MDL-DBSCAN方法检测道路拐角。以这种方式，可以检测到频繁出现在历史轨迹中的流行道路拐角，并且使用这些道路拐角可以提取出一系列以道路拐角为中心的路线从而抽象轨迹。

[0141] 模式匹配方法。在上一步中，每个轨迹都被转换成有序的道路拐角序列。随后，本申请实施例使用前缀投射顺序模式挖掘(prefix-projected sequential pattern mining，PrefixSpan)方法从历史轨迹中发现隐藏的移动频繁模式。PrefixSpan是一种递归方法，首先它查找频繁的前缀序列，然后将其投射到数据库中，查找频繁的后缀并将其和前缀连接，以获得频繁序列模式而生成候选序列。FP-tree是一种索引结果，该方法用于存储投射的数据库以提高效率。

[0142] 一旦得到频繁移动模式，本申请实施例执行模式匹配过程以找到候选模式，在该候选模式中，前缀可以和以道路拐角为中心的路线匹配，这些路线使用最长最后匹配策略(longest last matching strategy)从查询轨迹中生成。最长最后匹配策略专注于查询路由相对于要匹配的频繁模式的相对匹配覆盖率，以找到最长的模式作为预测路线。

[0143] 步骤502：将预设区域划分成多个不相交的网格，应用网络嵌入的方法预测未来时间戳中每个网格的任务数量，根据各个网格预测的任务数量，预测任务的位置。

[0144] 为了达到一个更好的全局任务分配，理解将来要发布的任务是何地何时何种类型是至关重要的。由于空间任务必须在特定位置进行完成，本申请实施例将任务视为空间点事件。平面核密度估计(KDE)方法已经被广泛应用于空间点时间分析和检测，通过计算事件强度作为密度估计，从而在空间上生成点事件的平滑密度表面。

[0145] 本申请实施例使用平面核密度估计方法计算空间任务的密度，通过将研究区域划分成不相交且均匀的网格(例如，20×20)以预测其位置。在预测任务位置之前，本申请实施例需要估计潜在不同类型任务的数量，这些任务在将来的时间实例中可以会落入每个网格中。而不是仅仅考虑每个网格中任务数量的时间相关性，本申请实施例构建了一个基于空间任务的网络，并且应用网络嵌入的方法通过考虑未来时间实例中每个网格的时空关系以预测任务数量。

[0146] 本申请实施例利用历史数据预测未来时间戳中每个网格的任务数量。首先，基于任务的时空信息构建一个网络，即基于空间任务的网络G＝(V,E)，如图8所示，在该网络图中V包含两种类型的节点(即，基于时间单元的节点和基于空间任务的节点)，E包含两种类型的边(即，空间邻近边和任务相关边)。显然，基于空间任务的网络节点和边属于多种类型，所以该网络是一个异构信息网络(Heterogeneous Information Network，HIN)。节点和边如下定义：

[0147] 定义8：基于时间单元(Temporal Cell-based，TC)的节点。

[0148] 基于时间单元的节点$C_i^{t_j}$表示在时间戳t_j处网格的特定空间单元C_i。

[0149] 定义9：基于空间任务(Spatial Task-based，ST)的节点。

[0150] 基于空间任务的节点B_m表示特定任务类别m的节点。

[0151] 定义10：空间邻近边。

[0152] 空间邻近边$e\left(C_i^{t_j}, C_k^{t_g}\right)$连接两个TC节点，表示TC节点的空间邻近关系(Spatial Proximity Relation，SPR)。边的权重$\mathcal{W}\left(C_i^{t_j}, C_k^{t_g}\right)$和$C_i^{t_j}$、$C_k^{t_g}$两个的空间距离具有负相关性。

[0153] 定义11：任务相关边。

[0154] 任务相关边$e\left(C_i^{t_j}, B_m\right)$连接一个TC节点和一个ST节点，表示任务相关关系(Task Relevance Relation，TRR)，即有特定类别m的任务在时间戳t_j发布在空间单元C_i中。边的权重$\mathcal{W}\left(C_i^{t_j}, B_m\right)$表示在时间戳$t_j$发布在空间单元$C_i$中的带有特定任务类别m的任务数量。

[0155] 在一种示例性实施例中，应用网络嵌入的方法预测未来时间戳中每个网格的任务数量，包括：

[0156] 基于任务的时空信息建立基于空间任务的网络，所述网络的节点包括基于时间单元的节点$C_i^{t_j}$和基于空间任务的节点B_m，所述网络的边包括空间邻近边$e\left(C_i^{t_j}, C_k^{t_g}\right)$和任务相关边$e\left(C_i^{t_j}, B_m\right)$，所述基于时间单元的节点$C_i^{t_j}$表示在时间戳$t_j$处网格的特定空间单元

C_i,所述基于空间任务的节点B_m表示特定任务类别m的节点,所述空间邻近边$e\left(C_i^{t_j}, C_k^{t_g}\right)$表示两个基于时间单元的节点的空间邻近关系,任务相关边$e\left(C_i^{t_j}, B_m\right)$表示有特定类别m的任务在时间戳$t_j$发布在空间单元$C_i$中;

[0157] 建立空间路径以捕捉空间信息,建立任务相关路径以捕捉与任务相关的信息,建立时间排序的任务相关路径以捕捉时间信息,空间路径包括基于时间单元的节点和空间邻近节点,任务相关路径包括基于时间单元的节点、基于空间任务的节点和任务相关边,在时间排序的任务相关路径中,基于时间单元的节点按照一个时间单位的增长速率根据时间以升序排序;

[0158] 沿着上述三个路径使用λ长度的随机游走方法,以基于空间任务的网络G作为输入,为每个节点输出一系列长度为λ的随机游走序列;

[0159] 在得到每个节点的随机游走序列后,通过SkipGram神经网络模型学习每个节点的向量表示,基于历史数据和其他节点数据,使用回归算法估计每个网格中的任务数量。

[0160] 为了将每个节点编码为低维向量并且维护结构信息,本申请在图8中应用网络嵌入的方法。深度游走(Deep Walk)是最近提出方法,该方法从网络中的截断随机游走学习节点的隐式表示。深度游走将基于随机游走的邻近度和SkipGram模型相组合,该模型可以最大程度的提高出现在一个窗口中单词之间的共现概率。但是由于基于随机游走的邻近度没有考虑HIN的异质性,因此,深度游走在应用到本申请时,存在缺陷。受到HIN中基于元路径邻近模型的启发,其中,元路径是节点类别的序列,在对特定关系进行建模之间具有边类型,本申请实施例基于空间任务的网络,设计三种类别的路径以捕捉空间信息、与任务相关的信息和时间信息。然后,本申请实施例提出一个路径约束的随机游走(PC-DeepWalk)算法,将网络嵌入到低维空间中,使得网络中的原始节点表示为向量。三种类别的路径设计如下:

[0161] 1)空间路径以$C_1^{t_1} \xrightarrow{SPR} C_2^{t_2} \xrightarrow{SPR} \dots \xrightarrow{SPR} C_i^{t_i} \dots$形式存在,其中只包含TC节点和空间邻近节点。

[0162] 2)任务相关路径以$C_1^{t_1} \xrightarrow{TRR} B_1 \xrightarrow{TRR} C_2^{t_2} \xrightarrow{TRR} B_2 \xrightarrow{TRR} \dots \xrightarrow{TRR} C_i^{t_i} \xrightarrow{TRR} B_i \dots$形式存在,其中包含TC节点、ST节点和任务相关边。

[0163] 3)时间排序的任务相关路径是任务相关路径的特例,以捕捉任务释放模式的事件趋势,其中TC节点按照一个时间单位的增长速率根据时间以升序排序,即$C_1^{t} \xrightarrow{TRR} B_1 \xrightarrow{TRR} C_2^{t+1} \xrightarrow{TRR} B_2 \xrightarrow{TRR} \dots \xrightarrow{TRR} C_i^{t+i-1} \xrightarrow{TRR} B_i \dots$。

[0164] 然后,本申请实施例沿着提出的路径使用λ长度的随机游走方法,将基于空间任务的网络G作为输入并且将每个节点多个λ长度的随机游走序列作为输出。获得每个节点的随机游走序列后,本申请实施例使用跳字(SkipGram)神经网络模型学习每个节点的表示向量。SkipGram神经网络模型是Word2vec神经网络模型中的一种,Word2vec神经网络模型在给定无标签的语料库的情况下,为语料库中的单词产生一个能表达语义的向量。然后,本申请实施例通过考虑历史数据和其他节点数据,使用回归算法估计每个网格中的任务数量,

可由如下公式计算：

[0165] $$N_i^{t_{j+1}} = \alpha N_i^{t_j} + (1-\alpha) \sum_{C_{i'}^{t_j} \in \mathbb{C}^{t_j}, i' \neq i} \frac{\mathrm{sim}\left(C_i^{t_j}, C_{i'}^{t_j}\right)}{\sum_{C_{i'}^{t_j} \in \mathbb{C}^{t_j}, i' \neq i} \mathrm{sim}\left(C_i^{t_j}, C_{i'}^{t_j}\right)} N_{i'}^{t_j} + \beta$$

[0166] 其中，$N_i^{t_j}$表示节点$C_i^{t_j}$的任务数量，$\mathbb{C}^{t_j}$表示在时间t_j处的所有节点，$\mathrm{sim}\left(C_i^{t_j}, C_{i'}^{t_j}\right)$表示节点$C_i^{t_j}$和$C_{i'}^{t_j}$之间的相关性，此相关性可由两者向量的点积计算。因此$\sum_{C_{i'}^{t_j} \in \mathbb{C}^{t_j}, i' \neq i} \frac{\mathrm{sim}\left(C_i^{t_j}, C_{i'}^{t_j}\right)}{\sum_{C_{i'}^{t_j} \in \mathbb{C}^{t_j}, i' \neq i} \mathrm{sim}\left(C_i^{t_j}, C_{i'}^{t_j}\right)} N_{i'}^{t_j}$表示两个节点之间的传播效果。A是控制历史数据和其他节点贡献的第一参数，β是避免过拟合的第二参数。为了获得和的最优值，本申请实施例使用随机梯度下降(Stochastic Gradient Descent)进行训练并且使用均方误差(Mean Squared Error)作为损失函数。

[0167] 获得网格$C_i^{t_{j+1}}$中的任务数量$N_i^{t_{j+1}}$后，本申请实施例在网格中任务样本的位置使用KDE方法计算潜在任务的共现概率，其中，任务样本$\{s_1, s_2, \ldots, s_n\}$由此网格中产生的一系列历史任务组成。在网格$C_i^{t_{j+1}}$中，潜在任务s在位置$l \in L_{C_i^{t_{j+1}}}$被释放的概率可以计算如下：

[0168] $$f(l) = \frac{1}{\left|L_{C_i^{t_{j+1}}}\right| H^2} \sum_{s_i.\hat{l} \in L_{C_i^{t_{j+1}}}} K\left(\frac{\hat{l} - s_i.\hat{l}}{H}\right)$$

[0169] 其中，$L_{C_i^{t_{j+1}}}$是网格$C_i^{t_{j+1}}$中一系列历史任务样本的位置，H是带宽，每个任务位置$s_i.l \in L_{C_i^{t_{j+1}}}$(纬度lat和经度lon)由$s_i.\hat{l} = (\mathrm{lat}, \mathrm{lon})^T$表示，函数K(•)是高斯核函数，其中$K(X) = \frac{1}{2\pi} \exp\left(-\frac{1}{2} X^T X\right)$。最优带宽H可以计算如下：

[0170] $$H = \frac{1}{2} \left|L_{C_i^{t_{j+1}}}\right|^{-\frac{2}{3}} \sqrt{\hat{h}}^T \sqrt{\hat{h}}$$

[0171] 其中，$\hat{h}\left(= \frac{1}{\left|L_{C_i^{t_{j+1}}}\right|} \sum_{s_i.\hat{l} \in L_{C_i^{t_{j+1}}}} (s_i.\hat{l} - \bar{C}_i^{t_{j+1}})^2\right)$是网格$C_i^{t_{j+1}}$中所有历史任务位置的方差，$\bar{C}_i^{t_{j+1}}\left(= \frac{1}{\left|L_{C_i^{t_{j+1}}}\right|} \sum_{s_i.\hat{l} \in L_{C_i^{t_{j+1}}}} s_i.\hat{l}\right)$表示这些位置的平均值，根据网格$C_i^{t_{j+1}}$中被预测的任务数量$N_i^{t_{j+1}}$，设置前$N_i^{t_{j+1}}$个位置(有高概率)作为被预测任务的位置。

[0172] 步骤503:根据预测的用户的未来时空分布以及任务的位置,进行任务分配。

[0173] 基于当前和被预测的用户和任务,本步骤提出两种任务分配算法以解决提出的PTA问题,最终目的实现最大任务分配。基本思想是尝试找到一个所有用户可能的有效任务序列的并集以使得本分配任务的数量最大化。在接下来的篇幅中,首先介绍极大有效任务集(MaxVTS)生成算法,该集合包含特定位置和特定路线的MaxVTS,MaxVTS将会在整个算法中使用。本申请实施例提出一种贪心算法,该算法从未分配的任务为每个用户迭代的找到一个“最佳”MaxVTS,直到所有的任务都被分配或者所有的用户被用完。本申请实施例还提出了一种基于图划分的分解算法进行任务分配,该算法可以为所有用户找到最佳MaxVTSs的并集以实现最优任务分配。

[0174] 在一种示例性实施例中,步骤503包括:

[0175] 基于用户可达距离和任务失效时间的约束,建立在给定时间段中每个用户能够执行的可达任务集合;

[0176] 获得每个用户的可达任务集合之后,采用动态规划算法计算每个用户的所有极大有效任务集的集合,动态规划算法主要是通过迭代地扩展任务集合来为每个用户寻找其所有的极大有效任务集的集合。,极大有效任务集的计算公式如下:

[0177] $$\mathrm{opt}(Q,s_j)=\begin{cases}1, & |Q|=1\\ \max\limits_{s_i\in Q,s_i\neq s_j}\mathrm{opt}(Q-\{s_j\},s_i)+\delta_{ij}, & \text{其他}\end{cases}$$

[0178] $$\delta_{ij}=\begin{cases}1, & t(s_j.l)\le s_j.e, t(s_j.l)\le t+n\\ 0, & \text{其他}\end{cases}$$

[0179] 其中,opt(Q,s_j)为在用户的可达范围内通过调度Q中的所有任务最终被完成的最大任务数量,δ_{ij}表示在给定时间段T={t,t+1,...,t+n}内在将s_j附加在R′后是否可以完成s_j,给定一个用户w和一个任务集合$Q\subseteq RS_w$,(Q,s_j)表示用户w从自己的位置w.l出发,执行任务集合Q中的一系列任务,且最后一个完成的任务是s_j,opt(Q,s_j)表示(Q,s_j)能完成的最大任务数量,R是实现这一最大任务数量的有序任务序列。s_i表示R中到达任务s_j的前一个任务(即任务序列R的倒数第二个任务),用R′表示opt(Q-{s_j},s_i)对应的任务序列。

[0180] 根据每个用户的所有极大有效任务集的集合,进行任务分配。

[0181] 本申请实施例中,每个用户的所有的极大有效任务集的集合的产生,包括以下步骤:

[0182] 1)查找可达任务:由于用户可达距离和任务失效时间的约束,每个用户只能在给定的时间实例T={t,t+1,...,t+n}集合中完成一部分任务。因此本申请实施例首先找到在给定时间段T中每个用户能够执行的任务集合。用户w的特定位置可达任务子集(location-specific reachable task subset,L-RS_w)应该满足以下条件:$\forall s\in L\text{-}RS_w$:

[0183] i)c(w.l,s.l)≤s.e-s.p;

[0184] ii)c(w.l,s.l)≤n+1;

[0185] iii)d(s_i.l,s_j.l)≤w.range;其中c(w.l,s.l)表示从w.l到s.l的旅行时间,d(a,b)表示位置a和b之间的给定距离。

[0186] 至于用户w的特定路线可达任务子集(route-specific reachable task subset,

R-RS_w)应该满足以下条件：$\forall s \in R\text{-}RS_w$:

[0187] i)c(w.l,s.l)≤s.e-s.p,

[0188] ii)c(w.l,s.l)≤n+1,

[0189] iii) $D(s_j.l, \mathbb{P}_w) \leq \frac{w.range}{2}$,其中D(a,b)表示位置a和路线b之间的给定距离。

[0190] 上述条件保证了用户可以在给定时间段T内在任务s失效之前从其起点到达任务s的位置,其中,任务s位于其特定位置/路线距离范围内。

[0191] 2)查找极大有效任务集:给定每个用户的可达任务集合,接下来本申请实施例查找MaxVTS的集合。本申请实施例提出动态规划的算法按照集合大小的升序迭代扩展任务集合并且在每次迭代中找到每个用户的MaxVTSs。对于一个集合的每个任务,本申请实施例考虑最终完成任务的场景并且找到所有完成的任务序列。具体来说,给定一个用户w和一系列任务$Q \subseteq RS_w$,本申请实施例定义opt(Q,s_j),其中,Q中的任务从w.l到s_j.l,为在用户的可达范围内通过调度Q中的所有任务最终被完成的最大任务数量。R表示Q上的相应任务序列以获得最优值。本申请实施例还使用表示在到达之前中倒数第二个任务,使用R′表示opt(Q-{s_j},s_i))中相关的任务序列。然后opt(Q,s_j)可以被计算如下：

[0192] $$opt(Q, s_j) = \begin{cases} 1, & |Q| = 1 \\ \max\limits_{s_i \in Q, s_i \neq s_j} opt(Q - \{s_j\}, s_i) + \delta_{ij}, & \text{其他} \end{cases}$$

[0193] $$\delta_{ij} = \begin{cases} 1, & t(s_j.l) \leq s_j.e, t(s_j.l) \leq t + n \\ 0, & \text{其他} \end{cases}$$

[0194] δ_{ij}表示在给定时间段T＝{t,t+1,...,t+n}内在将s_j附加在R′后可以完成s_j。基于上面的公式本申请实施例可以获得每个用户的所有MaxVTSs。

[0195] 当Q只包含任务s_i时,问题很简单并且opt({s_i},s_i)设为1,当|Q|＞1,本申请实施例需要遍历搜索Q以检查有效任务集合的有效性并且找到获得opt(Q,s_j)最优值的特定的s_i。在图1和图3中,举例如下,用户w_1有4个可达任务{s_1,s_2s_3,s_4},计算用户w_2的L-MaxVTSs,本申请实施例的算法首先计算opt({s_1},s_1)＝1,opt({s_2},s_2)＝1,opt({s_3},s_3)＝1,opt({s_4},s_4)＝1,对于所有的大小从2到4的集合,本申请实施例迭代的计算opt值和相关的R。例如,遵循任务序列(s_1,s_2)只能完成s_1,所以opt({s_1},s_2)＝1,但是遵循(s_2,s_1),s_1和s_2都可以被完成,所以opt({s_1},s_1)＝2。同理,亦可以求出R-MaxVTSs。

[0196] 在一种示例性实施例中,根据每个用户的所有极大有效任务集的集合,进行任务分配,包括：

[0197] 将用户集合W和任务集合S作为输入,初始化一个空的任务分配；

[0198] 开始迭代,在每次迭代中,从剩下的将分配的用户中随机选择一个用户并且从未分配的任务中为选择的用户找到最大的极大有效任务集,并将找到的最大的极大有效任务集加入到当前任务分配中；

[0199] 迭代结束,在所有迭代中找到最大任务分配。

[0200] 本实施例中,获得每个用户的MaxVTSs后,一个直接的方法是为每个用户从没有分配的任务中分配最大有效集合,直到所有的任务被分配、用户被分完,由于这个方法没有考

虑分配任务的最佳策略,因此该方法被称为贪心任务分配(GTA)。

[0201] 下面的算法描述GTA的简要过程,将用户集合W和任务集合S作为输入,在算法的第1行中初始化一个空的任务分配,在每次迭代中,GTA开始从剩下的将分配的用户中随机选择一个用户w∈W并且从未分配的任务中为选定的用户找到最大MaxVTS,其中最大MaxVTS被加入到当前任务分配中(算法的第2-6行),最后本申请实施例在所有迭代中找到最大任务分配(算法的第7行)。

[0202] Input:W,S

[0203] Output:可行的任务分配结果A和相应的被分配任务的数量|A|

[0204] 第1行A←Φ

[0205] 第2行循环:对W中的每个用户w:

[0206] 第3行Q_w←max{MaxVTS(w,S)};

[0207] 第4行A=A∪Q_w

[0208] 第5行S=S-Q_w

[0209] 第6行W=W-w

[0210] 第7行返回A和|A|

[0211] 在另一种示例性实施例中,根据每个用户的所有极大有效任务集的集合,进行任务分配,包括:

[0212] 对于用户集合和任务集合,根据用户之间的依赖关系构造用户依赖图,所述依赖关系包括互相独立或互相依赖,当两个用户没有共同的可达任务时,两个用户互相独立;当两个用户有共同的可达任务时,两个用户互相依赖;

[0213] 将用户依赖图划分成多个节点集,以分解用户的依赖关系;

[0214] 根据划分的多个节点集,建立平衡树,所述平衡树中的兄弟节点互相独立;

[0215] 通过深度优先搜索算法为树节点中的每个用户计算有效任务集合,以找到最优分配。

[0216] 在本实施例中,在枚举每个用户的有效任务集合的所有可能组合时,主要的计算挑战在于巨大的搜索空间,这随着用户数量呈指数增长。但是,实际上一个用户只与少数几个具有相似或相交的旅行路线的用户共享相同的任务。本申请实施例首先构造一个用户依赖图。对依赖图采用图划分方法并且将每个子图的用户集合组织成树结构,因此该问题被分解成多个独立子问题。然后设计了深度优先搜索算法查找最优任务分配。

[0217] 1)用户依赖图构造:如果两个用户没有共同的可达任务则称为相互独立,反之则称为互相依赖,基于用户之间的依赖/非依赖关系构造用户依赖图(Worker Dependency Graph,WDG),特别的是,给定一个用户集合W和一个任务集合,本申请实施例使用WDG,即G(V,E),编码用户之间的依赖关系,其中每个节点v∈V表示一个用户w_v∈W,如果两个用户w_v和w_u相互依赖则u和v之间存在边e(u,v)∈E。

[0218] 2)图划分:最后,本申请实施例采用基于k度图简约(Graph Reduction-based,GR)的方法通过划分WDG图分解用户的依赖关系。图划分的结果包含一系列节点X={X_1,...,X_n},应该满足以下条件:

[0219] i)∪$_{i∈n}X_i$=V,

[0220] ii)$\forall(u,v)\in E,\exists X_i\in X$,其中$X_i$包含u和v,

[0221] iii)如果X_i,X_j和X_k是节点且X_k是从X_i到X_j的路线,则$X_i \cap X_j \subseteq X_k$。

[0222] k度图简约方法是通过删除度小于k的节点(节点的度是指和该节点相关联的边数),将图简化成另一个节点减少的简单图,以此来找到节点集的集合X,具体步骤如下:

[0223] 1)对于给定的用户依赖图G,我们按照节点度数的升序对G中的节点进行简约处理,即重复步骤2)和3);

[0224] 2)对于指定的度i(0≤i≤k),首先找到度为i的节点v,并检查节点v的所有邻居是否构成一个团(clique,是指一个两两之间有边的节点集合),若无法构成一个团,那么通过添加边来构造团;

[0225] 3)节点v和它的邻居节点构成一个团,并存储于一个堆栈中,之后,在原图中删除v及其对应的边;

[0226] 4)重复上述步骤2)和3),直至满足下列条件之一:(1)原图被简约成一个简单图(示例性的,所述简单图可以为三角形)或空图(其节点数量为0);(2)不存在度小于或等于k的节点。

[0227] 5)堆栈中的所有团和未被上述简约过程删除的团便构成图分割的节点集。

[0228] 本申请实施例从删除度为0的节点(即孤立节点)开始执行上述节点删除过程,然后按照节点度的升序处理它,上述过程一旦满足以下条件之一将会停止:i)图被简约为一个简单图(即单个三角形)或空集;ii)不存在度小于或等于k的节点。图9描述了给定一个WDG的简约过程,该过程从删除节点w_1及其边开始度为2的简约处理。节点w_1及其邻居节点被放入堆栈中。随后,与w_1相同的处理原则,分别去除节点w_2,w_3,w_5。最终,原图被简约成一个简单的三角形,可以找到图的团并且作为图划分的节点集输出:X＝{{w_1,w_2,w_3},{w_2,w_3,w_4},{w_3,w_4,w_5},{w_4,w_5,w_7},{w_4,w_6,w_7}},如图10所示。

[0229] 3)树构建:根据图划分的属性,如果两个节点集没有共享相同的节点,则属于两个节点集的用户彼此独立。在这一步骤中,本申请实施例的目标是将用户的节点集组织成树结构的形式,以此使得兄弟节点之间互相独立,本申请实施例可以独立的解决每个兄弟节点上的最优分配子问题。本申请实施例通过以下的递归树构建(Recursive Tree Construction,RTC)算法建立平衡树:

[0230] i)从用户依赖图G中删除每个节点集Xi∈X的节点,将G划分为多个子图,其中最大的子图用G_{max}表示。

[0231] ii)选择能生成最小|Gmax|值(|Gmax|是图Gmax的节点数量)的节点集,记为Xmin,当存在多个能生成|Gmax|值的节点集时,选择最小的Xi作为Xmin;然后,将Xmin设置为树的根节点。

[0232] iii)对剩余的子图继续采用k度图简约算法,并在k度图简约算法的结果中递归地执行RTC算法。

[0233] 以图10为例,由于删除节点集X3后,最大子图的节点数是2,小于删除其它节点集所得到的|Gmax|值,因此,我们设置Xmin＝X3,在用户依赖图中删除X3,并将X3作为树的根节点(即剩余其它子图的父亲节点)。接着,我们对剩余的子图继续采用k度图简约算法和RTC算法,得到最终的树结构,如图11所示。将用户依赖关系图转化为树结构之后,我们利用深度优先搜索算法为每个用户计算最合适的有效任务集,从而得到最优任务分配结果。

[0234] 使用RTC算法,最终的树结构在图11中显示。将用户依赖图转化为树结构后,可以

使用深度优先搜索过程为树节点中的每个用户计算合适的有效任务集合,以找到最优分配。

[0235] 本申请实施例为空间众包提出了一种基于数据驱动的预测性空间任务分配(DPSTA)框架,其目的是当用户和任务在给定的时间段内动态出现时,优化全局任务分配。本申请实施例提出了两种新颖的策略,分别根据旅行历史预测用户的未来位置和路线。本申请实施例还设计了一种有效的图嵌入机制以估计任务的时空分布。本申请实施例提出了贪心和最优的两种任务分配算法,以权衡分配的效率和有效性。本申请实施例在真实数据集上进行了广泛的实验,结果证明本申请实施例的空间众包任务分配方法可以有效地解决实时任务分配。

[0236] 本申请实施例使用两个真实的数据集进行实验,TF(Twitter-Foursquare)数据集和GM(gMission)数据集,其中TF数据集包含类别信息的签到数据,GM是一个基于研究的通用空间众包平台。对于TF数据集,伴有地理标记的签到数据可用于模拟本申请实施例的问题,其中签到数据集收集了Twitter上2010年9月至2011年1月的数据。由于原始的Twitter数据集不包含地点的类别信息,本申请实施例借助其API从Foursquare中提取与每个地点相关联的类别信息。最终产生的数据集提供了纽约纬度40.231°到41.231°及经度-74.435°到-73.435°的签到数据,其中包含了2056位用户的29046的签到数据。GM数据集中包括532个用户和713个任务,其中,每个用户包含位置、到达时间、下线时间,任务包含位置、发布时间、失效时间以及用于分类任务的任务描述。

[0237] 实验设置

[0238] 当使用TF数据集时,本申请实施例假设用户是空间众包系统中接收和处理任务的人,因为在不同地点签到的用户可能是这些地点附近执行空间任务的最佳候选者,并且他们的位置最接近这些签到地点。对于每个签到地点,本申请实施例分别使用其位置和一天中最早的签到时间作为任务的位置和发布时间。因此签到的类别视为任务的种类并且在此地点签到等同于接受此任务。当本申请实施例使用GM数据集时,由于gMission缺少用户/任务的历史数据,因此本申请实施例按照如下方式生成了在历史记录(例如最近一个月)中加入系统的用户/任务。对于每个用户/任务,本申请实施例将其位置设置为中心,并以高斯分布随机产生其历史位置,该位置的发生时间以均匀分布的方式散布在每天中。

[0239] 对于这两个数据集,本申请实施例通过以下方式模拟每天每个用户的轨迹。将一个用户的日常位置输入道路交通仿真(Simulation of Urban Mobility,SUMO)以概率性方式产生GPS轨迹。而且本申请实施例设置时间片段的粒度为一小时(例如9:00am-10:00am),在此期间任务请求和可用用户将被打包入本申请实施例的框架中。本申请实施例在实验中的6个时间实例(由当前时间实例和未来的5个时间实例组成)中将任务分配给合适的用户。表1显示了本申请实施例实验的设置,其中所有参数的默认值标有下划线。所有的算法均在Intel Core i5-2400 CPU@3.10GHZ 8GB内存的机器上实现。

[0240]

参数	值
历史数据大小$\|\mathcal{T}\|$（即训练位置数据的百分比）	20%，40%，60%，80%，100%
任务数量|S|（TF）	1K，2K，3K，4K，5K
任务数量|S|（GM）	300,400,500,600,700
任务的有效时间 e-p	1，2，3，4，5
用户的可达半径 range	2km，2.5km，3km，3.5km，4km

[0241] 表1.实验参数

[0242] 实验结果

[0243] 用户预测性能

[0244] 本申请实施例评估用户预测阶段的性能及其对后续任务分配的影响。本申请实施例选取70%的用户/任务的位置数据作为训练,20%作为测试,剩下的10%作为验证集。

[0245] 为了预测用户位置,将两种代表性方法和ST-RNN进行比较:1)RNN:仅仅根据用户的行为顺序估计具有时间依赖性的未来位置;2)GWP:基于网格的用户预测(Grid-based Worker Prediction)。为了衡量每个模型的性能,本申请实施例提出准确率$\mathrm{acc(l)}=\frac{\left|\tilde{l}|d(\tilde{l},l)\leq\varepsilon,l\in L^{W}_{\{t,\ldots,t+n\}}\right|}{\left|L^{W}_{\{t,\ldots,t+n\}}\right|}$作为评价指标,其中$\tilde{l}$表示真实位置l的预测位置,d(a,b)表示点a和b之间的欧几里得距离,ε是空间偏差阈值(设为0.5km),$L^{W}_{\{t,\ldots,t+n\}}$表示在{t,...,t+n}所有用户出现的位置。如果$d(\tilde{l},l)\leq\varepsilon$则表示l被准确预测。注意,所有方法的时间窗口宽度被设为4。

[0246] 为了基于上述预测算法的任务分配的有效性,本申请实施例通过采用最优任务分配(OTA)算法来比较已存在或者这却预测的实际分配任务的数量。本申请实施例在TF和GM数据集上进行所有的实验。

[0247] $|\mathcal{T}|$的影响。在第一组实验中,本申请实施例改变训练集$|\mathcal{T}|$的大小并且研究其对用户位置预测的影响。从图12(a)可知,当使用更多的训练轨迹所有方法的准确度都会提高。在这些方法中,在两种数据集中,ST-RNN的准确度最高,其次是RNN和GWP。从图12(b)可知,任务分配的结果很大程度上取决于预测准确度,因为一个更高的准确度通常意味着更多被正确预测的用户。对于所有的$|\mathcal{T}|$值,这些方法中ST-RNN性能最高,验证了本申请实施例提出的算法的最优性。

[0248] 对于路线预测评估,本申请实施例引入另一个准确率$\mathrm{acc}(\mathbb{P})$,即被正确预测的道路连接数和总的道路连接数的比值。然后本申请实施例通过改变训练集的大小,将基线方法,即模式匹配方法(PMA)和本申请实施例的混合模型(标记为HYBRID)相比较。

[0249] 如预期那样,随着$|\mathcal{T}|$的增加两种算法的准确率也逐渐增加(如图13(a))。本申请实施例的混合模型比PMA在准确率方面有着显著的提高,表明随着$|\mathcal{T}|$的增加带来更多益处。在任务分配方面,图13(b)显示任务分配结果受预测准确度的影响。不管两个数据集中的$|\mathcal{T}|$大小,低准确率($\mathrm{acc}(\mathbb{P})\leq 0.24$)的PMA分配的任务比HYBRID少。

[0250] 任务预测性能

[0251] 本申请实施例引入两个竞争对手:DeepWalk和基于网格的任务预测(Grid-based Task Prediction,GTP),并且使用acc(l)评价任务的位置预测准确率。同时,本申请实施例使用分配任务的数量(由OTA生成)通过改变$|\mathcal{T}|$以衡量任务分配的有效性。

[0252] $|\mathcal{T}|$的影响。如图14(a)所示,当涉及更多的训练数据时,所有的预测方法的准确率都呈上升状态。和其他两个基线方法相比,PC-DeepWalk可以改善位置预测的准确率并且生成最准确的位置,从而如图14(b)证实,使得分配任务数量也越来越多。

[0253] 任务分配性能

[0254] 本申请实施例根据分配的任务总数和CPU时间评估任务分配的有效性和效率,具体来说,分配的任务数量可以衡量任务分配策略的质量并且CPU时间是由每个时间实例执行任务分配的平均时间成本。本申请实施例基于用户的位置/路线预测和任务预测评价提出的贪心任务分配(GTA)和最优任务分配(OTA):特定位置的GTA(Location-specific GTA,L-GTA)、特定位置的OTA(Location-specific OTA,L-OTA)、特定路线的GTA(Route-specific GTA,R-GTA)、特定路线的OTA(Route-specific OTA,R-OTA)。本申请实施例引入最大任务分配(Maximum Task Assignment,MTA)算法作为基线方法,该方法分别在当前和未来时间实例中在没有预测的情况下进行任务分配。而且本申请实施例还实现了一种基于预测的代表性任务分配算法,即具有基于网格的用户/任务预测的基于网格的预测性任务分配(Grid-based Predictive Task Assignment,GPTA)。在GPTA中,将分配用户执行任务的质量得分设置为1以便达到最大分配任务数。

[0255] |S|的影响。首先本申请实施例调查任务数量如何影响任务分配的有效性和效率。正如预期那样,如图15(a)和15(c)所示,在TF和GM数据集中,随着|S|的增长所有算法的分配任务数量逐渐增多。MTA产生最小的任务分配数量,而R-OTA产生最大的任务集,其次是R-GTA、L-OTA、L-GTA和GPTA。毫不奇怪,R-OTA和L-OTA产生的任务分配数量比各自的使用贪心任务分配策略的竞争对手(即R-GTA和L-GTA)多。特定路线的任务分配算法(即R-OTA和R-GTA)分配的任务数量比特定位置的方法(即L-OTA和L-GTA)多,这源于沿着特定路线执行任务时用户有更大的可达范围。就运行时间而言,如图15(b)和15(d)所示,MTA算法运行最快,几乎不受|S|的影响。而R-OTA最耗时间,R-OTA(L-OTA)运行比R-GTA(L-GTA)慢主要是因为构建要搜索的树需要花费额外的时间。虽然GPTA比本申请实施例提出的方法效率高,但是它分配的任务数量少。

[0256] e-p的影响。接下来本申请实施例研究任务有效时间e-p的影响。正如图16(a)和16(c)所示,当任务的有效时间增加时所有方法的被分配的任务数量都随之增加,这背后的原因是,用户有更多的机会在更宽松的有效时间内被分配任务。与之前的结果相似,本申请实施例提出的特定路线的任务分配方法比特定位置的任务分配方法可以实现更多的分配任务,并且两者都比GPTA和MTA的效果好,这证实了本申请实施例提出的算法的优越性。从图16(b)和图16(d)可知,由于有更多的用户任务分配需要处理,因此所有方法随着更长的任务有效时间时间开销也逐渐增加。

[0257] range的影响。如图17(a)和17(c)所示,随着range的逐渐扩大,所有方法生成的分配任务数量都有着增长趋势,其背后原因与任务有效时间效果相似,即,用户的可达半径越

大,SC系统有更多的机会将更多的任务分配给用户。此外,对于所有的range值,L/R-OTA和L/R-GTA的效果都优于其他算法,这再次证实本申请实施例提出算法的有效性。如图17(b)和17(d)所示,所有的方法的CPU成本都会随着range的增大而增加,因为当range增大时,在一个时间实例中要分配的可用任务数量增加,从而导致更长的时间成本。

[0258] 基于同一发明构思,本申请实施例还提供了一种空间众包任务分配系统,包括处理器和存储器,所述处理器用于执行存储器中存储的计算机程序以实现如以上任意一项所述的空间众包任务分配方法的步骤。

[0259] 基于同一发明构思,本申请实施例还提供了一种计算机可读存储介质,所述计算机可读存储介质存储有计算机程序,所述计算机程序被处理器执行时实现如以上任意一项所述的空间众包任务分配方法的步骤。

[0260] 基于同一发明构思,如图18所示,本申请实施例还提供了一种空间众包任务分配系统,包括第一预测模块1801、第二预测模块1802和任务分配模块1803。

[0261] 具体的,第一预测模块1801用于根据用户的任务执行历史预测用户的未来时空分布;第二预测模块1802用于将预设区域划分成多个不相交的网格,应用网络嵌入的方法预测未来时间戳中每个网格的任务数量,根据各个网格预测的任务数量,预测任务的位置;任务分配模块1803用于根据预测的用户的未来时空分布以及任务的位置,进行任务分配。

[0262] 在一种示例性实施例中,第一预测模块1801根据用户的任务执行历史预测用户的未来时空分布,包括:将用户的任务执行历史作为时间序列数据,利用序列模式挖掘方法从所述时间序列数据中挖掘出现频率超过最小支持阈值的时间实例集合,并将每个连续的时间实例集合作为用户的可用时间;在所述可用时间开始时,基于时空递归神经网络模型预测每个未来用户的空间分布。本实施例中的时空递归神经网络模型的具体构建方法已在前述实施例中说明,具体构建过程请参考前文所述,此处不再赘述。

[0263] 在一种示例性实施例中,每个未来用户的空间分布包括每个未来用户的位置和/或路线。

[0264] 在一种示例性实施例中,第一预测模块1801基于时空递归神经网络模型预测每个未来用户的空间分布,包括:

[0265] 使用基于特征点的道路拐角提取方法从历史轨迹中提取道路拐角;

[0266] 使用基于道路拐角的轨迹映射方法表示历史轨迹数据和将被预测的查询轨迹,生成以道路拐角为中心的路线;

[0267] 使用前缀投射顺序模式挖掘算法从历史轨迹中挖掘数量大于最小支持度阈值的轨迹,作为满足最小支持度阈值的频繁移动模式;

[0268] 基于被发现的频繁移动模式,使用模式匹配方法以预测未来路线;

[0269] 当无模式匹配时,使用时空递归神经网络模型预测下一个行驶中的道路拐角。

[0270] 在一种示例性实施例中,第一预测模块1801使用基于特征点的道路拐角提取方法从历史轨迹中提取道路拐角,包括:通过线性拟合的方法生成一组特征点,所述特征点为轨迹方向发生显著变化的坐标点,使用基于噪音的多密度空间聚类方法检测道路拐角。

[0271] 在一种示例性实施例中,第一预测模块1801使用模式匹配方法以预测未来路线,包括:执行模式匹配过程以找到待匹配轨迹,通过计算待匹配轨迹与频繁移动模式的匹配覆盖率,找到最长的轨迹模式作为待匹配轨迹的预测路线。

[0272] 在一种示例性实施例中，第二预测模块1802应用网络嵌入的方法预测未来时间戳中每个网格的任务数量，包括：

[0273] 基于任务的时空信息建立基于空间任务的网络，所述网络的节点包括基于时间单元的节点$C_i^{t_j}$和基于空间任务的节点B_m，所述网络的边包括空间邻近边$e\left(C_i^{t_j}, C_k^{t_g}\right)$和任务相关边$e\left(C_i^{t_j}, B_m\right)$，所述基于时间单元的节点$C_i^{t_j}$表示在时间戳$t_j$处网格的特定空间单元$C_i$，所述基于空间任务的节点$B_m$表示特定任务类别m的节点，所述空间邻近边$e\left(C_i^{t_j}, C_k^{t_g}\right)$表示两个基于时间单元的节点的空间邻近关系，任务相关边$e\left(C_i^{t_j}, B_m\right)$表示有特定类别m的任务在时间戳$t_j$发布在空间单元$C_i$中；

[0274] 建立空间路径以捕捉空间信息，建立任务相关路径以捕捉与任务相关的信息，建立时间排序的任务相关路径以捕捉时间信息，所述空间路径包括基于时间单元的节点和空间邻近节点，所述任务相关路径包括基于时间单元的节点、基于空间任务的节点和任务相关边，所述时间排序的任务相关路径包括基于时间单元的节点、基于空间任务的节点和任务相关边，且所述基于时间单元的节点按照一个时间单位的增长速率根据时间以升序排序；

[0275] 沿着所述空间路径、任务相关路径以及时间排序的任务相关路径，使用λ长度的随机游走方法，以基于空间任务的网络G作为输入，获得每个节点多个λ长度的随机游走序列；

[0276] 通过SkipGram神经网络模型学习每个节点的向量表示，基于历史数据和其他节点数据，使用回归算法估计每个网格中的任务数量。

[0277] 本实施例中的每个网格中的任务数量的具体估计方法已在前述实施例中说明，具体估计过程请参考前文所述，此处不再赘述。

[0278] 在一种示例性实施例中，任务分配模块1803根据预测的用户的未来时空分布以及任务的位置，进行任务分配，包括：

[0279] 基于用户可达距离和任务失效时间的约束，建立在给定时间段中每个用户能够执行的可达任务集合；

[0280] 通过迭代地扩展可达任务集合，计算每个用户的所有极大有效任务集的集合，极大有效任务集的计算公式如下：

[0281] $$\mathrm{opt}(Q, s_j) = \begin{cases} 1, & |Q| = 1 \\ \max\limits_{s_i \in Q, s_i \neq s_j} \mathrm{opt}(Q - \{s_j\}, s_i) + \delta_{ij}, & \text{其他} \end{cases}$$

[0282] $$\delta_{ij} = \begin{cases} 1, & t(s_j.l) \leq s_j.e, t(s_j.l) \leq t + n \\ 0, & \text{其他} \end{cases}$$

[0283] 其中，opt(Q,s_j)为在用户的可达范围内通过调度Q中的所有任务最终被完成的最大任务数量，其中，Q中的任务从w.l到s_j.l，R为实现这一最大任务数量的有序任务序列，s_i表示R中到达任务s_j之前的前一个任务，R′表示opt(Q-{s_j},s_i))中对应的任务序列，δ_{ij}表示在给定时间段T＝{t,t+1,...,t+n}内在将s_j附加在R′后可以完成s_j的概率；

[0284] 根据每个用户的所有极大有效任务集的集合，进行任务分配。

[0285] 在一种示例性实施例中，任务分配模块1803根据每个用户的所有极大有效任务集的集合，进行任务分配，包括：

[0286] 将用户集合和任务集合作为输入，初始化一个空的任务分配；

[0287] 开始迭代，在每次迭代中，从剩下的将分配的用户中随机选择一个用户并且从未分配的任务中为选择的用户找到最大的极大有效任务集，并将找到的最大的极大有效任务集加入到当前任务分配中；

[0288] 迭代结束，在所有迭代中找到最大任务分配。

[0289] 在一种示例性实施例中，任务分配模块1803根据每个用户的所有极大有效任务集的集合，进行任务分配，包括：

[0290] 对于用户集合和任务集合，根据用户之间的依赖关系构造用户依赖图，所述依赖关系包括互相独立或互相依赖，当两个用户没有共同的可达任务时，两个用户互相独立；当两个用户有共同的可达任务时，两个用户互相依赖；

[0291] 将所述用户依赖图划分成多个节点集，以分解用户的依赖关系；

[0292] 根据划分的多个节点集，建立平衡树，所述平衡树中的兄弟节点互相独立；

[0293] 通过深度优先搜索算法为树节点中的每个用户计算有效任务集合，以找到最优分配。

[0294] 在一种示例性实施例中，任务分配模块1803将用户依赖图划分成多个节点集，以分解用户的依赖关系，包括：

[0295] 按照节点度数的升序依次对用户依赖图中的节点进行简约处理，直至满足下列条件之一：(1)原图被简约成一个简单图或空图；(2)不存在度小于或等于k的节点；所述简约处理包括：对于指定的度i，i为自然数且0≤i≤k，找到度为i的节点v，并检查节点v的所有邻居是否构成一个团，所述团指一个两两之间有边的节点集合，若无法构成一个团，通过添加边来构造团；将节点v和它的邻居节点构成的团存储于一个堆栈中，在原图中删除节点v及其对应的边；

[0296] 堆栈中的所有团和未被所述简约处理删除的团构成划分成的多个节点集。

[0297] 在一种示例性实施例中，任务分配模块1803根据划分的多个节点集，建立平衡树，包括：

[0298] 从用户依赖图中删除每个节点集的节点，将用户依赖图划分为多个子图，其中，最大的子图记录为Gmax，选择能生成最小|Gmax|值的节点集作为根节点集Xmin，其中，|Gmax|是图Gmax的节点数量；

[0299] 对删除根节点集Xmin后的各个子图，采用k度图简约算法继续划分节点集，并对继续划分的节点集，递归地寻找根节点集。

[0300] 基于同一发明构思，本申请实施例还提供了一种空间众包任务分配方法，包括步骤1901至步骤1903。

[0301] 其中，步骤1901包括：云端服务器接收客户端的任务分配请求，获取用户的任务执行历史以及已发布的任务的时空信息；

[0302] 步骤1902包括：云服务器根据用户的任务执行历史预测用户的未来时空分布，并将预设区域划分成多个不相交的网格，应用网络嵌入的方法预测未来时间戳中每个网格的

任务数量，根据各个网格预测的任务数量，预测任务的位置；

[0303] 步骤1903包括：云端服务器根据预测的用户的未来时空分布以及任务的位置，进行任务分配，将任务分配的结果发送至客户端。

[0304] 在一种示例性实施例中，客户端可以为接收和处理任务的用户，也可以为用于管理多个用户的任务分配的服务商。

[0305] 本实施例中，云端服务器具体如何预测用户的未来时空分布、未来时间戳中每个网格的任务数量以及任务的位置等，请参照前文所述，此处不再赘述。

[0306] 在一种示例性的应用场景中，如图20所示，服务商发送任务分配请求至云端服务器，云端服务器获取用户的任务执行历史以及已发布的任务的时空信息，具体的获取方法可以为以下任意一种方法：由服务商发送至云端服务器，或者由云端服务器从预先设置的存储服务器上获取。云端服务器使用本申请所述的空间众包任务分配方法，预测用户的未来时空分布、未来时间戳中每个网格的任务数量以及任务的位置等，根据预测的结果进行任务分配，并将任务分配的结果发送至服务商，服务商根据任务分配的结果，发送一个或多个任务到对应的用户。用户接收到任务后，处理任务并返回任务结果至服务商。

[0307] 在一种示例性实施例中，用户可以提交行程计划至服务商，再由服务商将用户提交的行程计划发送至云端服务器或用户可以直接提交行程计划至云端服务器，云端服务器根据用户提交的行程计划，生成用户的未来时空分布。

[0308] 在一种示例性实施例中，用户接收到任务后，可以对下发的任务进行修改。例如，当用户接收到一个任务后，用户可以在下发的任务上添加任务反馈，示例性的，当用户发现某个任务由于一些客观原因(比如道路施工等)无法完成时，用户可以在该任务上添加一条这样的任务反馈：由于道路施工，无法执行该任务。

[0309] 在一种示例性实施例中，用户对下发的任务进行修改后，可以向一个或多个用户分享修改后的任务。例如，当用户在某个任务上添加一条“由于道路施工，无法执行该任务”这样的任务反馈后，可以将该任务分享给其他用户，以使得其他用户知晓该任务的反馈情况。

[0310] 本实施例通过利用云端服务器强大的计算处理能力来预测用户的未来时空分布、未来时间戳中每个网格的任务数量以及任务的位置等，能够在任务分配过程中达到最佳分配，实现空间众包任务的高效分配，具有较强的可实施性，可以为服务商或用户提供高质量的服务。

[0311] 本领域普通技术人员可以理解，上文中所公开方法中的全部或某些步骤、系统、装置中的功能模块/单元可以被实施为软件、固件、硬件及其适当的组合。在硬件实施方式中，在以上描述中提及的功能模块/单元之间的划分不一定对应于物理组件的划分；例如，一个物理组件可以具有多个功能，或者一个功能或步骤可以由若干物理组件合作执行。某些组件或所有组件可以被实施为由处理器，如数字信号处理器或微处理器执行的软件，或者被实施为硬件，或者被实施为集成电路，如专用集成电路。这样的软件可以分布在计算机可读介质上，计算机可读介质可以包括计算机存储介质(或非暂时性介质)和通信介质(或暂时性介质)。如本领域普通技术人员公知的，术语计算机存储介质包括在用于存储信息(诸如计算机可读指令、数据结构、程序模块或其他数据)的任何方法或技术中实施的易失性和非易失性、可移除和不可移除介质。计算机存储介质包括但不限于RAM、ROM、EEPROM、闪存或其

他存储器技术、CD-ROM、数字多功能盘(DVD)或其他光盘存储、磁盒、磁带、磁盘存储或其他磁存储装置、或者可以用于存储期望的信息并且可以被计算机访问的任何其他的介质。此外,本领域普通技术人员公知的是,通信介质通常包含计算机可读指令、数据结构、程序模块或者诸如载波或其他传输机制之类的调制数据信号中的其他数据,并且可包括任何信息递送介质。

任务的时空分布			
	位置	生成时间	失效时间
s_1	(3.5, 3)	2	5
s_2	(3.5, 2)	3	6
s_3	(3, 1.5)	4	8
s_4	(5, 2.5)	4	7
s_5	(5.5, 1.5)	3	8
s_6	(8, 1.5)	2	5
s_7	(7.2, 2.8)	4	6
s_8	(6.5, 2.8)	3	7

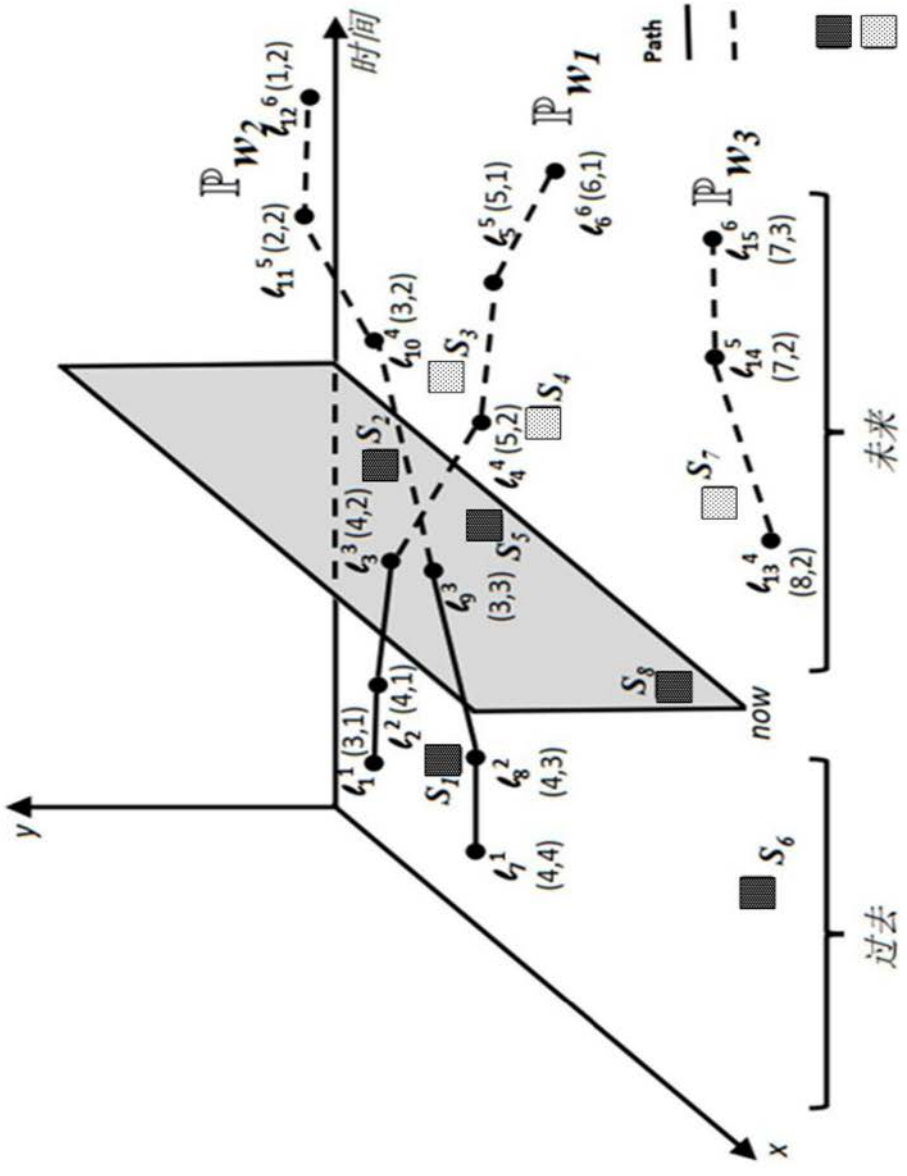


图1

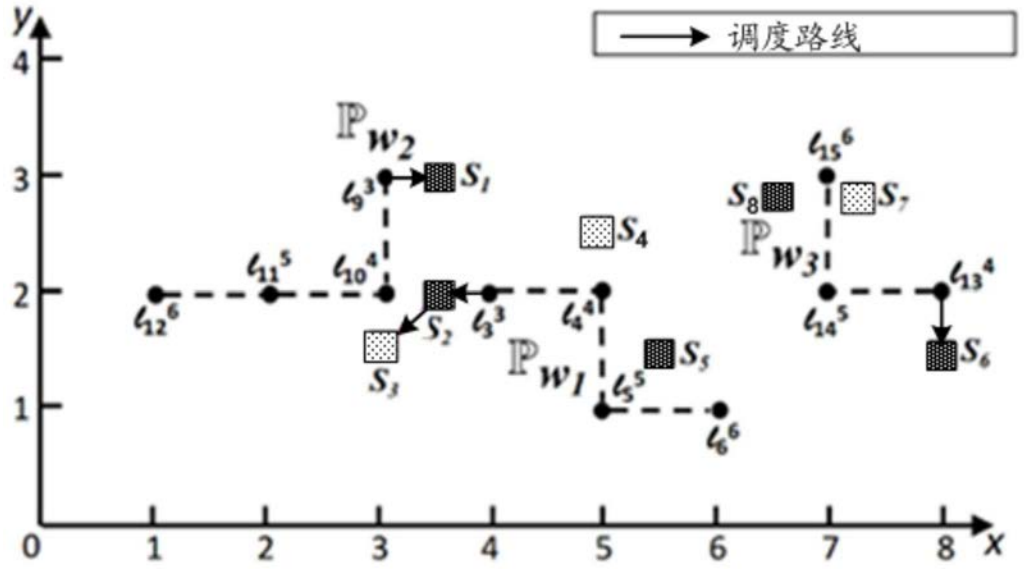

调度路线
y
x
0
1
2
3
4
5
6
7
8
$\mathbb{P}_{w_2}$
$\mathbb{P}_{w_1}$
$\mathbb{P}_{w_3}$
S_1
S_2
S_3
S_4
S_5
S_6
S_7
S_8

图2

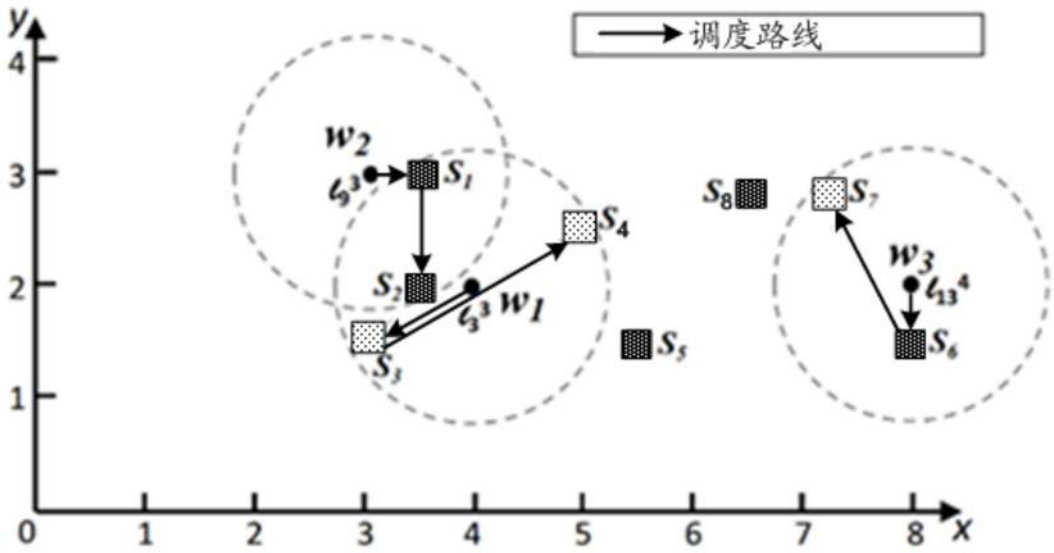

调度路线
y
x
0
1
2
3
4
5
6
7
8
w_2
w_1
w_3
S_1
S_2
S_3
S_4
S_5
S_6
S_7
S_8

图3

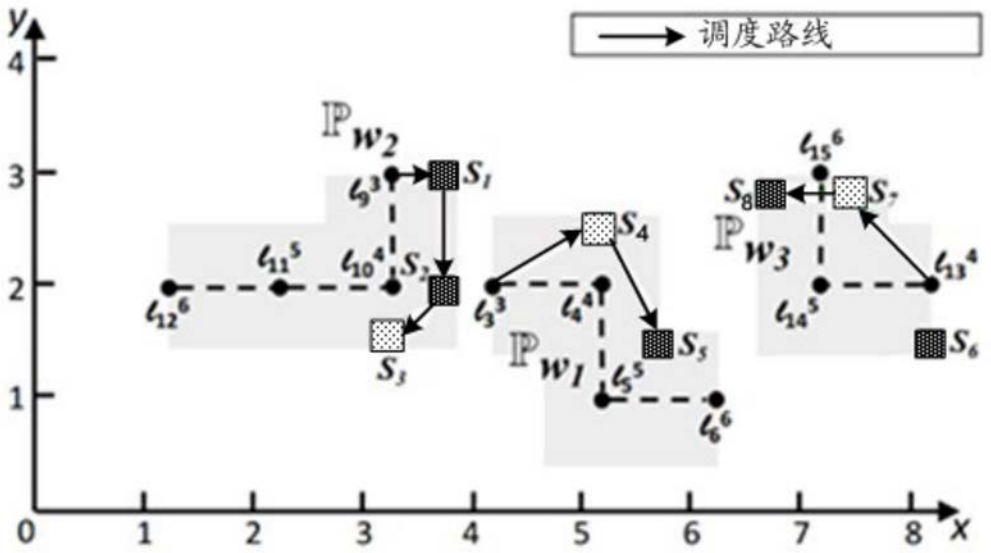

调度路线
y
x
0
1
2
3
4
5
6
7
8
$\mathbb{P}_{w_2}$
$\mathbb{P}_{w_1}$
$\mathbb{P}_{w_3}$
S_1
S_2
S_3
S_4
S_5
S_6
S_7
S_8

图4

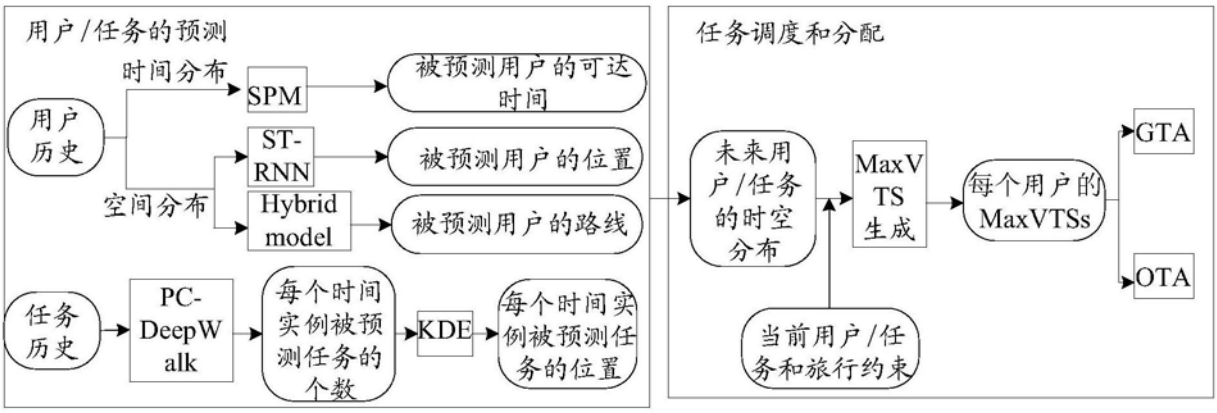

用户/任务的预测
时间分布
空间分布
用户历史
SPM
ST-RNN
Hybrid model
被预测用户的可达时间
被预测用户的位置
被预测用户的路线
任务历史
PC-DeepWalk
每个时间实例被预测任务的个数
KDE
每个时间实例被预测任务的位置
任务调度和分配
未来用户/任务的时空分布
MaxVTS生成
每个用户的MaxVTSs
GTA
OTA
当前用户/任务和旅行约束

图5(a)

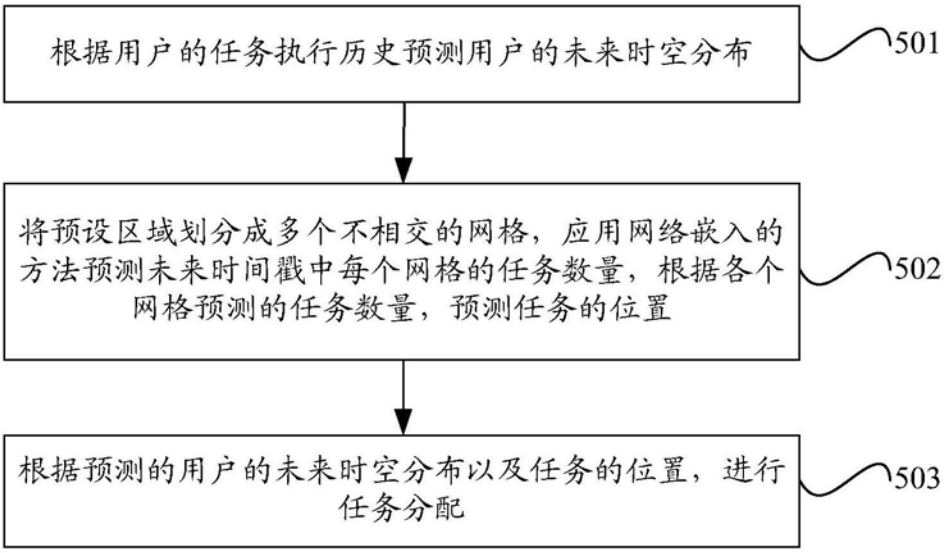

根据用户的任务执行历史预测用户的未来时空分布
501
将预设区域划分成多个不相交的网格，应用网络嵌入的方法预测未来时间戳中每个网格的任务数量，根据各个网格预测的任务数量，预测任务的位置
502
根据预测的用户的未来时空分布以及任务的位置，进行任务分配
503

图5(b)

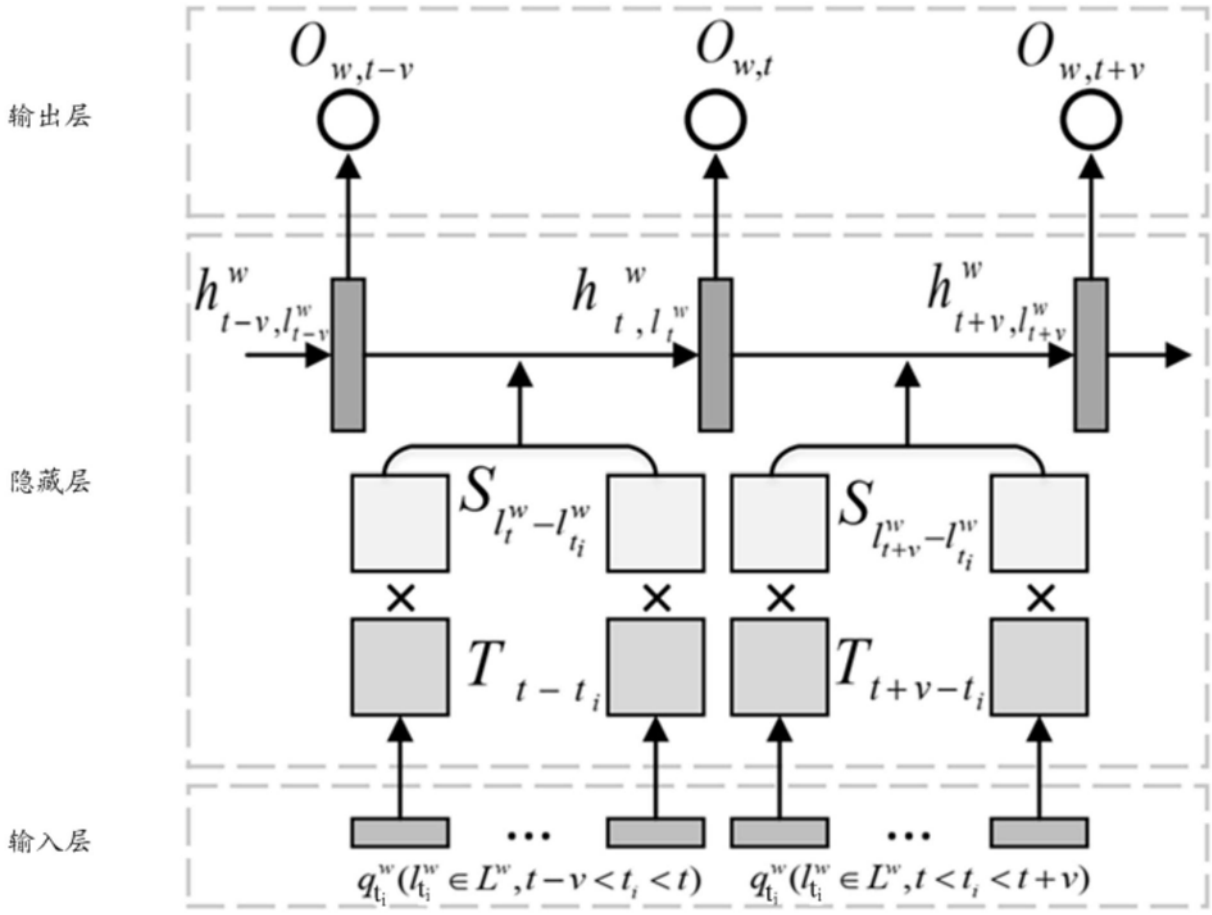

输出层
隐藏层
输入层
$O_{w,t-v}$
$O_{w,t}$
$O_{w,t+v}$
$h^w_{t-v,l^w_{t-v}}$
h^w_{t,l^w_t}
$h^w_{t+v,l^w_{t+v}}$
$S_{l^w_t - l^w_{t_i}}$
$S_{l^w_{t+v} - l^w_{t_i}}$
T_{t-t_i}
T_{t+v-t_i}
$q^w_{t_i}(l^w_{t_i} \in L^w, t-v < t_i < t)$
$q^w_{t_i}(l^w_{t_i} \in L^w, t < t_i < t+v)$

图6

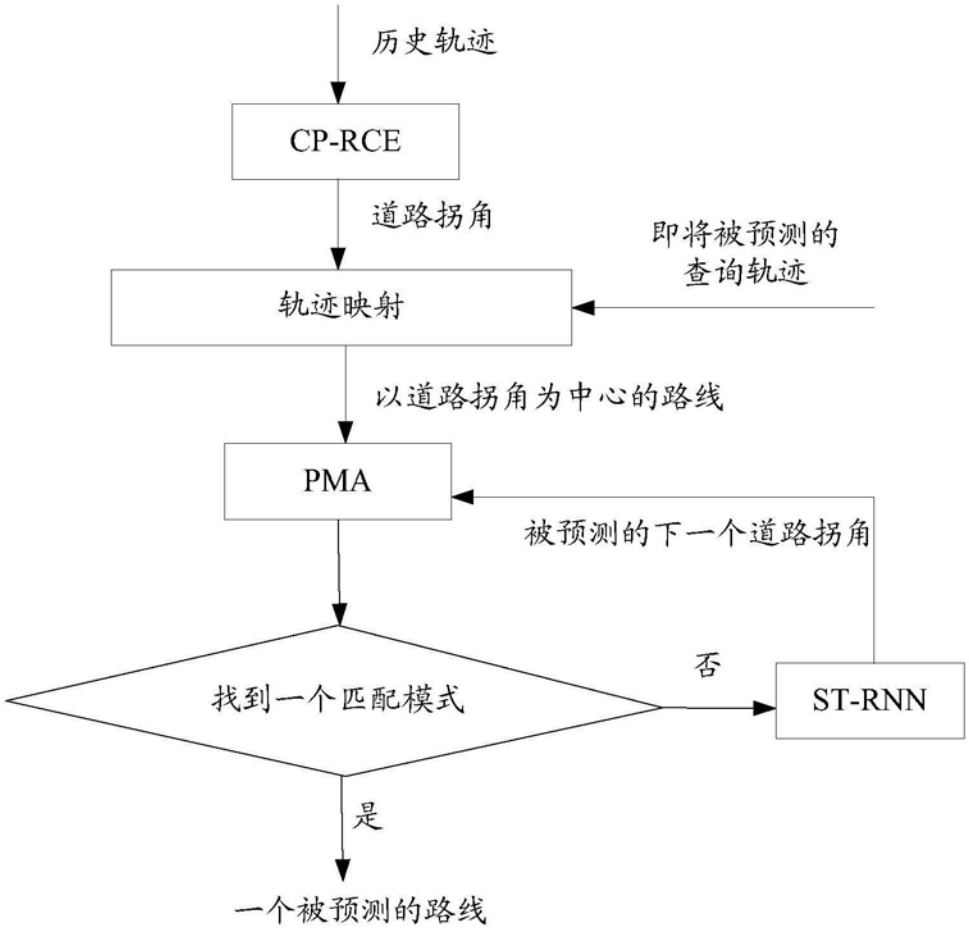

历史轨迹
CP-RCE
道路拐角
即将被预测的
查询轨迹
轨迹映射
以道路拐角为中心的路线
PMA
被预测的下一个道路拐角
找到一个匹配模式
否
ST-RNN
是
一个被预测的路线

图7

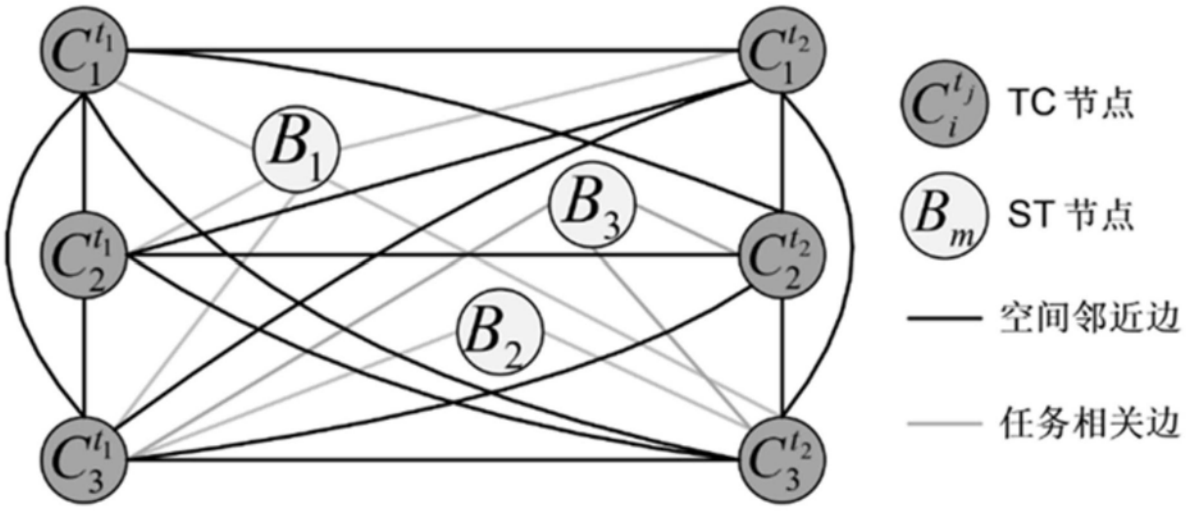

$C_1^{t_1}$
$C_1^{t_2}$
B_1
B_3
$C_2^{t_1}$
$C_2^{t_2}$
B_2
$C_3^{t_1}$
$C_3^{t_2}$
$C_i^{t_j}$ TC 节点
B_m ST 节点
空间邻近边
任务相关边

图8

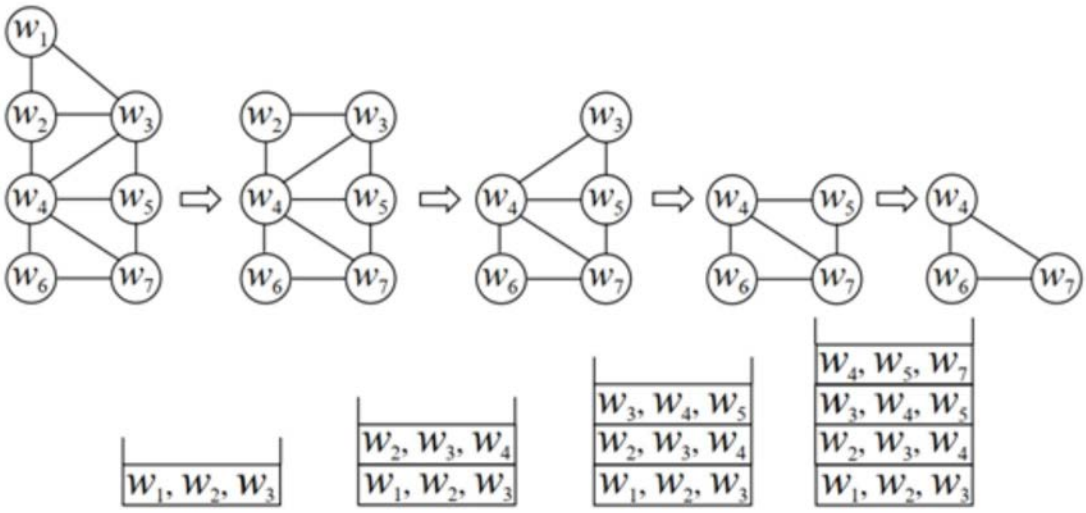

w_1
w_2 w_3
w_4 w_5
w_6 w_7
w_1, w_2, w_3
w_2, w_3, w_4
w_3, w_4, w_5
w_4, w_5, w_7

图9

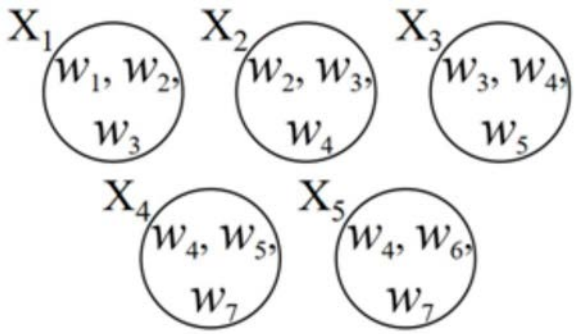

X_1 w_1, w_2, w_3
X_2 w_2, w_3, w_4
X_3 w_3, w_4, w_5
X_4 w_4, w_5, w_7
X_5 w_4, w_6, w_7

图10

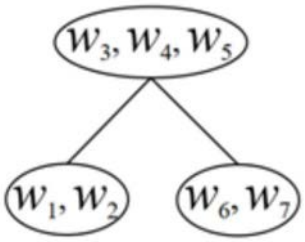

w_3, w_4, w_5
w_1, w_2
w_6, w_7

图11

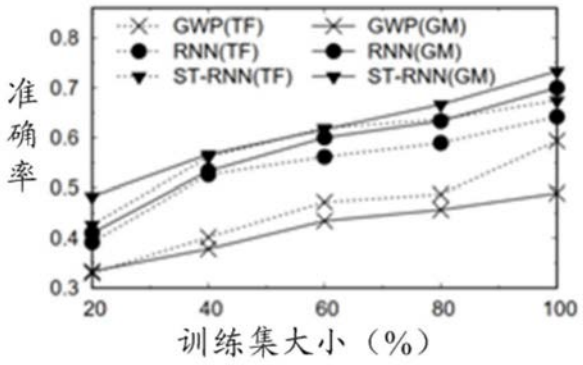


图12(a)

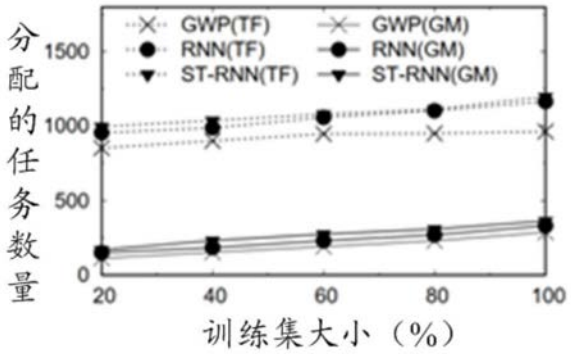


图12(b)

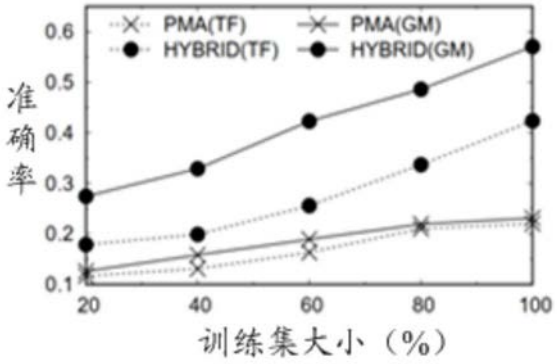


图13(a)

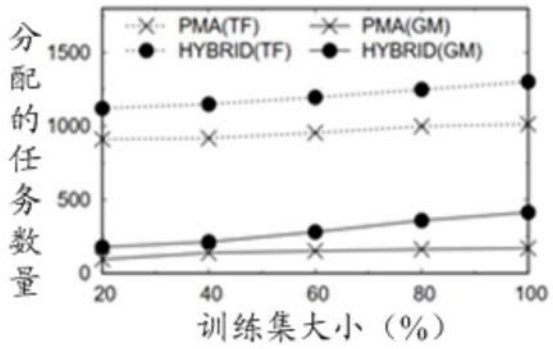


图13 (b)

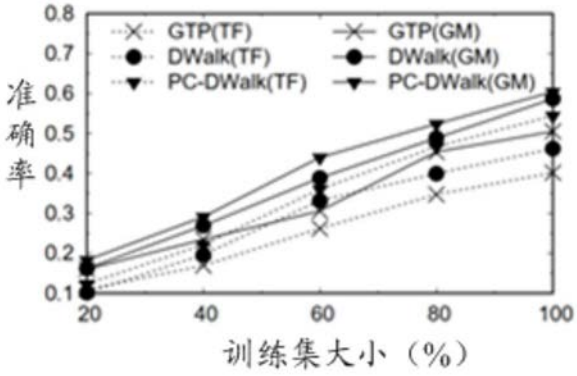


图14 (a)

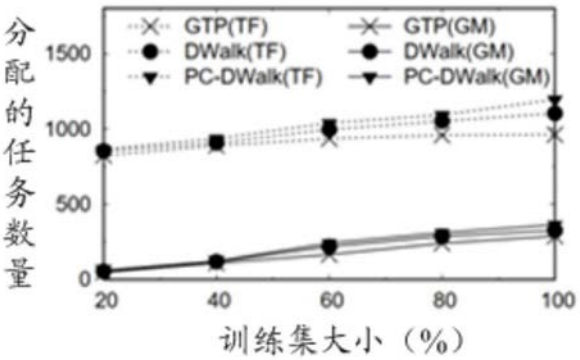


图14 (b)

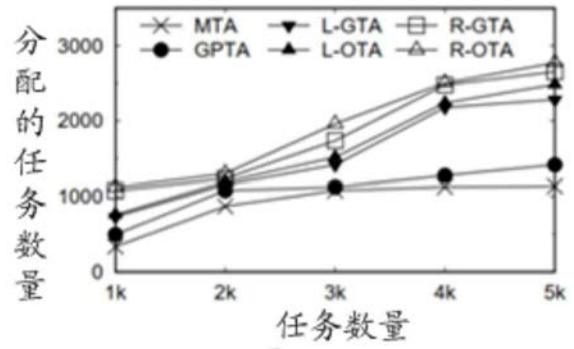


图15(a)

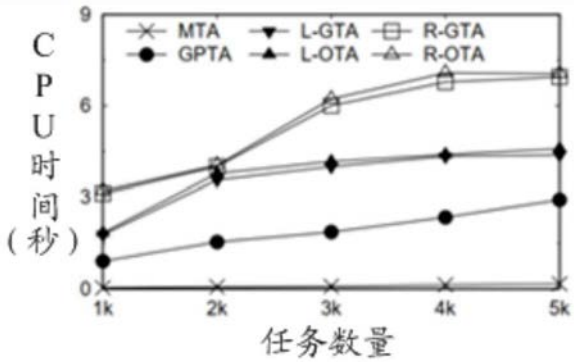


图15(b)

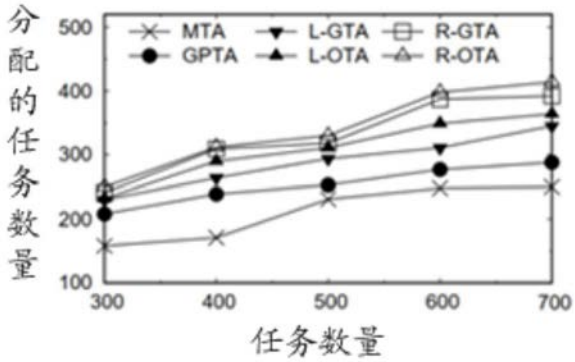


图15(c)

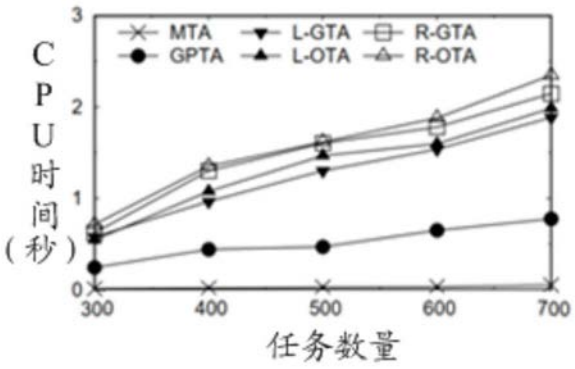


图15 (d)

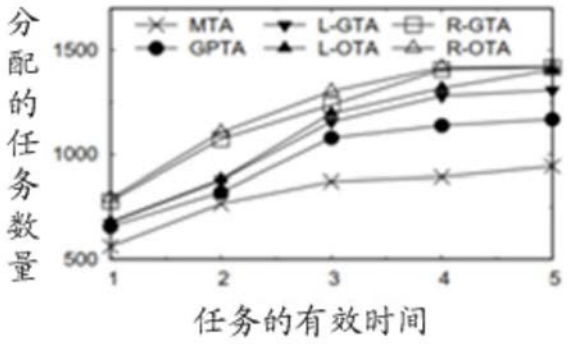


图16 (a)

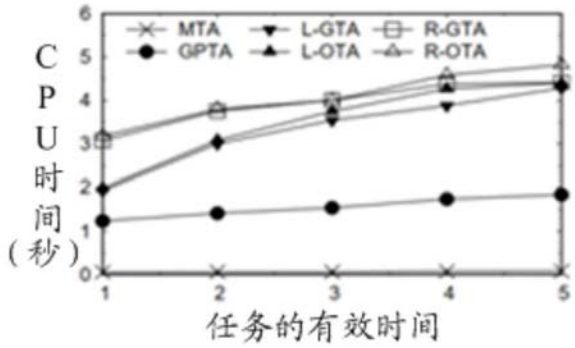


图16 (b)

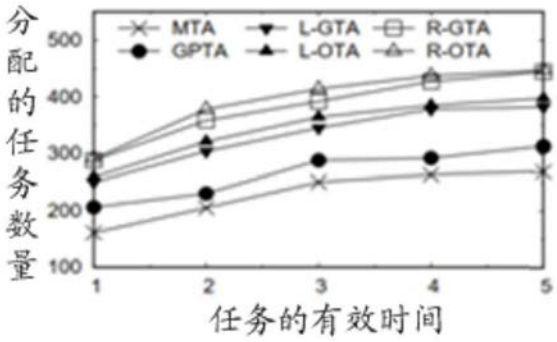


图16 (c)

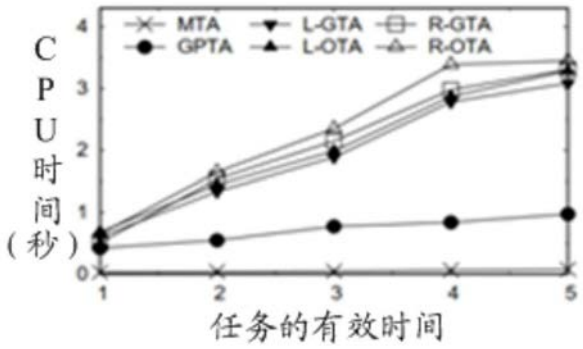


图16 (d)

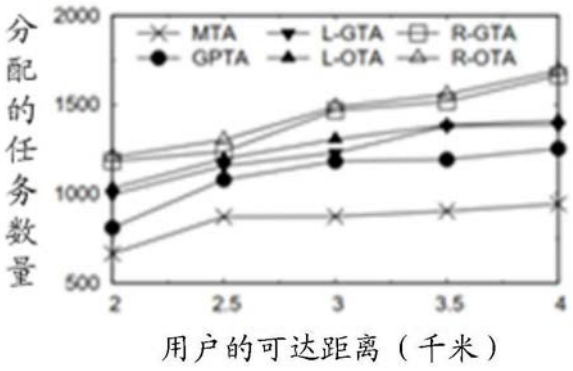


图17 (a)

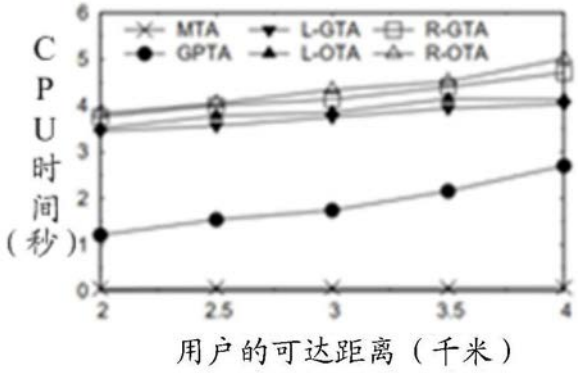


图17(b)

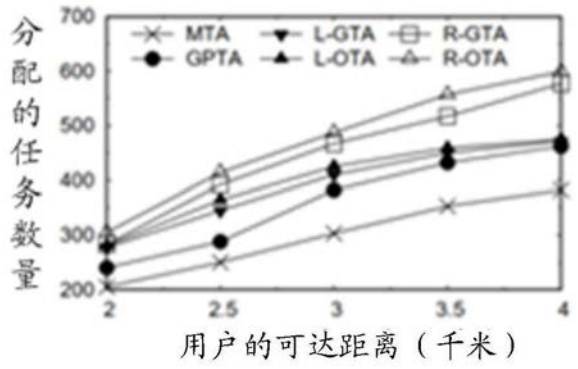


图17(c)

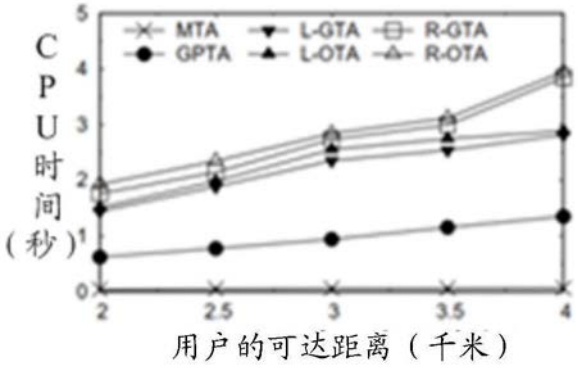


图17(d)

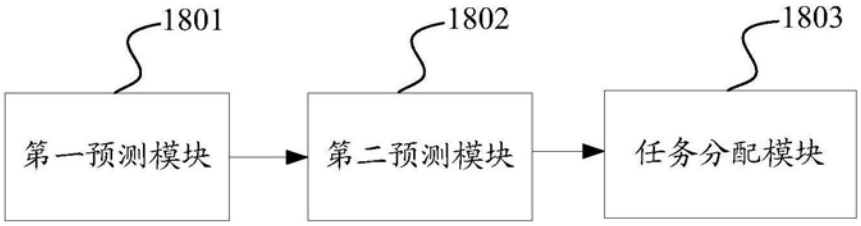

1801
1802
1803
第一预测模块
第二预测模块
任务分配模块

图18

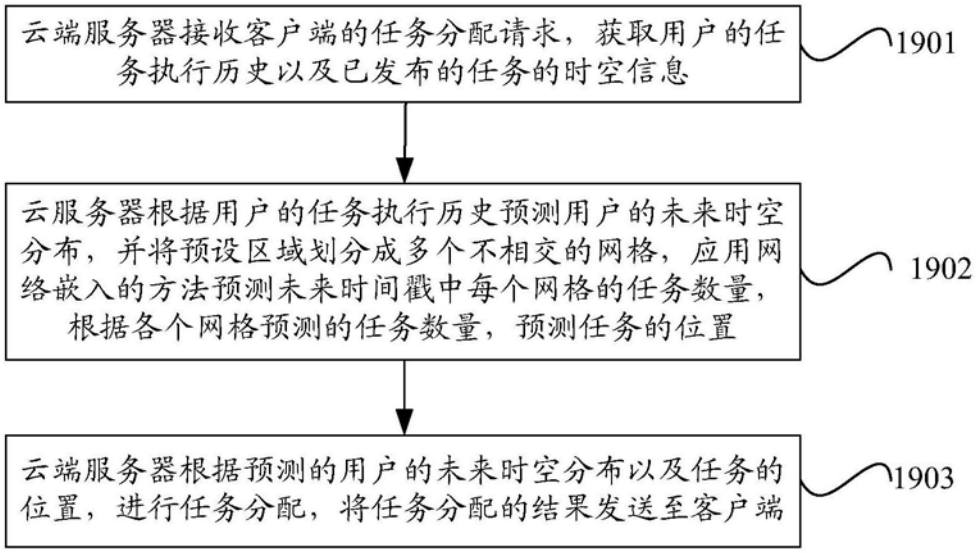

云端服务器接收客户端的任务分配请求，获取用户的任务执行历史以及已发布的任务的时空信息
1901
云服务器根据用户的任务执行历史预测用户的未来时空分布，并将预设区域划分成多个不相交的网格，应用网络嵌入的方法预测未来时间戳中每个网格的任务数量，根据各个网格预测的任务数量，预测任务的位置
1902
云端服务器根据预测的用户的未来时空分布以及任务的位置，进行任务分配，将任务分配的结果发送至客户端
1903

图19

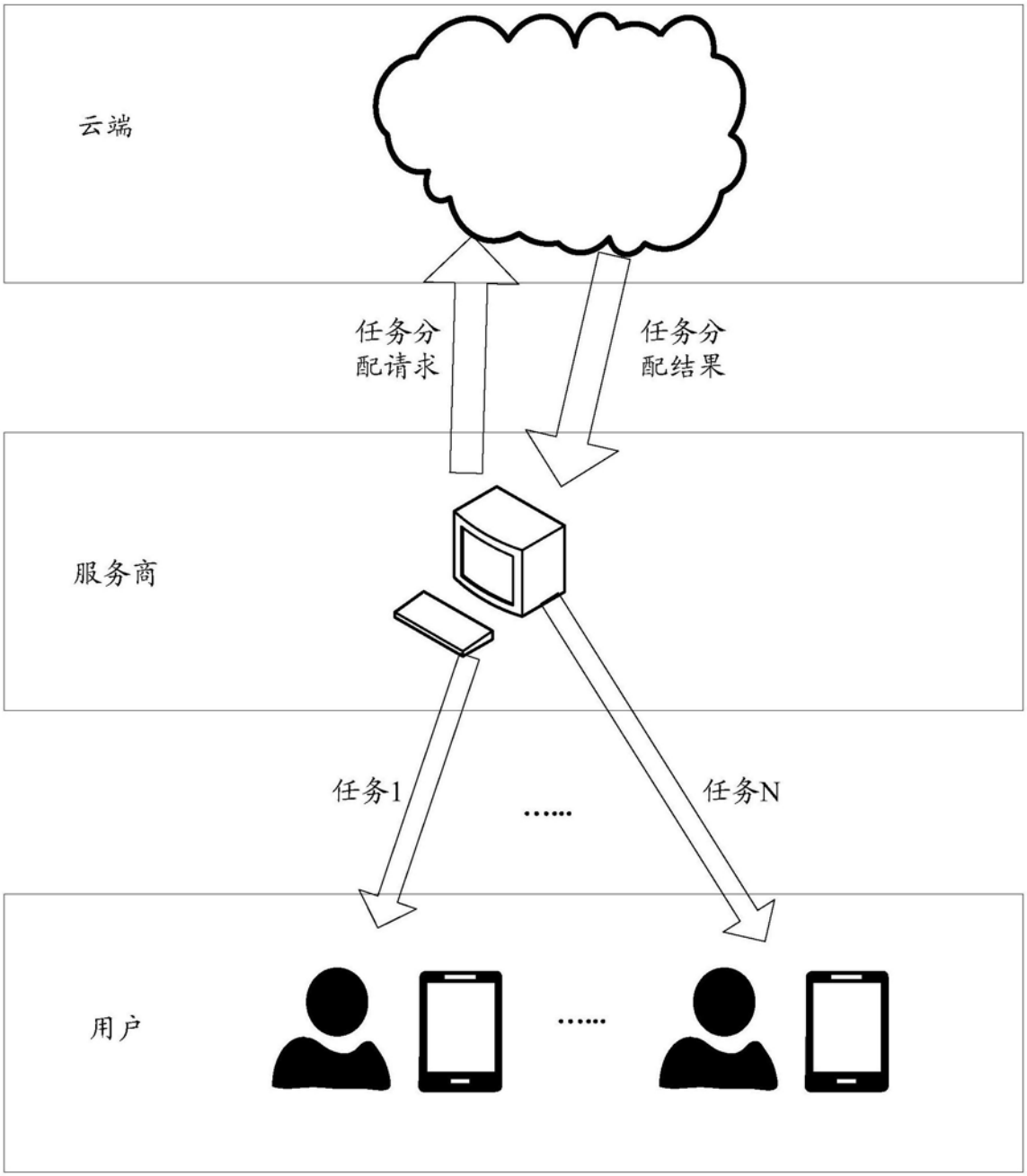


图20