软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2012,23(Suppl.(1)):159−168

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

基于物理轨迹数据和社会网络的泛化行程推荐*

孟祥旭+, 王晓东, 周兴铭

(并行与分布处理国家重点实验室(国防科学技术大学 计算机学院),湖南 长沙 410073)

Intention Oriented Itinerary Recommendation by Bridging Physical Trajectories and Online Social Networks

MENG Xiang-Xu+, WANG Xiao-Dong, ZHOU Xing-Ming

(National Key Laboratory of Parallel and Distributed Processing (College of Computer Science, National University of Defense Technology), Changsha 410073, China)

+ Corresponding author: E-mail: yumengkk@21cn.com

Meng XX, Wang XD, Zhou XM. Intention oriented itinerary recommendation by bridging physical trajectories and online social networks. *Journal of Software*, 2012,23(Suppl.(1)):159−168 (in Chinese). http://www.jos.org.cn/1000-9825/12018.htm

Abstract: Human itineraries are often initiated by some general intentions and will be optimized after considering all kinds of constraints and available information. This paper proposes a category-based itinerary recommendation framework to help the user transfer from intentions to itinerary planning, which join physical trajectories and information of location based social networks. The main contributions are: (1) Build the category based activity scheduling model; (2) Design and implement the category tree based POI (point or interest) query strategy and algorithm; (3) Propose the Voronoi graph based GPS trajectory analysis method to build traffic information networks; (4) Combine social networks with traffic information networks to implement category based recommendation by ant colony algorithm. The study conducts experiments on datasets from FourSquare and GeoLife project. A test on satisfaction of recommended items is also performed. Results show that the satisfaction reaches 80% in average.

Key words: itinerary planning; trajectory mining; location-based system; multi-level categories

摘 要: 人类活动行程的制定往往基于宽泛的最初意向,通过综合考虑各种约束条件加以优化而完成.当前,基于位置点名称查找的行程制定方法,不支持用户一次性提交多个具有时序关系的宽泛出行意向,更不能同时为多个地理位置点提供详细的最优驾车方案.基于位置社交网络信息和车辆历史轨迹数据,探索了支持用户多个模糊意向输入的泛化行程推荐框架,主要工作包括:(1) 对泛化的行程推荐问题进行建模;(2) 设计并实现了基于分类树的地理位置点(POI)查询策略和算法;(3) 提出了基于 Voronoi 图的 GPS 轨迹分析模型,并实现了任意两个位置点间最优行驶路径计算方法;(4) 联合社会网络和语义交通信息图,基于蚁群算法进行行程的推荐,并实现了原型系统.实验及问卷调查结果表明,推荐结果的用户满意度可达 80%.

关键词: 行程规划;轨迹挖掘;LBS(location-based system);多级目录

* 基金项目: 国家自然科学基金(61170260)
收稿时间: 2012-05-05; 定稿时间: 2012-08-17

1 背景介绍

在日常生活中,用户的出行安排本质上是寻找一组能够满足生活需求的地理位置,并且希望到达这些位置的交通最便利.正如文献[1]所述,用户通常希望找到一组服务对象来满足多种需求.比如,一个旅行者有多个需求:购物,住宿,就餐,观光.这些需求将由多个地理位置对象满足.当然,评价该组目标的好坏有多种标准,有些人希望服务最好,有些人希望开销最低.在行程安排的最初阶段,用户往往只是有一个大概的意向,之后需要从多个方面获取相关信息,仔细比较,权衡后才会做出最后决定.比如,咨询旅行社、请求朋友推荐、查看旅游论坛等,为了保障旅行顺利,还要考虑目的城市的交通信息.

当用户希望到一个较陌生的城市旅行时,他的初步意向比较模糊.比如一个外地游客到北京观光的需求:

示例 1:首先去一个快餐店吃早餐,然后去有历史韵味的博物馆参观,中午找一个有特色的中餐厅就餐,下午去广场走一走,晚上首先去给亲属购买些礼物,之后去酒吧.当然,希望去的地方名气较大,交通要便利.在最终决定去哪些地点之前,希望对所去的地点具有清楚的了解,对交通上花费的时间有准确的估计.

通过如上分析可知,陌生游客对行程规划的需求具有如下特征:

(1) 出行目标较模糊,对于希望观光的对象只有一个大概描述(只是一个类别,不涉及具体地理位置对象),比如著名的旅游景点、出名的小吃店等.

(2) 希望清楚地了解各个景点的位置以及在任意两个景点间的乘车路线和乘坐出租车需要的最短、平均时间.

(3) 希望了解其他客户对于地理位置点的评价,比如该位置点的特色、口碑如何,以便最终选择一个最佳方案.

(4) 有多个优化目标,有些是能够清楚表述的,比如“在路上浪费的时间最短”(可以进行数学上的量化),有些是一个模糊的概念,比如“去的旅游景点尽量比较热门”(很难量化表示).

对于计算机而言,实现这类查询或者推荐非常困难,因为其表述比较模糊,做出推荐所需的信息非常宽泛.即使人类导游也很难给出这种推荐,因为他很难对城市的旅馆、饭店、旅游景点等海量信息均非常熟悉,并且知道其具体位置点及出行路线.同时,基于 GIS 的最短路线搜索,由于没有考虑到道路状况的实时变化、出行时间(周末,上下班时间),因此很难确定位置点间的最佳行驶路径和所需最短时间.为了解决这种难题,帮助用户做出最佳决策,我们联合使用多个用户的历史轨迹和社会化网路的信息,提供基于分类的最佳行程推荐服务.幸运的是,当前的社交网络和轨迹挖掘技术可以为该服务提供有效支持.

(1) 当前,大部分移动设备都具有定位功能,如基于网络的定位或者 GPS 定位,使得收集移动用户的轨迹数据变得非常简单.同时,在城市中,交通、规划等部门也会收集大量的车辆、手机等设备的位置信息.如微软的 T-drive 工程[2,3],基于出租车司机对城市道路系统最为熟悉这一事实,通过挖掘大量的出租车行驶记录,为驾驶员提供实时的导航服务.

(2) 基于位置的社会化网络(location based social network,简称 LBSN),如 FourSquare[4]和街旁网[5]等,支持用户对自己喜欢的、去过的地理位置点进行标注,给出评分,添加评论等.这些信息包含了百万级用户的经验知识,对行程的推荐具有极高的参考价值.

这些新技术的普及为实现精准的行程推荐提供了机会,据我们所知,当前还没有商业系统提供基于泛化需求的行程推荐服务.本文将对该问题进行建模(见第 2 节),并基于蚁群算实现行程的推荐(见第 3 节、第 4 节).

2 模型和框架

2.1 活动调度建模

活动一定与地理位置点相关联,比如吃饭必须在一个餐厅地理位置点内进行.因此,一个活动实例可以描述为一个三元组: $A:=\langle P,T,condition\rangle$. P 为位置点(Place 的缩写),表示活动所在的地理位置点,该位置点表示一个具体的地点,如故宫、奥林匹克公园等.T 表示 Time,为活动的时间描述.Condition 可以是对活动任意属性的约束,

比如活动的支出不高于 100 元/人,出行工具为出租车等.位置具有层次性,用户通常不能详细地描述想去的具体位置点,而只是给出一个类别,如酒吧、购物中心等.本文采用 Foursquare 中的地理位置点分类方法.一个类别 C (category)是一组位置点的集合:

$$C := \{P_1, P_2, P_3, ..., P_n\}.$$

类别之间存在层次关系,位置点可以包含在类别中:

$$C_1 \subset C_2 \ , P_i \in C_i.$$

一个路径 Path 表示两个地理位置点间的连接路段,i 为描述信息,比如行驶时间:

$$Path := \langle P_s, P_e, i \rangle.$$

一个行程 Trip 是多个活动实例和路径的时序描述:

$$Trip := \{A_1, Path_1, A_2, Path_2, ..., A_n\}.$$

本文使用问号(?)表示类别中任意一个位置点,星号(*)表示类别中的所有位置点,则用户的模糊行程需求可用如下方式描述:

$$Need := \langle \langle ?C_1, T_1, condition_n \rangle, ..., \langle ?C_n, T_n, condition_n \rangle : optimize \rangle,$$

其中,*optimize* 表示优化目标,如路程时间最短、景点热门.根据该模型,第 1 节的示例 1 可以描述为

$\langle \langle ?\text{Fast Food Restaurant}, Workday-8:00, Taxi \rangle, \langle ?\text{Historic Site}, Workday-9:00, Taxi \rangle,$

$\langle ?\text{Chinese Restaurant}, Workday-12:00, Taxi \rangle, \langle ?\text{Plaza}, Workday-13:00, Taxi \rangle,$

$\langle ?\text{Mall}, Workday-18:00, Taxi \rangle, \langle ?\text{Wine Bar}, Workday-20:00, Taxi \rangle : [\text{Min}TravelTime, AllPopular] \rangle.$

2.2 城市网络模型

在确定行程的推荐顺序时,需要设置一个开销函数计算行程的总开销 Cost,该开销可以是花费的时间,也可以是所有位置点的服务均最佳,多个位置点间的距离最短.当需要确保交通时间开销最小时,则需要知道每两个位置点间的驾驶时间开销.我们根据历史的出租车轨迹,建立一个城市的语义交通信息图: $G = \{P, Path\}$ .P 是所有的地理位置点的集合,*Path* 为位置点间的边集合.因此,图的构建需要获取所有位置点 P,以及任何两个位置点间的 *path* 信息.详细的语义交通信息图构建方法将在第 3 节加以描述.

2.3 行程规划框架

本系统从社交网络获取地理位置点信息,通过 GPS 轨迹挖掘建立语义交通信息图.针对具体的开销函数为用户推荐多个备选行程方案,并给出每个行程的详细解释.比如,推荐行程中位置点的热门程度,两个相邻位置点间的行驶时间.

如图 1 所示,整个系统包含 3 个信息库:

地理位置点信息库:从 LBSN 爬取目标城市的所有位置点,这些位置点是由不同的用户提供和评价的.这些协同标记的信息可以更加客观地表示位置点的热门程度;

地点目录信息:保存位置点的分类信息.本文采用 Foursquare 提供的位置点分类树;

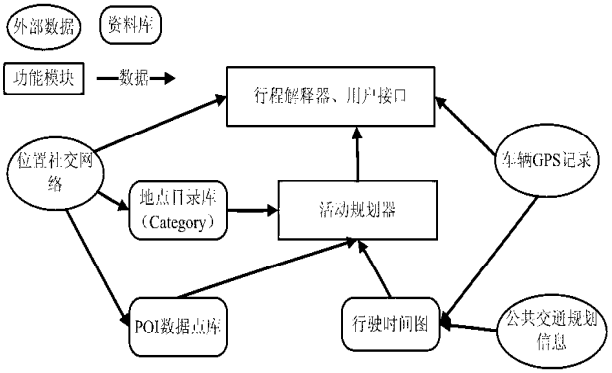

Fig.1 Framework of joint itinerary planning

图 1 行程推荐框架图

交通信息网络:该信息库保存城市中任意两个语义位置点的最短行驶时间和最佳路径.本文从公共交通机构获得用户关注的语义位置,通过挖掘出租车的历史轨迹数据获得准确的最短路径信息(第 3 节加以描述).

同时,活动规划器负责使用如上信息,根据用户的泛化需求为用户推荐合理的行程.用户的 UI 结合 Map 系统更加直观地展示具体的行程,并支持用户交互式地添加修改需求.

3 基于 Voronoi 划分的语义交通信息图

当前,交通部门采集大量公共交通车辆的轨迹信息.高频度、长时间的 GPS 数据,具有数据量巨大、处理困难的特性.同时,物理轨迹点只包含空间和时间数据,不能够体现语义信息[6].然而,用户更关心人类易于理解的语义信息,比如“周日早晨 8:00 从北京火车站驾车到颐和园最短需要多少时间,最快捷的路径是哪一条”.历史的轨迹数据,尤其是对城市熟悉的人的历史轨迹数据,包含了如上所需的信息.将物理轨迹转换为可用的语义知识,存在如下困难:

(1) 如何确定用户容易理解、最为熟悉的物理位置点;

(2) 如何获得任意两个位置点间的最短路径和行驶的统计信息.

本节提出基于公共交通站点的语义位置点确定方法和基于空间 Voronoi 划分的语义交通信息图构建算法,使获取任何两个位置点间的最短驾驶时间和最优驾驶路径成为可能.

3.1 基于Voronoi图的语义位置点生成方法

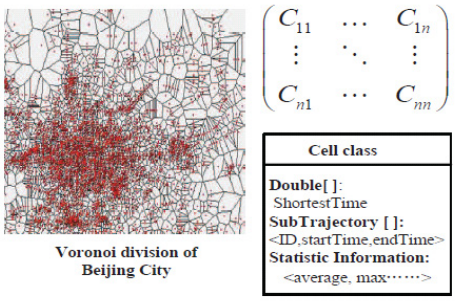

Fig.2 Voronoi based traffic information graph

图 2 Voronio 划分及语义交通信息图

每个大型城市都包含成百上千的地理位置点,居民难以全部记忆.通常,使用最多的地理位置描述方式是“交通网络”和“重要位置”表达方式.比如,“北京市海淀西路 163 号”,“奥体公园南门东 400 米”.幸运的是,城市规划部门通常根据人口分布、空间距离、路网情况,确定若干公共交通站点,作为城市轨道、公交车辆的停靠点.这些点也成为了人们最为熟悉的语义标识.这些停靠点的科学设定,可以保证对于城市中任何一个重要地理位置,总存在一个停靠点,从这个停靠点到达该重要位置点的时间小于某个阈值,比如 10 分钟.

综上所述,本文基于城市的公共交通停靠点,建立城市空间语义图模型.如图 2 所示,基于公交停靠点,对整个城区进行 Voronoi 图划分.每个划分的 cell 赋予一个固定的 ID,并且以该停靠点的名字命名,以方便人们的记忆.那么,一个划分 cell 生成一个语义位置点,具有固定的 ID 和名字.Voronoi 的特性可以保证,任何一个地理位置点都可以找到一个最近的公交站点(语义位置点).停靠站点数据可以在交通部门的网站上获得,Voronoi 的空间划分算法采用经典的平面扫描法[7].

本文以任意一个停靠点分别作为起点和终点建立语义交通信息图,该图以邻接矩阵的方式保存.矩阵的一个元素描述任意两个语义位置点间的交通信息统计数据.具体的交通统计信息和统计算法由系统管理员根据需求确定.比如,1 周 7 天、每天 2 个小时为一个单位建立的最短行驶时间和最优行驶路径.在本文的演示系统中,由于只有 7 天的出租车行驶数据.所以,只是统计工作日(周一到周五)的交通活跃时间段(6:00~22:00)和休息日(周六,周日)活跃时间段任意两个语义位置点间最短行驶时间和最短行驶路径.

3.2 基于轨迹经过模型的信息图生成算法

将整个城市的空间进行划分之后,开始每两个 cell 之间的交通数据统计信息.在传统的 GIS 系统中,可以通过交通图的最短路径搜索算法计算最短路径,并按照相距的欧几里德距离估计大概行驶时间.然而,在城市中,

由于受交通管制、单行道、车流状况等因素的影响,物理距离越近的路径不一定行驶时间越短.同样,由于场景的复杂性,不存在一种时间估计模型能够较准确地计算两点之间的时间开销最短路径.

Algorithm 1. Voronoi-Based time matrix building.

Input: Trajectory: *traj*; Voronoi-Map: *map*;

Output: *Time-Matrix*[*CellNum*][*CellNum*]: *matrix*.

```
1.  Point p=traj.next();   //Fetch a new GPS point
2.  SemiList tmpList=new SemiList();
3.  while (P!=null) do
4.     SemiPoint S_n=M.semi(P);   //Get the semi-point based on physical location
5.     if (S_n is different from temList.last()) then
6.        for each SemiPoint S_i∈temList do
7.           Matrix[S_i.ID][S_n.ID]=StatisticFuction();
8.        end for
9.        tmpList.add(semiPoint);
10.    end if
11.    p=traj.next();
12. end while
13. return matrix;
```

大量的历史 GPS 轨迹详细记录了车辆在任意时间段的行驶情况,每个 GPS 点既包括经纬度坐标,也包含在轨迹上任意一点的时间数据.通过这些信息,我们可以采取统计学方法计算出每两个位置点的行驶轨迹和行驶时间.为了与空间划分模型相融合,我们采用每个 GPS 点所在的 Voronoi 的 cell-ID 作为其语义点(*SemiPoint*).该种方法可以将百万级的用户兴趣点关联到几万个公交停靠点,大大降低了存储和计算开销.而带来的误差不会超过汽车穿行一个 Voronoi 图的 cell 的时间(北京市两个公交站点间的行驶时间不会超过 10 分钟),这种 10 分钟之内的误差对于行程规划这种对时间精度要求不高的应用而言是可以忍受的.现实的困难是,并不是任何两个空间 cell 之间都有足够多的行程(一个 Trip,表示从一个出发点到一个目的地的轨迹)以统计任何两个 cell 之间的行驶时间.因此,本文提出将行程路过的区域,即使不是起始点和终点,也作为统计的数据源.比如,一个 Trip 为 A_s-A_e.如果将其路过的区域也进行描述,则表示为

$$\langle A_s, A_1, A_2, A_3, ..., A_n, A_e \rangle.$$

如算法 1 所示,当需要统计 A_1 到 A_3 两个 cell 之间的行驶数据时,该行程 A_s-A_e 的子路径 A_1-A_2-A_3 也作为一个统计数据源.尤其是进行最短路径统计时,这种方式可以发现隐藏在较长行程中的快捷路径.如算法 1 中第 5 行所示,当车辆驶入一个新的 cell 时,算法遍历该行程中所有的历史语义位置点,保存每个历史语义位置点和新语义点的行驶时间,并进行统计信息更新(第 6 行~第 8 行).

4 基于类别的泛化行程推荐算法

4.1 位置点的层次结构描述

位置点的分类方法有很多,为了符合应用需求,本文采用当前最大的基于位置服务的在线网站 Foursquare 的位置点分类方法[4].图 3 展示了部分目录的树形结构.当用户给出希望访问的位置类别时,属于该类别的位置点和该类所属子类的位置点均在推荐的范围之内.

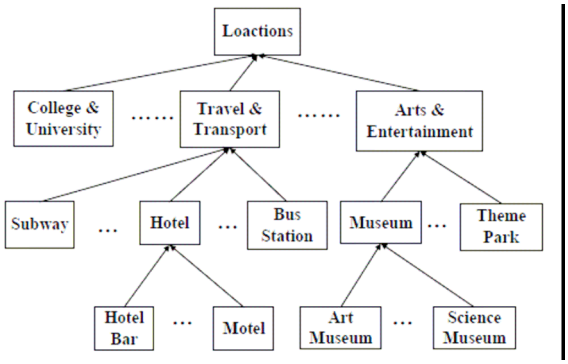

Fig.3 Hierarchy of the location points

图 3 位置点的类别层次

4.2 基准算法:基于遍历的单目标优化

本节给出采用最短时间开销作为开销函数的行程推荐算法.根据用户的需求可以设定不同的开销函数.基准算法为基本的三阶段推荐算法,该算法通过枚举每个实例可选的全部位置点,遍历所有的可行行程,并找出最优的一条作为推荐.第 1 步,为每个活动实例推荐 k 个最热门的地理位置点;第 2 步,从每个实例的 k 个位置点中各取出一个位置点,并通过查询语义交通信息图获取邻接两点的 *path*,从而生成一个行程推荐条目;第 3 步,根据候选条目的开销,对整个候选集由低到高排序,生成完整的推荐列表.

基准算法简单、直观,但也存在如下不足:

组合爆炸问题:当 k 的值较大时,备选的位置点较多,采取基准算法,需要遍历所有的组合及 k 的 dim 次方.所需的计算时延较大,当 dim 为 6,k 为 10 时,所需的计算时延即为 2 328ms,不能满足在线服务的时延要求.

单目标的过度优化问题:当对最短路径进行优化,且 k 较大,备选的地理位置点较多时,所得的推荐结果即变成了在地理空间汇集严重的位置点,而损失了位置点的流行程度,即热门程度.

综上,对于行程推荐问题,属于多目标优化问题.同时,该问题对最优解的要求不够严格,近似最优解即可满足需求.因此,我们需要一种近似最优解策略,能够快速得到多目标最优化近似解.

4.3 基于蚁群算法的多目标行程规划

多目标优化问题是在科学与工程中普遍存在的一类优化问题.一般情况下,可认为是需要权衡一个总目标下的多个目标的优化来达到总目标最优,所以需要考虑各个子目标的权重,并且具有高维度、大尺度的特点,所以优化困难,而传统优化手段对目标函数形式要求较为苛刻.比如,基准的遍历算法可以将最优化排序函数转化为多个目标的加权平均.但是,在本文的问题中,我们并不能准确地设计权重值,以达到目标最优.比如,交通时延和景点的热门程度很难一起融合.

幸运的是,蚁群算法、模拟退火算法、遗传算法、神经网络算法等智能算法已经有了较多的应用场景.本文将改进基本的蚁群算法,用以解决行程推荐问题,主要解决子目标加权总目标的最优求解.蚁群算法的长处还在于可以根据系统负载情况调整蚁群的规模和运行次数,从而提升系统的整体服务质量.

4.3.1 蚁群系统模型建立

蚁群算法(ant colony algorithm)是一种仿生模拟进化算法[13].ACS 的思想是模拟蚂蚁寻食行为,即使用大量 Ant 在搜索空间中随机搜索,并且用信息素 Pher0Inone 来加强搜索路线,引导其他 Ant 的搜索,同时引入信息素的挥发(evaporation)机制来避免陷入局部最优,这种引入挥发机制的正反馈方式使得该算法能够找到全局的多个最优解,而不会像其他搜索算法那样很快陷入局部最优解,并且因其本身的并行性,能够方便地实现并行计算.本文将蚁群算法移植到行程推荐领域,并作如下修改:保持全局优化目标函数不变,依旧寻找路上最短路径;引入启发式规则,将旅游景点的热门程度融入到蚂蚁的信息素更新步骤,使得蚂蚁倾向于选择热门度较高的位

置点.

一个行程规划表述为 n 个不相交集合中元素的有序排列(*Path*).

$$Path=\langle S_1,S_2,...,S_n\rangle.$$

位置点 p_{ij} 表示第 i 个集合中的第 j 个点,假设每个集合中元素个数均为 k.每两个相邻的子集的元素之间均存在一条边,连接两个位置点:

$$E(p_{ij},p_{(i+1)l}).$$

为了描述蚁群系统,引入如下标号:

$A^t(k)$:第 k 个蚂蚁在第 t 次行程中的状态,蚂蚁的总数为 m;

$D(p_{ij},p_{(i+1)l})$:两个位置点之间的最短时延;

$I^t(p_{ij},p_{(i+1)l})$:t 次循环中两个位置点之间边上的信息素强度;

$P_k^t(p_{ij},p_{(i+1)l})$:表示 t 次循环,第 k 个蚂蚁从 p_{ij} 转移到 $p_{(i+1)l}$ 的概率;

$H(p_{ij})$:表示位置点 p_{ij} 的热门程度.

4.3.2 基于蚂蚁算法的规划

蚂蚁算法的执行流程:

(1) 初始化,将每个蚂蚁随机地放置到第 1 集合的位置点上(算法 2 的第 1 行~第 4 行);

(2) 蚂蚁在 n 个集合之间按顺序行走,蚂蚁在两个城市之间的转移概率为(算法 2 的第 6 行~第 10 行)

$$P_k^t(p_{ij},p_{(i+1)l})=\frac{I^t(p_{ij},p_{(i+1)l})\times H(p_{(i+1)l})}{D(p_{ij},p_{(i+1)l})} \tag{1}$$

(3) 每当所有的蚂蚁都完成一次旅行时,计算每条路径的总开销,保存最短的路径.同时评价当前状态,评估是否满足结束条件.如果满足结束条件,则返回最佳行程.如果不满足,则更新每个边的信息素.当蚂蚁完成一次行程之后,各个边的信息量改变值如下(算法 2 的第 11 行~第 18 行)所示:

$$\Delta I_k^t(p_{ij},p_{(i+1)l})=\begin{cases}\dfrac{Q}{L^t{}_k}, & \text{if } A^k(t)\ cover\ E(p_{ij},p_{(i+1)l})\\ 0, & \text{else}\end{cases},$$

其中,$L^t{}_k$ 表示在本次行程中蚂蚁的总时延.

$$I^{t+1}(p_{ij},p_{(i+1)l})=I^t(p_{ij},p_{(i+1)l})\times(1-\alpha)+\sum_{k=1}^{m}\Delta I_k^t(p_{ij},p_{(i+1)l}) \tag{2}$$

(4) 重新放置蚂蚁,开始下一个行程周期(算法 2 的第 19 行).

根据如上流程,描述蚂蚁算法的伪代码如下:

Algorithm 2. Ant colony based activity scheduling.

```
1.  for all edge, ant do
2.    I^1(P_ij,P_(i+1)l)=Iinitial;
3.    Place ant on a randomly choose POI of S_1;
4.  end for
5.  Let Path_minDelay be the best path and L_min its delay;
6.  for t=1 to t_max do
7.    for k=1 to m do
8.      Build tour  Path_k^t  by applying n−1 times the following step;
9.      Choose next POI with probability computed by Formula(1);
10.   end if
11.   for k=1 to m do
```

12. Compute the delay L_k^t of $Path_k^t$ produced by ant k;
13. end for
14. if an faster path found then
15. Update the $Path_{\min Delay}$ and $L_{\min}$;
16. end if
17. for all edge do
18. Update the $I(t+1)(P_{ij},P_{(i+1)l})$ by Formula(3);
19. Replace ant on a randomly choose POI of S_1;
20. end for
21. end for
22. Return the $Path_{\min Delay}$ and $L_{\min}$;

5 演示系统及算法性能评测

演示系统的实验城市为北京市,在线社交数据来自 Foursquare,该网络注册用户已经超过 2 000 万,是最大的基于位置的社交网络.

(1) 公共交通数据:公交停站点数据来自北京公交网,站点共计 10 684 个,涵盖了全部市内及郊区线路.

(2) Foursquare 数据采集:调用 Foursquare 的 API,采集每个 Voronoi 划分 cell 中的热门地理位置点(包括该节点总“签到”次数和历史到访的游客数、用户提交的评论信息、优惠活动信息等).之后对全部 cell 的热门数据进行融合、去重.实验中有效的热门地理位置点为 30 784 个.为了提升分类信息的查找速度,按照类别对全部位置点建立了倒排索引[14].

(3) 语义交通信息图构建数据:本文采用微软亚洲研究院的 T-drive 数据,包含北京市在 2008-02-02 到 2008-02-08 一周时间内 10 357 辆出租车的行驶轨迹.

推荐结果及交互界面

表 1 列举了根据示例 1 的需求,分别采用基准算法和蚁群算法推荐的结果(每种算法采用了 k=3 和 k=6 两种参数设置).用户 UI 如图 4 所示,该推荐结果给出了推荐位置点的热门程度描述(括号内为“总签到”数),用户也可以点击这些位置点,从 Foursqure 网站获取用户的评价信息.同时,推荐结果给出了每两个位置点间的最小驾车时间开销(两个位置点之间的分钟数).

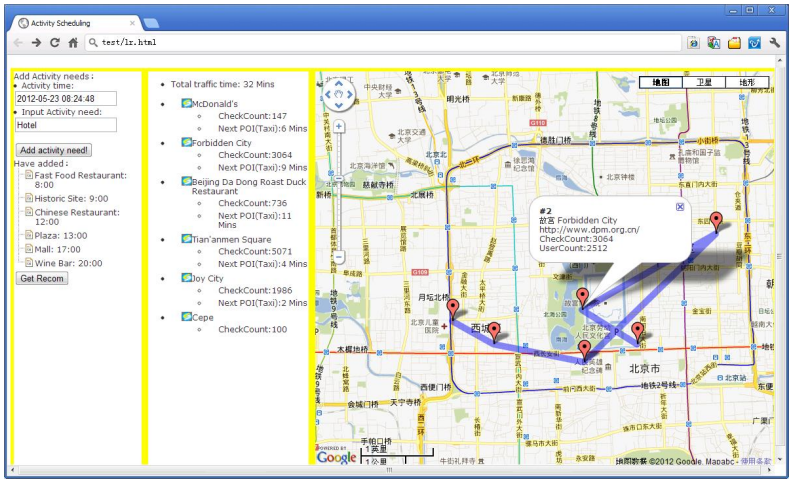

Fig.4 User interface

图 4 用户界面

表 1 中的用户评价值为 14 个用户给出的主观评价(满分为 10 分).由评价结果可知,对于示例 1 的推荐评价分数均在 7 以上,因此推荐结果具有较好的用户认可度.同时,比较 4 个结果,可以发现蚁群算法的推荐结果,虽然路上消耗的时间较长,但是更能够得到用户的认同.值得注意的是,当有多个候选位置点加入推荐时,推荐的结果评分反而会降低,这表明用户对景点的著名程度更加在意.

Table 1 Itinerary recommendations for Example 1 in Beijing City

表 1 北京市示例 1 的推荐结果

Category	推荐行程	最短行驶时间	用户评价
k=3(组合方法)	McDonald's 麦当劳(143)→6(分钟)→故宫 Forbidden City(2914)→9(分钟)→Beijing Da Dong Roast Duck Restaurant 北京大董烤鸭店(720)→ (分钟)→Jianwai SOHO(277)→4(分钟)→Sanlitun Village 三里屯 Village(5688)→5(分钟)→CJW The Place(167)	Time=25 分钟	*Ave.Score*=8.1
k=6(组合方法)	McDonald's 麦当劳(143)→3(分钟)→天坛东门 East Gate: Temple of Heaven(193)→3(分钟)→Duck de Chine 全鸭季(287)→3(分钟)→Jianwai SOHO(277)→2(分钟)→秀水街 Silk Street Market(1017)→0(分钟)→CJW The Place(167)	Time=11 分钟	*Ave.Score*=7.4
k=3(蚁群算法)	McDonald's 麦当劳(143)→6(分钟)→故宫 Forbidden City(2914)→9(分钟)→Beijing Da Dong Roast Duck Restaurant 北京大董烤鸭店(720)→11(分钟)→天安门广场 Tian'anmen Square(4908)→4(分钟)→大悦城 Joy City(1933)→2(分钟)→Cepe(94)	Time=32 分钟	*Ave.Score*=8.9
k=6(蚁群算法)	McDonald's 麦当劳(143)→6(分钟)→故宫 Forbidden City(2914)→9(分钟)→Beijing Da Dong Roast Duck Restaurant 北京大董烤鸭店(720)→5(分钟)→Jianwai SOHO(277)→0(分钟)→The Place 世贸天阶(1798)→5(分钟)→Enoterra(338)	Time=25 分钟	*Ave.Score*=8.2

6 相关工作

微软亚洲研究院的 GeoLife 工程,通过分析在 2008 年~2010 年采集的 165 个用户的轨迹数据,开发了出行导航系统[8].并且,通过 GPS 轨迹分析人类经常访问的地理位置、地理位置之间的关联关系等信息,从而回答:"北京最热门的旅游景点有那些?","人们去了故宫之后,还会去访问哪些地方?","最常用的出行路线有哪些?"等问题.但是,该工作并没有实现面向多个目标的行程推荐.

文献[9]通过对轨迹数据记录的挖掘,分析用户生活模式.该工作将社会网络中的用户和具体的空间位置关联起来,形成了空间社会网络图.该工作的主要贡献体现在,提出了进行社会网络和空间网络联合查询的一系列操作符,并在图数据库和关系数据库上进行了实现.该工作启发了本文作者联合使用空间网络和社交网络探索新的应用和技术.

空间关键字查询的经典场景是:给定一个物理位置和一组关键字,找出单独的一个或者 k 个与输入关键字最匹配的对象[10,11].文献[1]拓展了空间关键字查询,提出并实现了寻找一组满足给定关键字的对象,且这些对象间的空间距离最小,该种查询的应用场景局限性较强,不能充分利用各种背景知识.而本文的工作可以发挥集体智能,找出更加人性化的位置点,并给出合理的出行路径.

文献[12]的工作支持检索距离 k 个位置点最近的轨迹,该工作需要用户准确地给出 k 个位置点,然后才能获取经过 k 个位置点的历史轨迹,不能实现泛化的查询,也不能保证获取的轨迹开销最小.

7 总 结

本文通过联合社交网络数据和物理轨迹数据,利用地理位置点的类别层次推理,实现了泛化的行程推荐功能.推荐算法基于历史轨迹隐含的语义交通信息,给出多个行程点之间的最优路径,较之最短路径计算方法更加准确.同时,推荐结果可以展示社交网络上关于推荐位置点的评价信息,从而帮助用户进一步了解行程的好坏.推荐结果对于用户确定详细行程具有较高的借鉴价值.

References:

[1] Cao X, Cong G, Jensen CS, Ooi BC. Collective spatial keyword querying. In: Proc. of the SIGMOD 2011. 2011.

[2] Yuan J, Zheng Y, Xie X, Sun GZ. Driving with knowledge from the physical world. In: Proc. of the KDD 2011. New York, 2011.

[3] Yuan J, Zheng Y, Zhang CY, Xie WL, Xie X, Sun GZ, Huang Y. T-Drive: Driving directions based on taxi trajectories. In: Proc. of the GIS 2010. New York, 2010.

[4] Foursquare websit. 2012. http://www.foursquare.com

[5] JiePang net websit. 2012. http://www.jiepang.com

[6] Yan ZX, Chakraborty D, Parent C, Spaccapietra S, Aberer K. SeMiTri: A framework for semantic annotation of heterogeneous trajectories. In: Proc. of the EDBT 2011. 2011. 259−270.

[7] de Berg M, Cheong O, van Kreveld M, Overmars M. Computational Geometry Algorithms and Applications. 3rd ed., Springer-Verlag, 2009.

[8] Zheng Y, Chen YK, Xie X, Ma WY. GeoLife2.0: A location-based social networking service. In: Proc. of the Mobile Data Management—MDM. 2009. 357−358.

[9] Doytsher Y, Galon B, Kanza Y. Querying geo-social data by bridging spatial networks and social networks. In: Proc. of the LBSN 2010. 2010.

[10] Cao X, Cong G, Jensen CS. Retrieving top-*k* prestige-based relevant spatial Web objects. PVLDB, 2010,3(1):373−384.

[11] De Felipe I, Hristidis V, Rishe N. Keyword search on spatial databases. In: Proc. of the ICDE. 2008. 656−665.

[12] Tang LA, Zheng Y, Xie X, Yuan J, Yu X, Han JW. Retrieving *k*-nearest neighboring trajectories by a set of point locations. Advances in Spatial and Temporal Databases, 2011,6849:223−241.

[13] Dorigo M, Gambardella LM. Ant colony system: A cooperative learning approach to the traveling salesman problem. IEEE Trans. on Evolutionary Computation, 1997,1(1).

[14] Patil M, Thankachan SV, Shah R. Inverted indexes for phrases and strings. In: Proc. of the Sigir 2011. 2011.

孟祥旭(1982－),男,河北唐山人,博士生,主要研究领域为无线网络,基于位置的服务系统.

王晓东(1973－),男,博士,研究员,CCF 高级会员,主要研究领域为移动计算,数据库系统,无线网络安全.

周兴铭(1938－),男,教授,博士生导师,CCF 高级会员,主要研究领域为高性能计算,移动计算.