

## 基于偏好的个性化路网匹配算法\*

高 需<sup>1,2</sup>, 武延军<sup>2</sup>, 郭黎敏<sup>3</sup>, 丁治明<sup>3</sup>, 陈军成<sup>3</sup>

<sup>1</sup>(中国科学院大学, 北京 100049)

<sup>2</sup>(中国科学院 软件研究所 协同创新中心, 北京 100190)

<sup>3</sup>(北京工业大学 计算机学院, 北京 100124)

通讯作者: 高需, E-mail: gaouxu@nfs.iscas.ac.cn



**摘 要:** 定位技术的普遍应用,使得随时随地获取个人位置成为可能,进一步推动了基于位置的服务等新型应用的发展,产生了海量轨迹数据.精确的路网匹配对提高这些新型应用的服务质量具有重要的研究意义,然而受众多因素的影响,大部分轨迹的采样率较低,比如由签到类应用或低功耗设备生成的低采样轨迹,给路网匹配带来了巨大的挑战.研究基于偏好的个性化路网匹配(driving preference based personalized map-matching,简称DPMM),提出了在动态道路网络中的用户驾驶偏好模型.基于该模型,提出了两阶段路网匹配算法:局部匹配搜索用户最可能采用的几条局部 Skyline 路径;设计了全局匹配的动态规划算法,该算法返回在用户驾驶偏好下最可能的多条全局路径作为最终匹配结果.实验结果充分表明,该方法是有用的和高效的,具有一定的使用价值.

**关键词:** 时空数据;轨迹;路网匹配;多目标优化;Skyline 路径;动态规划

**中图法分类号:** TP181

中文引用格式: 高需,武延军,郭黎敏,丁治明,陈军成.基于偏好的个性化路网匹配算法.软件学报,2018,29(11):3500–3516.  
<http://www.jos.org.cn/1000-9825/5297.htm>

英文引用格式: Gao X, Wu YJ, Guo LM, Ding ZM, Chen JC. Personalized map-matching algorithm based on driving preference. Ruan Jian Xue Bao/Journal of Software, 2018, 29(11): 3500–3516 (in Chinese). <http://www.jos.org.cn/1000-9825/5297.htm>

### Personalized Map-Matching Algorithm Based on Driving Preference

GAO Xu<sup>1,2</sup>, WU Yan-Jun<sup>2</sup>, GUO Li-Min<sup>3</sup>, DING Zhi-Ming<sup>3</sup>, CHEN Jun-Cheng<sup>3</sup>

<sup>1</sup>(University of Chinese Academy of Sciences, Beijing 100049, China)

<sup>2</sup>(Collaborative Innovation Center, Institute of Software, The Chinese Academy of Sciences, Beijing 100190, China)

<sup>3</sup>(School of Computer, Beijing University of Technology, Beijing 100124, China)

**Abstract:** With the increasing proliferation of position technologies, there comes huge volumes of trajectory data, which are used in many modern applications such as path planning and location based services. Accurate road network matching can improve the service quality of these new applications. However, the low sampling trajectories bring a major challenge for map-matching. This paper studies the problem of matching individual law-sampling trajectory to a dynamic multi-criteria road network based on user's driving preferences. First, a driving preference model in the dynamic road traffic network is proposed. Based on this model, a two-stage map-matching algorithm is developed. While local matching searches multiple local likely Skyline paths, a global matching dynamic programming algorithm is designed and the most probable  $k$  global paths are selected as the matching result. Experiments show that the proposed method is effective and efficient.

\* 基金项目: 国家自然科学基金(61402449, 91546111); 中国科学院战略性科技先导专项课题(XDA06010600); 北京市教委重点项目(KZ201610005009)

Foundation item: National Natural Science Foundation of China (61402449, 91546111); Strategic Priority Research Program of the Chinese Academy of Sciences (XDA06010600); Key Project of Beijing Municipal Education Commission (KZ201610005009)

收稿时间: 2017-01-25; 修改时间: 2017-03-12; 采用时间: 2017-04-12; jos 在线出版时间: 2018-04-16

CNKI 网络优先出版: 2018-04-16 10:59:45, <http://kns.cnki.net/kcms/detail/11.2560.TP.20180416.1059.004.html>

**Key words:** spatio-temporal data; trajectory; map-matching; multi-objective optimization; Skyline path; dynamic programming

搭载有定位模块(例如北斗、GPS、GSM、WIFI 和蓝牙等)的智能终端不断普及,使得随时随地获得个人位置信息成为可能,从而产生了轨迹大数据,并在基于位置的服务中得到了广泛应用,例如现代交通管理、智能导航、位置推荐等.路网匹配(即将轨迹配准到道路网络)是此类应用不可缺少的关键模块,而精确的路网匹配对提高服务质量起到了至关重要的作用.

然而,受建筑物遮挡、定位模块错误、低能耗的模块设计、多模式和多样化精度的定位技术等因素的影响,大部分轨迹的采样率较低.社交网络的签到和其他位置共享服务(例如 Weibo、Facebook、Twitter 等)也仅能提供低采样的轨迹.相似的低采样轨迹也可能来自具有位置标签的图片(例如 Flickr、Panoramio)和稀疏的道路监控获取的车辆快照,此类位置序列也形成了低采样轨迹.低采样轨迹的两个相邻采样点分别匹配到道路网络中不相邻的边,补全轨迹实际经过的所有中间缺失边而构成完整的路径,是本文研究的主要内容.

如图 1 所示, $p_1$  和  $p_2$  是轨迹中的两个相邻采样点,分别匹配到路网中不相邻的边  $e_1$  和  $e_{13}$ ,然而难以确定轨迹实际采用的真实路径.可能的路径是  $P_1 \sim P_4$  中的某一条,查找出具体的路径称为路网匹配或路径推理.

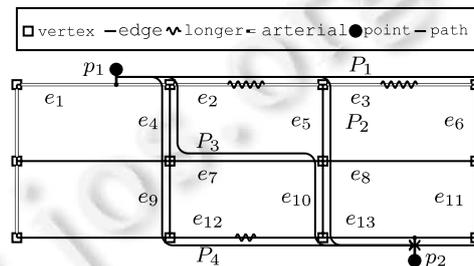


Fig.1 Low-Sampling trajectory and its possible paths

图 1 低采样轨迹及其可能路径

已有路网匹配方面的研究大多仅考虑单属性道路网络,比如仅考虑旅行距离<sup>[1]</sup>、旅行时间<sup>[2,3]</sup>或路径频繁度<sup>[4]</sup>.然而在实际的路径规划中存在着多重属性,不再是单一属性的最短路径、最快路径、最频繁路径或最省油路径,而是多个属性的权衡(比如以权重将多属性线性组合为单一属性且组合后的属性代价最优,或选择至少在某个属性上比其他所有路径优的路径).例如,在旅行距离和旅行时间双属性道路网络中,一个用户的旅行距离和旅行时间权重分别为 0.3 和 0.7,这表明该用户喜爱较短的旅行时间;而另一个用户的旅行距离和旅行时间权重分别为 0.8 和 0.2,表明他更喜爱较短的旅行距离.这两个用户选择了不同路径,但产生同样的低采样轨迹,那么该轨迹将匹配到不同的路径.然而,已有工作由于忽略了轨迹产生者之间的个性化差异,最终把属于两个用户的相似轨迹匹配到相同路径,而产生较大的误差.为了解决该问题,本文将构建多属性路网中的用户偏好模型,从而体现用户在路径规划时的个性化差异.

如图 1 所示,路径  $P_1 \sim P_4$  都有可能是连接低采样轨迹相邻采样点  $p_1$  和  $p_2$  匹配到的边  $e_1$  和  $e_{13}$ ,从而构成轨迹的完整路径.然而,作为旅行距离和旅行时间的均衡路径  $P_2$  与最快路径  $P_1$ 、最短距离路径  $P_3$  和最省油路径  $P_4$  差异较大,不同用户可能选择不同路径.已有研究表明,来自不同用户的相似的低采样轨迹,根据路网匹配考虑的属性代价区别,最终可能匹配到的路径有所不同<sup>[1-4]</sup>.

一般来讲,在多属性路网中的每条路径都对应一个多属性的综合权重,如图 1 所示路径  $P_2$  在双属性路网中(即旅行距离和旅行时间),两者的权重分别为 0.5 和 0.5.大部分用户在路径规划时不再选取单一属性最优<sup>[5]</sup>,而是在多属性之间权衡,得到 Skyline 或 Pareto 路径.若用户知道了基本的路网代价,就倾向于不去选择所有旅行代价都比其他路径要弱的路径(非优化路径).

本文提出在动态多属性道路网络中,从用户所有历史高采样轨迹中学习出用户驾驶偏好,并以多目标优化或数据分析中普遍使用的 Skyline 或 Pareto 技术指导路网匹配,解决了现有路网匹配算法忽略用户个性化

驾驶行为的不足.

本文研究了基于偏好的个性化路网匹配问题,主要贡献如下所示.

- 驾驶偏好模型.根据以往路径规划数据构造了细粒度的驾驶偏好概率模型.
- 多目标优化.在低采样轨迹路网匹配中引入多目标优化方法,即局部匹配为某条 Skyline 路径.
- 路网匹配精度有大幅度提升.我们在真实数据集上做了充分的实验,实验结果表明,所提出的方法比已有方法在相邻采样点采样时间间隔为 20min 时,保证路网匹配精度不低于 65%.

本文第 1 节介绍相关工作.第 2 节给出问题描述.第 3 节引入细粒度驾驶偏好模型.第 4 节提出两阶段路网匹配算法:局部匹配和全局匹配.第 5 节描述在真实数据集上的实验.第 6 节总结全文.文中用到的主要符号见表 1.

Table 1 A list of notations

表 1 符号列表

名称	解释
$f_e^i(t)$	在 $t$ 时刻,从边 $e$ 的起点出发到达 $e$ 的终点耗费的第 $i$ 个属性代价
$m$	道路网络中边的属性或代价的维度
$\mathcal{Y}, \mathcal{Y}[i], Y$	轨迹,轨迹中第 $i$ 个采样点和轨迹集合
$P, P_y, \mathcal{P}, \mathcal{P}_{(s,d)}$	路径, $\mathcal{Y}$ 的路网匹配路径,路径集合,从 $s$ 出发到达 $d$ 的所有路径集合
$P=P_i \diamond P_{i+1}$ 或 $P_i P_{i+1}$	路径 $P_i$ 和 $P_{i+1}$ 尾、首相接构成的路径 $P$
$l, e, t_a, e, t_l$	路径中边的数量,到达和离开边 $e$ 的时间戳
$I(P), \gamma, F$	路径 $P$ 的偏好向量,第 $i$ 个路径偏好向量分量,驾驶偏好向量集合
$S^i(P)$	路径 $P$ 在第 $i$ 个路网属性上的旅行代价
$pdf, \mathcal{AF}(P, pdf)$	某用户的驾驶偏好密度函数,路径 $P$ 在驾驶偏好 $pdf$ 下的适配度函数

## 1 相关工作

路网匹配算法按照是否需要其他轨迹协助,分为无依赖的单轨迹路网匹配和有依赖的多轨迹协同路网匹配.单轨迹路网匹配算法又可分为基于几何结构<sup>[6,7]</sup>、网络拓扑结构<sup>[6-10]</sup>、统计<sup>[1,10-13]</sup>的路网匹配算法.其中,基于几何结构的路网匹配主要考虑轨迹采样点或采样序列与道路网络中边的几何形状之间的关系,如 Point-to-Point 路网匹配算法<sup>[6]</sup>、Point-to-Curve 路网匹配算法<sup>[6]</sup>和 Curve-to-Curve 路网匹配算法等<sup>[6,14,15]</sup>;基于网络拓扑结构的路网匹配算法综合了道路网络中边的连接性、旅行时间和旅行距离等因素<sup>[7-9]</sup>.把基于几何的方法和基于网络拓扑的方法结合起来设计新的路网匹配算法<sup>[6,7]</sup>.在基于统计的路网匹配算法中,首先要计算每个采样点的候选匹配边或顶点集合,并为每个匹配赋予一个和采样点与匹配元素间欧式距离或路网拓扑旅行代价相关的权重,代表算法有 ST-matching<sup>[11]</sup>和 IVMM<sup>[10]</sup>.然而这两种算法对于过低采样的轨迹路网匹配效果较差,在时间复杂度方面,假设低采样轨迹 GPS 长度为  $n$ ,匹配用路网有  $m$  条边,每个采样点相邻的候选边数量为  $k$ ,则 ST-matching 和 IVMM 的时间复杂度都为  $O(nk^2 m \log m + nk^2)$ .认为低采样轨迹是在真实路径中有噪音的稀疏观察点,并采用隐马尔可夫模型设计路网匹配算法,这些算法以高斯模型描述单一观察点、以观察点与路网中边的几何特征和路网的拓扑结构确定观察点之间的转移概率,最后匹配的路径具有最大的后验概率<sup>[1,12,13]</sup>.

与本文最相关的工作是数据驱动的低采样轨迹路网匹配算法,这类方法也称为多轨迹协同路网匹配.这种路网匹配方法的优势在于用户的轨迹往往遵循较高的时间和空间旅行规律<sup>[16]</sup>.HRIS 以频繁路径体现时空旅行规律,提出了数据驱动的两阶段路径推理系统:第 1 阶段为局部推理,该阶段计算低采样轨迹任意相邻两点之间的多条可能路径;第 2 阶段为全局推理,该阶段使用动态规划算法从局部匹配结果中组合并计算出最可能的路径作为路网匹配结果.InferTra<sup>[2]</sup>是一个基于图的轨迹推理系统,该推理系统使用吉布斯抽样算法和学习到的网络便携模型推理出轨迹的完整路径.STRS<sup>[3]</sup>是最新的路径恢复方法,该方法同时考虑历史轨迹的时间和空间概率分布模型,相比于其他低采样轨迹的路网匹配算法,STRS 的准确率有较大提升.以轨迹聚类体现时空旅行规律,Li 等人提出了连接多个边聚类而构成完整路径<sup>[17]</sup>,每个边聚类是路径的  $k$  条分段,每条轨迹被划分为多个 Multi-Track<sup>[18]</sup>.

另一个与本文相关的工作是驾驶偏好的构建和带有偏好的路径查询.张凯评等人给出了兼顾查询者熟悉路径的个性化路径规划框架<sup>[19]</sup>.Silva 等人提出了多目标路径规划,并以事先给出的路网静态属性重要程度来简化最终路径,不同属性的重要程度即为驾驶偏好<sup>[20]</sup>.Balteanu 等人提出了支持两个静态属性的驾驶偏好,比如行驶距离和经过的红绿灯数量<sup>[21]</sup>.杨彬等人提出了在多属性动态道路网络中的概率型驾驶偏好模型,可用于求解个性化 Skyline 路径<sup>[22]</sup>.戴健等人研究了个性化路径推荐方法,该方法采用概率型驾驶偏好模型<sup>[22]</sup>,从那些与查询者具有相似驾驶偏好的用户的历史轨迹中构建临时道路网络,在该网络上查询最频繁路径,该方法认为相似用户的轨迹具有更高的参考价值<sup>[23]</sup>.

在多目标路径规划方面,有关 Skyline 或 Pareto 路径的研究也得到了研究者的广泛关注<sup>[24-26]</sup>.Kriegel 等人提出了基于图嵌入的 Skyline 路径查询算法<sup>[27]</sup>.Shekelyan 等人为了降低 Skyline 路径数量,提出了线性 Skyline 路径问题,并给出了基于凸优化的精确算法和快速的近似算法<sup>[28]</sup>.杨彬等人设计了动态网络下的快速 Skyline 路径查询算法 Multi-Dijkstra<sup>[22,29]</sup>,该算法在多属性情况下表现良好,当属性数量较低时(比如小于 3)性能较差.

然而,现有针对低采样轨迹的路网匹配或路径推理算法都忽略了用户个性化驾驶偏好,因此,距离较远的相邻采样点,该采样点来自驾驶偏好各不相同的多个用户(即实际所选路径很可能不同),但是使用任一已有路网匹配算法,最终得到的路网匹配结果是相同的<sup>[2-5]</sup>.

当前已有用户驾驶偏好模型难以直接应用在个性化路网匹配中来:首先,在路网匹配时,难以为用户静态指定唯一的驾驶偏好,静态的双属性难以满足多属性动态路网情形<sup>[8]</sup>.在我们的问题设置中,已有驾驶偏好模型由于考虑的用户驾驶偏好粒度较粗,难以直接使用<sup>[22]</sup>.

在 Skyline 或 Pareto 最优路径方面,随着路网属性数量或两点间的距离增长,最终路径的数量呈指数方式增长,而难以直接应用到路网匹配中<sup>[27]</sup>.而线性 Skyline 路径在一定程度上大幅度降低了 Skyline 路径的数量,但极有可能也裁剪掉了用户实际选择的路径.

为了解决这些问题,本文提出了基于驾驶偏好的 Skyline 或 Pareto 路径查询算法,在多属性动态道路网络中对低采样轨迹进行局部路网匹配(第 3 节、第 4 节).

## 2 问题描述

动态道路网络(简称路网)是一个多属性有向图  $G=(V,E,F)$ ,其中顶点  $v \in V$  是路网中的交叉路口或道路端点,  $e=(s,d) \in E: V \times V$  是一条有向边或道路中路段,  $e,s$  和  $e,d$  分别是边  $e$  的起点和终点,  $F:(T \rightarrow \mathbb{R}^+)^m$  是一个由时间定义的动态函数向量集合,每个向量的维度为  $m$ .  $f_e^i(t) \in f_e \subset F(1 \leq i \leq m)$  定义了边  $e$  的第  $i$  个动态属性,该函数以时间  $t$  为参数,其值域为  $[0,+\infty)$  上的实数,表示在  $t$  时刻从  $e,s$  出发到达  $e,d$  的代价.

若  $m=3$ ,路网的属性可以是旅行距离( $f^1$ )、旅行时间( $f^2$ )和油耗( $f^3$ ).若从顶点  $e,s$  的出发时间是  $t_0$ ,则  $f_e^i(t_0)$  返回第  $i$  个旅行代价.例如,  $f_e^2(t_0)$  和  $f_e^3(t_0)$  分别是在  $t_0$  时刻从  $e,s$  出发到  $e,d$  要耗费的时间和油耗.本文采用冻结网络代价模型,即当从边的起点出发时间固定,则通过该边的代价是固定的<sup>[30]</sup>.

**定义 1(轨迹).** 给定轨迹集合  $Y$ ,轨迹  $\mathcal{Y} \in Y$  是一个位置序列  $(p_1, p_2, \dots, p_n)$ ,其中,  $n=|\mathcal{Y}|$  是采样点个数,  $p_i, t$  是经过  $p_i, loc$  的时间戳,且满足条件  $p_i, t < p_{i+1}, t(1 \leq i < n)$ ,  $\mathcal{Y}[i]$  表示第  $i$  个采样点.

当轨迹的采样时间间隔  $\Delta t = p_{i+1}, t - p_i, t(1 \leq i < n)$  超过 2min 时,称其为低采样轨迹;否则,称其为高采样轨迹.本文主要考虑低采样轨迹路网匹配问题,而高采样轨迹用来估计动态路网属性和用户的驾驶偏好(见第 3 节).

**定义 2(路径).** 给定路径集合  $\mathcal{P}$ ,路径  $P = (e_1, \dots, e_\ell) \in \mathcal{P}$  是  $G$  的一个无环有向子图,其中,  $\ell=|P|$  是  $P$  中边的数量,  $P, e_j$  是  $P$  的第  $j$  条边,  $e_j, t_a$  和  $e_j, t_l(1 \leq j < \ell)$  分别是到达和离开边  $e_j$  的时间戳.另外,任意  $1 \leq j < \ell$ ,条件  $e_j, d = e_{j+1}, s$  为真,即在路径的相邻边中,当前边的终点是下一条边的起点.  $P$  在第  $i$  个属性上旅行代价为  $S^i(P) = \sum_{j=1}^{\ell} f_{e_j}^i(e_j, t_a)$ ,  $S^i(\cdot)$  有可列可加性,即  $P = P_1 P_2$ ,则  $S^i(P) = S^i(P_1 P_2) = S^i(P_1) + S^i(P_2)$ ,其中,  $P_1$  终点是  $P_2$  起点.

通过已有路网匹配算法可将高采样轨迹  $\mathcal{Y}$  精确地匹配到道路网络<sup>[1]</sup>,  $\mathcal{Y}$  匹配后的路径记为  $P_{\mathcal{Y}}$ ,简称匹配路径.求低采样轨迹  $\mathcal{Y}$  的匹配路径  $P_{\mathcal{Y}}$  是本文研究的内容.

问题描述:给定动态多属性道路网络  $G$ ,某用户的历史轨迹集合  $Y$ ,针对某个低采样轨迹  $\gamma \in Y$ ,依据从高采样轨迹中学习到的用户驾驶偏好(第3节),我们为  $\gamma$  推荐该用户最有可能选择的  $k$  条路径.我们称这样的低采样轨迹的路网匹配为基于驾驶偏好的个性化路网匹配.

### 3 驾驶偏好模型

本节给出了从高采样轨迹  $\gamma$  的匹配路径  $P_\gamma$  中提取用户驾驶偏好的方法,结合对驾驶偏好的聚类,用混合高斯模型描述不同偏好的概率分布情况.

#### 3.1 路径偏好

在不同交通状况下,用户根据个人对路网中不同代价喜爱程度来选择路径,我们引入路径偏好来描述用户这种路径选择行为.

**定义 3(路径偏好).** 给定路径  $P = (e_1, \dots, e_\ell) \in \mathcal{P}$ ,  $\Gamma(P) = (\gamma^1, \gamma^2, \dots, \gamma^m) \in \mathcal{F}$  是路径  $P$  的偏好向量,简称路径偏好.其中,  $\gamma^i$  是从源点  $e_{1,s}$  到终点  $e_{\ell,d}$  在第  $i$  个属性上的最小旅行代价与  $P$  在第  $i$  个属性上的旅行代价比值,记为  $\Gamma(P) \cdot \gamma^i$  或  $P \cdot \gamma^i$ ,且  $\gamma^i = \min\{S^i(P') \mid P' \in \mathcal{P}_{(e_1,s,e_\ell,d)}\} / S^i(P)$ ,其中,  $S^i(P)$  返回路径  $P$  在第  $i$  个属性上的旅行代价,  $\mathcal{P}_{(e_1,s,e_\ell,d)}$  是所有从  $e_{1,s}$  到  $e_{\ell,d}$  的路径集合.

例如,路径  $P$  的起止顶点分别为  $s$  和  $d$ ,若从  $s$  到  $d$  的最小旅行距离和最小旅行时间分别为 10km 和 15min,而路径  $P$  的旅行距离和旅行时间分别为 13km 和 18min,那么  $P$  的路径偏好为(10/13,15/18).

#### 3.2 用户驾驶偏好

把来自同用户的任意高采样轨迹  $\gamma \in Y$  经路网匹配得到匹配路径,并为其构建路径偏好.这些路径偏好反映了用户以往历史的驾驶偏好,如果路径偏好的第  $i$  个向量元素  $\gamma^i$  比较大,则表明路径的第  $i$  个属性代价很接近该属性的最短路径;若偏好向量  $\gamma^i$  比较小,则表明用户不怎么关心第  $i$  个路网属性代价.因此,相似路径具有相似的路径偏好,对路径偏好聚类能够反映用户在不同交通状况下的路径选择态度.

直接用 DBSCAN 对路径偏好聚类是不可行的.例如,若路网属性为旅行时间(travel time,简称 TT)和旅行距离(travel distance,简称 TD),给定路径偏好  $\Gamma_1=(0.8,0.7)$ ,  $\Gamma_2=(0.8,0.6)$  和  $\Gamma_3=(0.7,0.8)$ ,则驾驶偏好  $\Gamma_1$  和  $\Gamma_3$  的距离比  $\Gamma_2$  和  $\Gamma_3$  的距离近,说明  $\Gamma_1$  和  $\Gamma_3$  属于同一个聚类.然而,由于  $\Gamma_1$  和  $\Gamma_2$  都偏爱短的旅行时间,应该属于同一个聚类,而  $\Gamma_3$  偏爱短的旅行距离而属于另一聚类.在图 2(a)所示的驾驶偏好中,用 DBSCAN 聚类算法<sup>[31]</sup>得到  $C_1 \sim C_4$  共 4 个聚类,其中,  $C_1$  和  $C_2$  聚类都表示与旅行时间相比用户更喜爱短的旅行距离,  $C_3$  聚类表示用户更喜爱短的旅行时间.然而在包含 11 个元素的  $C_4$  聚类中,8 个路径偏好说明用户喜爱短的旅行距离,而另外 3 个路径偏好表明用户更喜爱短的旅行时间,因此,要用层次化聚类将聚类  $C_4$  中不同的驾驶喜爱分离.否则,使用 11 个混合驾驶偏好概率替代 3 个喜爱较短旅行时间的个人驾驶偏好概率,会在后续的求解用户选择某条路径可能性时带来较大误差.而先层次化后,DBSCAN 的混合聚类提高了用户选择某条路径的概率的准确度.

鉴于此,我们将偏好向量  $\Gamma$  转换为分量非递减排序的路径偏好  $\tilde{\Gamma}$ <sup>[22]</sup>,再使用层次化聚类算法,粗略地为路径偏好聚类<sup>[32]</sup>,排序后的路径偏好向量的顺序相同,则它们属于同一个层次化聚类.例如,  $\Gamma_1 \sim \Gamma_3$  各分量经过非递减排序,得到  $\tilde{\Gamma}_1$  和  $\tilde{\Gamma}_2$  都为  $(TT,TD)$ ,而  $\tilde{\Gamma}_3$  是  $(TD,TT)$ ,因此,根据驾驶偏好的聚类需求,  $\tilde{\Gamma}_1$  和  $\tilde{\Gamma}_2$  属于同一个层次化聚类,而  $\tilde{\Gamma}_3$  属于另一个层次化聚类.如图 2(b)所示,位于斜线  $L$  右下侧驾驶偏好表明,用户更喜爱较短旅行时间形成一个层次化聚类(  $\tilde{\Gamma}_1$  和  $\tilde{\Gamma}_2$  落在该区域);而  $L$  左上侧驾驶偏好表明,用户更喜爱较短的旅行距离形成另外一个层次化聚类(  $\tilde{\Gamma}_3$  落在该区域).

然而,在层次化聚类结果中<sup>[22,32]</sup>,同聚类的不同路径偏好之间可能存在着巨大差异.例如,分别在旅行时间和旅行距离属性下,驾驶偏好  $\Gamma_1=(0.2,0.9)$ ,  $\Gamma_2=(0.8,0.9)$  和  $\Gamma_3=(0.7,0.8)$ ,虽然这 3 个驾驶偏好表明用户喜爱更短的旅行距离(如图 2(b)斜线  $L$  左上侧区域),由于  $\Gamma_1$  代表的路径在旅行时间上非常糟糕,很少有用户选择这样的路径.而  $\Gamma_2$  和  $\Gamma_3$  表明,用户喜爱较短的旅行距离,且同时兼顾旅行时间的情况,比如旅行时间不超过最快路径旅行

时间的 1.2 倍,如图 2(b)所示,经过层次化聚类算法得到由斜线  $L$  划分的两个层次化聚类( $C_1, C_2$  和  $C_4$  属于一个层次化聚类,而  $C_3$  和  $C_5$  属于另一个层次化聚类),在同一层次化聚类中,有必要经过 DBSCAN 算法对同一层次化聚类进一步聚类,从而得到  $C_1 \sim C_5$  共 5 个驾驶偏好聚类,其中,  $C_4$  有 8 个元素,而  $C_5$  有 3 个元素。

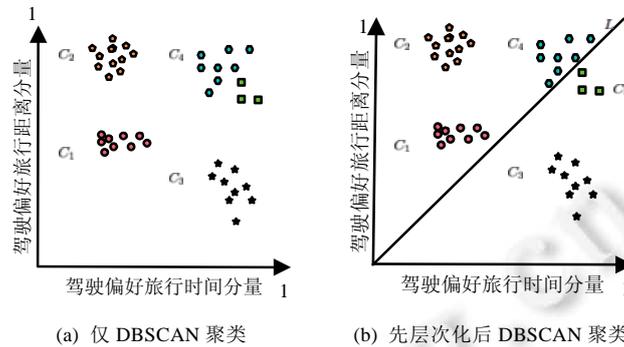


Fig.2 Hybrid hierachy-DBSCAN vs. DBSCAN clustering for driving preference

图 2 驾驶偏好的层次化与 DBSCAN 混合聚类和仅 DBSCAN 聚类的区别

为了弥补层次化聚类在对驾驶偏好进行聚类时存在的不足,我们把层次化聚类的结果使用 DBSCAN 算法<sup>[18]</sup>再次聚类.先层次化后 DBSCAN 的混合聚类方式,不仅体现了原有粗粒度驾驶偏好,也突出了同偏好下的细粒度差异;提高了驾驶偏好概率精确度,也更好地刻画了用户的驾驶偏好。

到目前为止,根据同用户所有历史路径,得到他们的路径偏好,并对其按照偏好的分量非递增排序;经过混合聚类最终得到  $K$  个路径偏好聚类.由于混合高斯模型(Gaussian mixture model,简称 GMM)可以描述任意复杂的概率函数,我们采用混合高斯模型来描述驾驶偏好向量中每个随机变量<sup>[33]</sup>,以此刻画用户的驾驶偏好.给定某用户的所有路径偏好,经过混合聚类算法,得到  $K$  个路径偏好聚类,把驾驶偏好中第  $i$  个属性看做是随机变量 ( $1 \leq i \leq m$ ),用 GMM 算法构建高斯混合分布<sup>[29]</sup>,求得  $GMM_i(\gamma^i) = \sum_{k=1}^{K_i} g_{i,k} \cdot \mathcal{N}(\gamma^i | \mu_{i,k}, \sigma_{i,k}^2)$ . 其中,  $g_{i,k}$  表示第  $i$  个属性的第  $k$  个高斯分布系数,且满足条件  $\sum_{k=1}^{K_i} g_{i,k} = 1$ . 则第  $k$  个高斯分量  $\mathcal{N}(\gamma^i | \mu_{i,k}, \sigma_{i,k}^2)$  的均值和方差分别为  $\mu_{i,k}$  和  $\sigma_{i,k}^2$ , 记第  $i$  个属性的概率密度分布函数为  $pdf_i$ .

**定义 4(驾驶偏好适配度).** 给定某用户的驾驶偏好密度函数  $pdf$  和路径  $P = (e_1, \dots, e_\ell)$ , 令  $P$  的路径偏好为  $\Gamma(P) = (\gamma^1, \dots, \gamma^m)$ , 则用户选择路径  $P$  的概率为  $\mathcal{AF}(P, pdf) = \prod_{i=1}^m \int_{\gamma^i - \Delta}^{\gamma^i + \Delta} pdf_i(\gamma^i) d\gamma^i$ , 其中,  $\Delta$  用来求路径中第  $i$  个偏好分量  $\gamma^i$  的  $\Delta$ -邻域  $[\gamma^i - \Delta, \gamma^i + \Delta]$ . 称  $\mathcal{AF}(P, pdf)$  是驾驶偏好适配度函数(driving preference adaption function, 简称  $\mathcal{AF}$ ), 该函数返回路径  $P$  在用户的驾驶偏好下的适配度, 适配度越高, 用户采用该路径的可能性就越高; 反之, 适配度越低, 用户采纳该路径的可能性就越小。

#### 4 两阶段路网匹配算法

本节介绍基于用户驾驶偏好的低采样轨迹路网匹配算法, 该算法包含两个阶段。

- 第 1 阶段为轨迹中所有相邻两采样点各自  $k_1$  条最近邻边之间推荐在用户驾驶偏好下最可能的 Top- $k_2$  Skyline 路径。
- 第 2 阶段为全局匹配阶段, 推荐与查询轨迹最匹配的 Top- $k_3$  全局路径。

例如, 在图 3(a)中, 采样点  $p_1, p_2$  和  $p_3$  的 2 条近邻边分别为  $(v_1, v_2)$  与  $(v_1, v_5)$ 、 $(v_7, v_{10})$  与  $(v_9, v_{10})$  和  $(v_{15}, v_{19})$  与  $(v_{18}, v_{19})$ , 则从  $(v_1, v_2)$  到达  $(v_7, v_{10})$  的 Top-2 Skyline 路径分别为  $P_3$  和  $P_4$ , 从  $(v_1, v_5)$  到达  $(v_9, v_{10})$  的 Top-2 Skyline 路径分别为  $P_1$  和  $P_2$ , 同样可求得从采样点  $p_2$  到  $p_3$  的两组 Top-2 Skyline 路径  $P_5$  与  $P_6$  和  $P_7$  与  $P_8$ . 图 3(b)说明了全局匹配阶段根据第 4.2 节给出的动态规划算法求得全局路径分数最高的两组路径  $P_2 P_5$  和  $P_4 P_8$ , 其中移除了置信度较小

的路径连接.

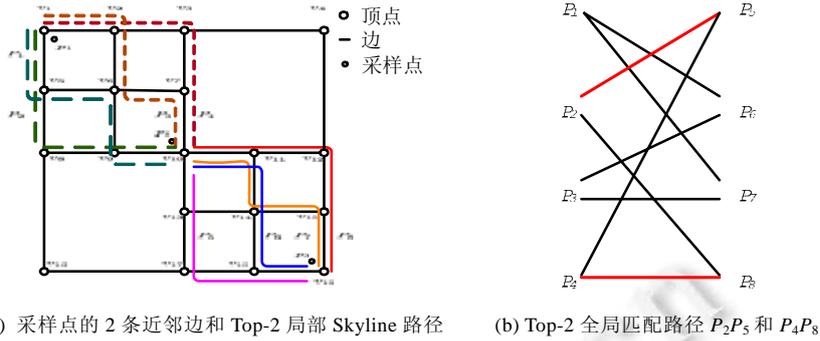


Fig.3 Two-Stage map-matching: first local map-matching stage and second global map-matching stage

图3 两阶段路网匹配:第1阶段局部匹配和第2阶段全局匹配

#### 4.1 局部匹配

在多属性道路网络中,用户在路径规划时往往同时考虑多个属性,所选择的路径至少要在一个属性上比其他路径优秀,很少选择那些在所有属性上都不优秀的路径,即用户选择的路径没有被其他路径支配,我们称这样的路径为 Skyline 路径.在局部路网匹配中,我们以相邻两采样点之间的 Skyline 路径为准,本节先给出局部路网匹配算法 Local Match(如算法 1 所示),然后给出求解 Skyline 路径的 ComputeSkyline 算法.

算法 1. Local Match.

Input: 低采样轨迹:  $\mathcal{Y}$ ; 驾驶偏好:  $\Gamma$ ; 采样点紧邻边数量:  $k_1$ ; Skyline 路径数量:  $k_2$ ; 时间松弛变量:  $\delta$ .

Output: 候选顶点对间 Top- $k$  Skyline 路径:  $\mathcal{P}_{SKY}$ .

```

1  $\mathcal{P}_{SKY}[\mathcal{Y}] \leftarrow \emptyset$ ;
2 for  $i=1 \rightarrow |\mathcal{Y}|-1$  do
3    $E_i \leftarrow$  距离采样点  $\mathcal{Y}[i]$  最近的  $k_1$  个候选边集合;
4    $E_{i+1} \leftarrow$  距离采样点  $\mathcal{Y}[i+1]$  最近的  $k_1$  个候选边集合;
5   for  $\forall e_i \in E_i$  do
6     for  $\forall e_{i+1} \in E_{i+1}$  do
7        $v_i \leftarrow$  距离  $\mathcal{Y}[i]$  在边  $e_i$  上投影点最近的顶点;
8        $v_{i+1} \leftarrow$  距离  $\mathcal{Y}[i+1]$  在边  $e_{i+1}$  上投影点最近的顶点;
9        $\mathcal{P}_{SKY}[i] \leftarrow \mathcal{P}_{SKY}[i] \cup \text{top}(\text{ComputeSkyLine}(v_i, v_{i+1}, \text{time}(v_i), \text{time}(v_{i+1}), \delta, \Gamma, k_2))$ ;
10 return  $\mathcal{P}_{SKY}$ ;

```

如算法 1 所示,由于较高的属性维度和较远的两点间距离会导致过多的 Skyline 路径,这将增加路网匹配的时间复杂度.为了降低局部匹配路径的数量,我们选择那些与用户驾驶偏好适配度最高的  $k_2$  条 Skyline 路径作为局部匹配结果.给定某用户的驾驶偏好  $\Gamma$  和该用户生成的低采样轨迹  $\mathcal{Y}$ , 令每对相邻采样点  $\mathcal{Y}[i]$  和  $\mathcal{Y}[i+1]$  的  $k_1$  条最近邻边集合分别是  $E_i$  和  $E_{i+1}$  (第 3 行、第 4 行), 为分别来自两个集合的边对  $(e_i, e_{i+1}) (e_i \in E_i, e_{i+1} \in E_{i+1})$  推荐  $k_2$  条 Skyline 路径 (第 9 行), 其中,  $v_i$  是边  $e_i$  的端点  $e_i.s$  或  $e_i.d$ , 且  $v_i$  距离采样点  $\mathcal{Y}[i]$  投影到邻边  $e_i$  位置最近 (第 7 行、第 8 行);  $\text{time}(v)$  返回轨迹到达  $v$  的时刻.

下面给出 Skyline 路径定义, 然后讨论在局部路网匹配下问题设定中的 Skyline 路径特征, 并给出高效的 Skyline 路径查询算法 ComputeSkyline.

定义 5 (skyline 路径). 在具有  $m$  个属性的路网  $G$  中, 令  $\mathcal{P}_{(s,d)}$  是从起点  $s$  出发到达终点  $d$  的所有路径集合, 则  $P \in \mathcal{P}_{(s,d)}$  支配路径  $Q \in \mathcal{P}_{(s,d)}$ , 记为  $P < Q$ , 当且仅当满足如下两个条件.

- 1) 任意  $i \in [1, m]$ , 使得  $S^i(P) \leq S^i(Q)$  成立;

2) 存在  $i \in [1, m]$ , 使得  $S^i(P) < S^i(Q)$  成立.

即路径  $P$  的代价在所有属性上不比  $Q$  差, 且至少在一个属性上比  $Q$  优. 路径集合  $\mathcal{P}_{SKY} \subset \mathcal{P}_{(s,d)}$  中任意路径是 Skyline 路径, 当且仅当任意  $P \in \mathcal{P}_{SKY}$ , 不存在其他路径  $Q \in \mathcal{P}_{(s,d)}$  使得  $Q < P$  成立.

**引理 1 (skyline 路径的前缀子路径也是 skyline 路径).** 令路径  $P = (e_1, \dots, e_\ell) \in \mathcal{P}$  是 Skyline 路径, 则  $P$  的前缀子路径  $P' = (e_1, \dots, e_{\ell'})$  也是 Skyline 路径, 其中,  $1 \leq \ell' < \ell$  且  $P'.e_i = P.e_i$ .

证明: 给定 Skyline 路径  $P$ , 假设  $P$  的前缀子路径  $P'$  不是 Skyline 路径. 记  $P = P'P''$ , 即  $P'$  和  $P''$  尾首连接构成路径  $P$ , 那么肯定存在另外一条与  $P'$  共享起止点的 Skyline 路径  $Q' (Q' \neq P')$ , 且  $Q'$  支配  $P'$ , 得到:

1) 任意  $i \in [1, m]$ , 使得  $S^i(Q') \leq S^i(P')$  成立;

2) 存在  $i \in [1, m]$ , 使得  $S^i(Q') < S^i(P')$  成立.

此时, 新路径  $Q = Q'P''$ , 根据  $S^i(\cdot)$  的可列可加性可知,  $S^i(Q) = S^i(Q') + S^i(P'')$  和  $S^i(P) = S^i(P') + S^i(P'')$  成立, 则

1) 任意  $i \in [1, m]$ , 使得  $S^i(Q) \leq S^i(P)$  成立;

2) 存在  $i \in [1, m]$  使得  $S^i(Q) < S^i(P)$  成立.

即  $Q$  支配  $P$ . 根据 Skyline 路径的定义可知, 路径  $P$  不是 Skyline 路径. 这与  $P$  是 Skyline 路径矛盾, 因此不存在路径  $Q'$  支配  $P'$ , 即  $P'$  不是 Skyline 路径不成立, 也即 Skyline 路径的前缀子路径是 Skyline 路径. 引理得证.  $\square$

我们从源点  $s$  出发但没有到达终点  $d$  的路径为部分路径 ( $s$  和  $d$  是采样轨迹相邻两采样点各自近邻边的两个端点), 称已经到达  $d$  的路径为完整路径. 由于在 Skyline 路径查询中, 只有终点相同的路径之间才可用支配关系相互剪枝, 因此, 到达  $d$  的完全路径之间可以进行比较, 从而应用支配关系来剪枝掉某条完整路径.

由于在路网  $G$  中旅行时间是动态的, 不能保证车辆在道路中先进先出, 原有基于静态属性的多目标路径查询算法难以用来解决动态网络情形, 即同终点的部分路径之间, 由于到达的时间差异而导致后期路径的边权重的改变, 而导致难以比较其支配关系. 若采用穷举法枚举所有从起点到终点的 Skyline 路径, 再求得与用户驾驶偏好适配度高的 Top- $k_2$  Skyline 路径, 这种穷举法效率较低下, 仅适合路网规模较小的情况.

因此, 我们提出了多剪枝策略的 Skyline 算法 ComputeSkyline (如算法 2 所示).

4 个关键的剪枝策略如下所示.

1) 共享终点的部分路径之间剪枝策略, 称为树剪枝策略 (tree pruning strategy).

2) 终点处 Skyline 完整路径与经预估到达终点的部分路径之间的剪枝策略, 称为预估剪枝策略 (estimation pruning strategy).

3) 终点处完整路径之间剪枝, 称为完整路径剪枝策略 (complete path pruning strategy).

4) 局部路径从当前顶点开始预估经由最快路径到达终点, 到达终点时间比已知时间是否晚了超过  $\delta \min$ , 称为时间松弛剪枝策略 (time relaxed pruning strategy).

**剪枝策略 1 (树剪枝策略).** 起止点相同的局部路径  $P$  和  $Q$ , 若  $P$  支配  $Q$ , 则可安全地把路径  $Q$  剪枝掉, 即可把所有以  $Q$  为前缀并扩展到终点的路径扩展树剪掉.

证明: 如图 4(a) 所示, 给定共享起止点的局部路径  $P$  和  $Q$ , 满足条件  $P$  支配  $Q$ , 假设存在从起点到终点的路径  $R = QQ'$  是 Skyline 路径. 由于路径  $R = QQ'$  的前缀子路径  $Q$  被  $P$  支配, 这与引理 1 矛盾, 因此假设不成立. 即不存在以  $Q$  为前缀而到达终点的路径是 Skyline 路径. 因此,  $Q$  可被安全剪枝掉. 树剪枝策略得证.  $\square$

**剪枝策略 2 (预估剪枝策略).** 给定任意从查询起点到终点的完全路径  $P$  和尚未到达终点的局部路径  $Q$ , 且  $P$  是 Skyline 路径. 若存在理想的子路径  $Q'$  作为  $Q$  到达终点的后缀路径, 令包含理想子路径  $Q'$  的理想完整路径为  $R = QQ'$ , 称  $R$  是局部路径  $Q$  的预估路径, 其中, 理想后缀路径  $Q'$  在所有属性上都是从  $Q'$  的起点到终点的最短路径, 即任意  $i \in [1, m]$ , 则  $S^i(Q') = \min\{S^i(P') \mid P' \in \mathcal{P}_{(Q'.e_1, s, Q'.e_\ell, d)}\}$ . 若  $P$  支配  $Q$  的预估路径  $R$ , 则可把部分路径  $Q$  剪枝掉.

证明: 如图 4(b) 所示, 给定 Skyline 完整路径  $P$  和部分路径  $Q$ , 以及  $Q$  的预估后缀理想子路径  $Q'$  和预估理想完整路径  $R = QQ'$ , 且  $P$  支配  $R$ . 假设存在另外一条子路径  $Q''$  与  $Q$  连接最终到达终点, 记该完整路径为  $R' = QQ''$ , 且  $R'$  是 Skyline 路径. 由于  $Q'$  是理想后缀路径, 即  $S^i(Q') = \min\{S^i(P') \mid P' \in \mathcal{P}_{(Q'.e_1, s, Q'.e_\ell, d)}\} \leq S^i(Q'')$ , 根据路径的连接性可知,  $S^i(R) = S^i(Q) + S^i(Q')$ ,  $S^i(R') = S^i(Q) + S^i(Q'')$ . 因此,  $S^i(R) = S^i(Q) + S^i(Q') \leq S^i(R') = S^i(Q) + S^i(Q'')$ . 根据  $P$  支配  $R$  可知:

(1) 任意  $i \in [1, m]$ , 使得  $S^i(P) \leq S^i(R)$  成立; (2) 存在  $i \in [1, m]$ , 使得  $S^i(P) < S^i(R)$  成立. 因此, 如下两个条件成立: (1) 任意  $i \in [1, m]$ , 使得  $S^i(P) \leq S^i(R) \leq S^i(R')$  成立; (2) 存在  $i \in [1, m]$ , 使得  $S^i(P) < S^i(R) \leq S^i(R')$  成立. 即  $P$  支配  $R'$ . 根据 Skyline 路径定义可知, 路径  $R'$  不是 Skyline 路径, 这与假设  $R'$  是 Skyline 路径矛盾. 即不存在以  $Q$  为前缀达到终点的路径是 Skyline 路径. 因此, 当完整路径支配了部分路径  $Q$  的预估路径时, 则任意从  $Q$  扩展到终点的路径都不可能是 Skyline 路径, 即可把部分路径  $Q$  安全地剪枝掉. 预估剪枝策略得证.  $\square$

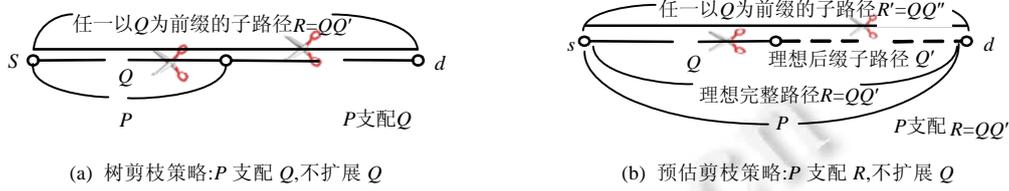


Fig.4 Pruning between shared endpoints local paths and pruning between estimated to destination path and whole path

图 4 共享起止点部分路径间树剪枝和预估到终点路径与完整路径之间预估剪枝

### 算法 2. ComputeSkyline.

Input: 起点:  $v_s$ ; 终点:  $v_d$ ; 出发时间:  $t_s$ ; 到达时间:  $t_e$ ; 时间松弛变量:  $\delta$ .

Output: 从起点  $v_s$  到终点  $v_d$  的 Skyline 路径.

```

1 Skyline 路径  $\mathcal{P}_{SKY} \leftarrow \emptyset$ ; 最晚到达时间  $Latest \leftarrow t_e + \delta$ ;
2 for  $i=1 \rightarrow m$  do
3    $P_{min}^i \leftarrow OneToAllSP(v_d, t_s, t_e, i)$ ;
4    $c(P_{min}^i, t_s) \leftarrow (S^1(P), \dots, S^m(P))$ ;
5    $\mathcal{P}_{SKY} \leftarrow UpdateSkyline(\mathcal{P}_{SKY}, P_{min}^i)$ ;
6 for  $v \in V \wedge v \neq v_d$  do  $v.LB \leftarrow \emptyset$ ;
7  $v_s.LB \leftarrow (v_s)$ ; 初始化顶点的标签队列  $Q$ , 且  $Q.dequeue(v_s.LB)$ ;
8 while 队列  $Q$  非空 do
9   初始化顶点的标签集合  $S \leftarrow \emptyset$ ;
10  while 队列  $Q$  非空 do
11    $LB \leftarrow Q.dequeue()$ ;
12   for  $\forall P \in LB$  do
13     for 路径  $P$  中最后一条边的每条相邻边  $e \wedge e \notin P$  do
14        $P_{next} \leftarrow$  追加边  $e$  到路径  $P$ ;  $v \leftarrow e.d$ ;
15        $c(P_{next}, t_s) \leftarrow (S^1(P), \dots, S^m(P))$ ;
16       估计到终点  $v_d$  的代价向量  $ECV \leftarrow c(P_{next}, t_s) + v.MCV$ ;
17       if  $ECV$  到达  $v_d$  时间比  $Latest$  晚 then continue;
18       if 路径  $P_{next}$  没有到达终点  $v_d$  then
19         if  $\mathcal{P}_{SKY}$  中的所有 Skyline 路径都不支配  $ECV$  then
20            $v.LB \leftarrow UpdateLabel(v.LB, P_{next})$ ;
21         else  $\mathcal{P}_{SKY} \leftarrow UpdateSkyline(\mathcal{P}_{SKY}, P_{next})$ ;
22       if 顶点标签  $v.LB$  不在  $S$  中 then 添加  $v.LB$  到  $S$ ;
23 把顶点标签集合  $S$  中的所有标签添加到队列  $Q$  中;
24 return  $\mathcal{P}_{SKY}$ ;

```

**剪枝策略 3(完整路径剪枝策略).** 给定完全路径  $P$  和  $Q$ ,若  $P$  支配  $Q$ ,则可把完整路径  $Q$  剪枝掉.

完整路径剪枝策略可根据 Skyline 路径的定义证明.

**剪枝策略 4(时间松弛剪枝策略).** 给定尚未到达终点的局部路径  $P$ ,若  $P$  从当前顶点在经过最快路径到达终点比预计时间晚了  $\delta_{\min}$ ,则可安全地把  $P$  剪枝掉.

下面分析求解 Skyline 路径的具体过程.首先,ComputeSkyline(如算法 2 所示)通过逆 Dijkstra 算法(OneToAllSP 函数)求解所有从其余顶点到达终点  $v_d$  的第  $i$  个属性的最小旅行代价路径(第 2 行~第 5 行).在 OneToAllSP 函数中,边  $e \in E$  的第  $i(1 \leq i \leq m)$  个属性值是该属性在边  $e$  上所有可能取值中的最小值.并且在每个顶点  $v$  处,记录从  $v$  到达  $v_d$  的各个属性最小的代价向量(minimum cost vector,简称 MCV),以此来预估部分路径到达终点处的理想代价,进而可以使用预估剪枝策略(第 16 行~第 20 行).随后修正逆 Dijkstra 算法返回的从  $v_s$  出发到达  $v_d$  的路径各属性代价的精确值,并通过 Skyline 更新算法验证该路径是否为 Skyline 路径(第 4 行、第 5 行).若这样的路径存在,那么可以作为基准 Skyline 路径存储在集合  $\mathcal{P}_{SKY}$  中,用于对部分路径的预估路径剪枝.当属性数量较多时,这些基准 Skyline 路径的加速效果很明显.

随后,以广度优先遍历方式,逐步探索从  $v_s$  出发到达  $v_d$  的所有路径(第 8 行~第 23 行).若这些路径是部分路径,可应用树剪枝和预估剪枝策略.由于在广度优先没有到达终点时,已有的 Skyline 路径最多为  $m$  条,数量较少且剪枝效果好,因此先进行预估剪枝,再进行树剪枝策略(第 19 行、第 20 行).

若待验证路径是 Skyline 路径,则将其添加到 Skyline 路径集合中,作为 Skyline 的基准验证路径,这是与基于标签的多目标路径查询算法最大区别.该基准路径有两种剪枝方式.

- 1) 完整路径剪枝(第 21 行),算法 UpdateSkyline 实现了该剪枝策略(如算法 3 所示),见剪枝策略 3.
- 2) 预估剪枝(第 16 行~第 20 行).

若预估的属性代价被 Skyline 路径支配,则从预估属性代价所属的部分路径出发到达终点的所有路径都不满足 Skyline 定义要求,无需进一步探索该部分路径.该部分路径应该被剪枝,见剪枝策略 2;若预估路径属性代价没有被基准 Skyline 路径支配,那么从该部分路径出发有可能存在从  $v_s$  到  $v_d$  的 Skyline 路径,则需要继续探索该路径(第 20 行).

**算法 3. UpdateSkyline.**

Input: Skyline 路径集合:  $\mathcal{P}_{SKY}$ ;完整路径:  $P$ .

Output: Skyline 路径集合:  $\mathcal{P}_{SKY}$ ;

```

1 for  $\mathcal{P}_{SKY}$  中每条路径  $P'$  do
2   if 路径  $P'$  支配路径  $P$  then
3     return  $\mathcal{P}_{SKY}$ ;
4 for  $\mathcal{P}_{SKY}$  中每条路径  $P'$  do
5   if 路径  $P$  支配路径  $P'$  then
6     从  $\mathcal{P}_{SKY}$  中移除路径  $P'$ ;
7 添加路径  $P$  到  $\mathcal{P}_{SKY}$  中;
8 return  $\mathcal{P}_{SKY}$ ;
```

为了在部分路径中使用树剪枝和预估剪枝策略,我们在每个顶点  $v(v \neq v_d)$  处引入标签  $\mathcal{LB}$ ,记作  $v.\mathcal{LB}$ .标签  $v.\mathcal{LB}$  包含所有从源点  $v_s$  出发到达  $v$  且有可能是 Skyline 前缀的路径.也就是说,扩展任意路径  $P \in v.\mathcal{LB}$  到  $v_d$  得到路径集合  $\mathcal{P}_{ext} = P \diamond \dots \diamond v_d$ ,则至少存在一条路径  $P' \in \mathcal{P}_{ext}$ , $P'$  是 Skyline 路径.添加到  $v.\mathcal{LB}$  中的部分路径能通过预估剪枝策略(算法 2 第 16 行~第 20 行)和树剪枝策略(算法 4 第 2 行、第 3 行、第 4 行~第 6 行),详情见剪枝策略 1.

另一方面,由于 Skyline 路径是低采样轨迹相邻采样点之间的路径,假设相邻采样点匹配到顶点  $v_s$  和  $v_d$ ,令用户到达  $v_s$  和  $v_d$  处的时间戳分别为  $t_s$  和  $t_e$ ,那么候选 Skyline 路径的前缀路径从  $t_s$  出发,在到达  $v_d$  顶点的时刻不能超过时间点  $t_e + \delta$ .若部分路径预估到达顶点  $v_d$  处的时间超出该时间点,则该路径也应被剪枝(算法 2 第 17 行),见剪枝策略 4.根据访问起止点时间戳和时间松弛变量,可以限定搜索的路网顶点数量,提高了算法效率.

#### 算法 4. UpdateLabel.

Input: 标签路径集合:  $\mathcal{LB}$ ; 以  $\mathcal{LB}$  所属顶点为终点的部分路径:  $P$ .

Output: 标签路径集合:  $\mathcal{LB}$ .

```

1 for  $\mathcal{LB}$  中每条路径  $P'$  do
2   if 路径  $P'$  支配  $P$  then
3     return  $\mathcal{LB}$ ;
4 for  $\mathcal{LB}$  中每条路径  $P'$  do
5   if 路径  $P$  支配  $P'$  then
6     从  $\mathcal{LB}$  中移除路径  $P'$ ;
7 添加路径  $P$  到  $\mathcal{LB}$  中;
8 return  $\mathcal{LB}$ ;

```

### 4.2 全局匹配

局部匹配获得低采样轨迹  $\mathcal{Y}$  中相邻两个采样点  $\mathcal{Y}[i]$  和  $\mathcal{Y}[i+1]$  ( $1 \leq i < |\mathcal{Y}|$ ) 之间多个候选的 Skyline 路径  $\mathcal{P}_{SKY}[i]$ . 本节介绍基于动态规划的全局匹配算法, 该算法将具有公共顶点的多条 Skyline 路径串联而得到全局匹配路径, 即低采样轨迹的匹配路径.

依次连接  $P_i \in \mathcal{P}_{SKY}[i]$  和  $P_{i+1} \in \mathcal{P}_{SKY}[i+1]$  构造路径  $P_i \diamond P_{i+1}$ , 其中,  $P_i[[P_i]] = P_{i+1}[[1]]$ , 即路径  $P_i$  的终点是  $P_{i+1}$  的起点. 则最终获得的路径数量非常庞大, 并且很多路径是无效的, 比如在路网属性分别为旅行时间和旅行距离时,  $P_i$  和  $P_{i+1}$  的路径偏好分别为 (0.8, 0.3) 和 (0.5, 0.9), 分别表明用户比较喜爱短的旅行时间和短的旅行距离, 即两者的驾驶偏好不在同一个驾驶偏好聚类中, 那么包含  $* \diamond P_i P_{i+1} \diamond *$  的路径都是无效路径. 为了反映局部 Skyline 路径的可连接性和连接的紧密程度, 我们定义了路径连接置信度 (path connective confidence, 简称 PCC).

**定义 6 (路径连接置信度).** 令路径  $P_a$  和  $P_b$  的驾驶偏好分别为  $\Gamma(P_a)$  和  $\Gamma(P_b)$ , 则路径  $P_a$  和  $P_b$  的连接置信度  $g(P_a, P_b)$  定义为

$$g(P_a, P_b) = \begin{cases} 1 - \frac{\sum_{i=1}^m (\Gamma(P_a) \cdot \gamma^i - \Gamma(P_b) \cdot \gamma^i)^2}{\sum_{i=1}^m (\max(\gamma^i) - \min(\gamma^i))^2}, & P_a[[P_a]] \cdot e = P_b[[1]] \cdot s \\ 0, & \text{otherwise} \end{cases}$$

其中,  $\min(\gamma^i)$  和  $\max(\gamma^i)$  分别返回所有用户驾驶偏好向量的第  $i$  个分量的最小值和最大值.

路径连接可信度表明,  $g(\cdot)$  值越大, 则路径  $P_a$  和  $P_b$  的驾驶偏好越相似, 用户通过路径  $P_a$  接着通过  $P_b$  的概率就越高, 即轨迹所匹配的路径包含  $P_a \diamond P_b$  的可能性越高; 反之,  $g(\cdot)$  取值越小, 两路径的驾驶偏好差异就越大, 而用户很难选择两条路径偏好不同的路径, 轨迹匹配的路径包含  $P_a \diamond P_b$  的可能性就越低.

当把多个局部 Skyline 路径首尾相接两两串联, 一直到轨迹最后一个采样点候选顶点处, 就构造出了该轨迹可能的匹配路径, 我们称其为全局路径. 然而这样的路径数量庞大, 为了能够在不同路径之间进行比较, 我们定义了全局路径分数来概括各路径匹配到轨迹的可能性大小.

**定义 7 (全局路径分数).** 令  $P = P^1 \diamond \dots \diamond P^n$  是某用户的低采样轨迹  $\mathcal{Y}$  的全局匹配路径, 且该用户的驾驶偏好分布密度函数为  $pdf$ , 则路径  $P$  的全局路径分数定义为

$$s(P) = \mathcal{AF}(P^1, pdf) \cdot g(P^1, P^2) \cdot \mathcal{AF}(P^2, pdf) \cdot \dots \cdot g(P^{n-1}, P^n) \cdot \mathcal{AF}(P^n, pdf) = \prod_{i=1}^n \mathcal{AF}(P^i, pdf) \prod_{i=1}^{n-1} g(P^i, P^{i+1}).$$

因此, 适配度越高, 用户采用路径  $P$  的概率越高.

通过全局路径分数, 将低采样轨迹的全局匹配转换为查找分数最高的 Top- $k_3$  全局路径问题. 为了执行该查询, 最基本的方法为枚举所有全局路径, 最后再返回分数最高的 Top- $k_3$  路径. 然而, 枚举所有路径, 在轨迹采样点较多或每个采样点候选边个数较多情况下性能表现非常糟糕. 假设我们考虑每个采样点的  $k_1$  条近邻边, 属于相邻采样点的每个候选边对之间具有  $k_2$  条 Skyline 路径, 若轨迹中采样点个数为  $|\mathcal{Y}| = n+1$ , 则所有全局路径的数量为  $(k_1 k_2)^n$ . 为了避免遍历所有这些路径, 我们设计了动态规划算法 TkGP (top- $k$  global path) 计算在用户驾驶偏好  $\Gamma$

下低采样轨迹最有可能匹配到的 Top- $k_3$  全局路径。

在算法 TkGP 中(如算法 5 所示),我们用矩阵  $M[·,·]$  存储路径连接关系, $M[j,i]$  记录了  $k_3$  个全局或部分路径,这些路径都以子路径  $P_j^i$  为后缀,且路径分数最高,即  $*\diamond P_j^i = \{P^1 \diamond P^2 \diamond \dots \diamond P_j^i, \forall P^m \in \mathcal{P}^m, 1 \leq m < i\}$ . 并且, $M[k,i+1]$  的值定义如下:  $M[k,i+1] = \text{topk}_{j=1}^{P_j^{i+1}} \{P | P = P' \diamond P_k^{i+1}, \forall P' \in M[j,i]\}$ . 其中,  $\text{top}(\mathcal{P}, \Gamma, k_3)$  返回驾驶偏好  $\Gamma$  下 Top- $k_3$  全局路径。

#### 算法 5. TkGP.

Input:  $(\mathcal{P}^1, \mathcal{P}^2, \dots, \mathcal{P}^n), k_3, \Gamma$ .

Output:  $\mathcal{P}$ .

```

1   $M[·,·] \leftarrow$  empty matrix;
2  for  $j=1$  to  $|\mathcal{P}^1|$  do  $M[j,1].add(\mathcal{P}_j^1)$ ;
3  for  $i=2$  to  $n$  do
4      for  $k=1$  to  $|\mathcal{P}^i|$  do
5          for  $j=1$  to  $|\mathcal{P}^{i-1}|$  do
6              for  $\forall P' \in M[j,i-1]$  do  $M[k,i].add(P' \diamond P_k^i)$ ;
7           $M[k,i] \leftarrow$  top( $M[k,i], k_3$ );
8  for  $k=1$  to  $|\mathcal{P}^n|$  do  $\mathcal{P}.add(M[k,n])$ ;
9  return top( $\mathcal{P}, \Gamma, k_3$ );
```

如算法 5 所示,假设待匹配的低采样轨迹采样点个数为  $n$ ,那么在算法 TkGP 中,首先将每个局部路径  $P_j^i \in \mathcal{P}^i$  插入到矩阵  $M[j,1]$  中,随后分别求解矩阵第  $i$  列各行 Top- $k_3$  分值最高的部分路径,其中,  $2 \leq i \leq n$ . 最后,扫描矩阵第  $n$  列(即  $M[* , n]$ )求得分值最高的 Top- $k_3$  全局路径。

- 空间复杂度

假设轨迹有  $n+1$  个采样点,相邻两采样点间平均有  $r$  条路径,则在全局匹配阶段,主要引入了具有  $nr$  个元素的矩阵,每个元素平均容纳  $k_3$  条路径,因此该阶段的空间复杂度为  $nrk_3$ . 在局部匹配方面,假设在时间松弛剪枝下单次 Skyline 计算平均仅需访问  $N$  个顶点,根据  $m$  维向量集合中 Skyline 解数量估计方法<sup>[34]</sup>,可确定在终点处大约有  $O((\ln(N))^{m-1})$  条 Skyline 路径,若内部顶点标签集合  $\mathcal{LB}$  中平均含有  $M$  条部分 Skyline 路径,这些路径在内部顶点处都要存储,则空间复杂度为  $O((\ln(MN))^{m-1})$ ,则整体空间复杂度为  $O((\ln(MN))^{m-1} + nrk_3)$ .

- 时间复杂度

假设轨迹有  $n+1$  个采样点,相邻采样点间平均有  $r$  条路径,即  $\mathcal{P}^i$  平均包含  $r$  条路径,求第  $i$  列  $M[* , i]$  分值最高的部分路径需要迭代  $k_3 r^2$  次操作,最终算法 TkGP 的多项式时间复杂度为  $O(nk_3 r^2)$ ,该时间复杂度与 HRIS 第 2 阶段相同. 在局部匹配阶段, Skyline 路径属于多目标路径优化范畴,已经证明是 NP-难的,且具有指数时间复杂度<sup>[35]</sup>. 根据空间复杂度可知,平均每个顶点处集合  $\mathcal{LB}$  中元素要与其余相邻顶点处比较,假设每个顶点平均有  $X$  个相邻顶点,则时间复杂度最坏情况为  $O((\ln(XMN))^{m-1})$ ,经过启发式剪枝策略,时间复杂度降低到  $1/3$  指数复杂度,即  $O((\ln(XMN))^{(m-1)/3})$ .

## 5 实验

本节在真实数据集上研究 DPMM 的精确度和性能. 所用数据集来自北京市 5 万多辆出租车在 2012 年 12 月将近 13 亿 GPS 轨迹点,平均采样时间间隔为约 38s,前半月轨迹用于离线计算用户驾驶偏好,精确度验证数据从后半月中随机选择来自不同用户的 10 000 条采样间隔不超过 20s 的轨迹,实验结果采用数据的用平均值. 用 HMM 路网匹配算法<sup>[1]</sup>对所有高采样轨迹进行路网匹配,匹配结果作为基准参考. 采用具有 171 505 个顶点和 433 391 条边的北京市路网. 所有算法用 C++ 语言实现,编译器为 gcc-4.8.4,运行在配置为 Intel i7-2600 CPU (3.4GHz), 32G 内存, 7200RPM 1TB 硬盘的工作站上,操作系统为 Linux 3.16. 实验中默认参数见表 2.

Table 2 Experimental parameter setting

表 2 实验参数设置

名称	解释	默认值
SI	采样时间间隔(sampling interval)	12min
$m$	道路交通路网中属性数量	4
$\delta$	时间松弛变量	10min
$k_1$	每个采样点近邻边数量	3
$k_2$	每对相邻采样点的近邻边间 Skyline 路径数量	5
$k_3$	分数最高全局路径数量	10

我们用高采样轨迹经过精确匹配的路径  $P_G$  为基准正确性参照,而低采样轨迹从该高采样轨迹中跳过若干个采样点扩大时间生成,对比算法的路网匹配路径记为  $P_M$ .我们以路网匹配准确率为度量标准,其匹配准确度为  $Accuracy(P_M, P_G) = |P_M \cap P_G| / |P_G|$ ,当有多条全局路径时记为集合  $\mathcal{P}_M$ ,精确度取值为其中的最大值,即

$$Accuracy'(\mathcal{P}_M, P_G) = \max_{P \in \mathcal{P}_M} \{|P \cap P_G| / |P_G|\}.$$

- 对比算法

我们选择基于统计的低采样轨迹路网匹配算法 ST-matching<sup>[11]</sup>和 IVMM<sup>[10]</sup>,数据驱动的路网匹配方法:基于历史轨迹的路径推理系统(history based route inference system,简称 HRIS)<sup>[4]</sup>和基于时空的路径恢复系统(spatio-temporal-based route recovery system,简称 STRS)<sup>[3]</sup>,作为有效性方面的对比算法.

- 采样时间间隔效果

我们评估 DPMM 与 4 种对比算法之间路网匹配准确率,此时采样时间间隔为 2min~22min.如图 5 所示,所有路网匹配的精度随着采样时间间隔增大,精确度都有所降低.当采样时间间隔小于 10min 时,ST-matching 和 IVMM 精确度都能保持在 50% 以上;当采样间隔大于 10min 时,ST-matching 和 IVMM 在匹配精确方面越来越差,HRIS 在采样隔超过 14min 时,精确度方低于 50%.STRS 随着采样时间间隔逐渐增大,下降相对缓慢.在采样时间间隔超过 10min 时,DPMM 比 STRS 的准确率高出 20%.

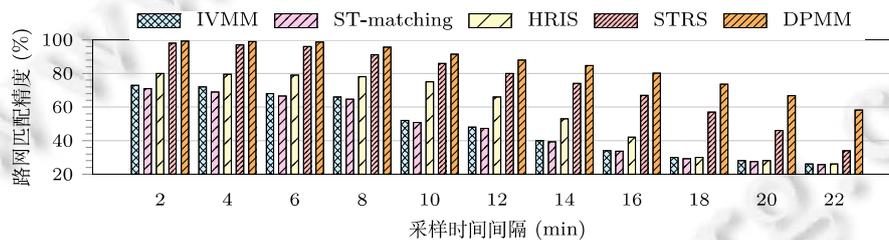


Fig.5 Overall evaluation for accuracy of road network matching

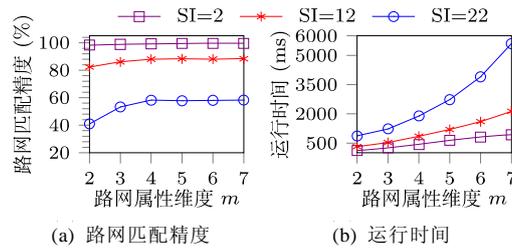
图 5 路网匹配精度比较

为了描述路网属性对匹配精度和运行时间的影响,特有以下设定:当  $m$  为 2~7 时,分别表示旅行时间、旅行距离、油耗、红绿灯数量以及二氧化碳( $\text{CO}_2$ )、一氧化碳( $\text{CO}$ )、碳氢化合物( $\text{HC}$ )和氮氧化物( $\text{NO}_x$ )等气体排放量.其中,油耗和气体排放量采用 VT-Micro 车辆微观气体排放模型<sup>[36]</sup>,该模型能够通过浮动车轨迹计算出油耗和二氧化碳等气体排放详细情况.在实验设置中, $m=2$  表明算法仅考虑旅行时间和旅行距离;当  $m=7$  时,考虑除氮氧化物之外所有路网属性.

- 路网属性维度效果

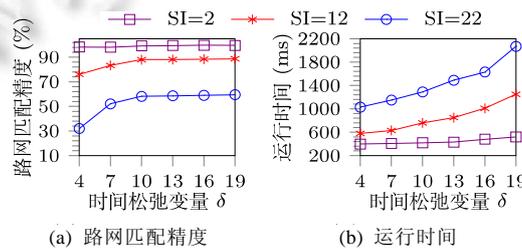
图 6 描述了路网属性维度  $m$  为 2~7 时,改变  $m$  对路网匹配精度和运行时间的影响.图 6(a)表明,路网匹配精度在相同采样间隔下随着  $m$  的增大而增加,且采样率越低路网匹配精确度增加越快;但当  $k=4$  时达到最大值.这主要是由于用户对绿色路径规划关心较少,几乎没有注意到驾驶带来的一氧化碳等气体排放对环境的影响;而当采样频率较高时( $SI=2$ ),由于两采样点之间距离较近而具有较少的 Skyline 路径,反而受  $m$  影响最小.图 6(b)说明,随着路网属性维度  $m$  的增加,平均运行时间也在逐步增大,尤其是两采样点距离越远,路网属性维度就越高;

同距离顶点间的 Skyline 数量急剧增长,距离越远两点之间的 Skyline 路径数量增长越迅速,需要更多的计算代价.

Fig.6 Effect of varying  $m$ 图 6 改变变量  $m$  的效果

- 时间松弛效果

图 7 说明了改变时间松弛变量  $\delta$  对路网匹配精度和运行时间的影响。 $\delta$  主要用来限定从起点出发到达终点不能太迟(见剪枝策略 4)。图 7(a)表明,采样间隔小的轨迹受  $\delta$  影响小,主要原因是两采样点距离较近存在较少的局部 Skyline 路径;而对于采样时间间隔大的轨迹,由于距离较远,局部 Skyline 数量较大,较小的  $\delta$  可能剪枝掉实际路径而降低路网匹配精度,即采样时间间隔越大,路网匹配精度受到  $\delta$  的影响越大。结合图 7(b)可知,较小的  $\delta$  值提高了时间松弛剪枝策略剪枝能力,并降低了采样时间间隔较大轨迹的路网匹配时间;较大的  $\delta$  值(不超过 10min)以削弱剪枝能力和延长运行时间为代价,提高了较低采样轨迹的路网匹配精度,但对采样时间间隔小的轨迹影响小;当  $\delta$  超过 10min 时,则轨迹实际路径不会被  $\delta$  剪枝掉,因此不会再提高精确度;但若继续增大  $\delta$ ,则仅会降低剪枝能力而延长运行时间,从而不能在提高精度方面获得收益。

Fig.7 Effect of varying  $\delta$ 图 7 改变变量  $\delta$  的效果

- 采样点近邻边数量效果

给定两个相邻采样点  $p_i$  和  $p_{i+1}$ , 距离每个采样点最近的  $k_1$  条候选边集合  $E_i$  和  $E_{i+1}$ , 要查询从任意  $e_i \in E_i$  到  $e_{i+1}$  的  $k_2$  条 Skyline 路径(见算法 1), 当  $k_2$  和全局路径  $k_3$  取默认值时, 本部分考察改变  $k_1$  时对路网匹配精度和运行时间的影响。如图 8(a)所示, 当  $k_1$  从 1 增加到 3 时, 能够提高距离较近的采样点之间路网匹配精度; 当超过 3 时, 影响较小; 而对于采样时间间隔较大轨迹的路网匹配精度而言, 当  $k_1$  较小时, 路网匹配精度很低, 随着  $k_1$  的增加, 路网匹配精度有较大提升。图 8(b)表明, 当  $k_1$  增加时, 一方面提高了局部匹配查询的起止点对数量而增加了 Skyline 查询次数而提高了局部匹配的复杂度, 同时也增加了全局匹配时动态规划运行时间, 因此运行时间随着  $k_1$  的增加而增大, 采样率越低, 则 Skyline 的查询距离越远, 使得计算 Skyline 路径的运行时间增加越快。

- 局部 Skyline 路径数量效果

图 9 说明了在局部匹配中起止顶点之间 Skyline 路径数量对路网匹配精度和运行时间的影响。图 9(a)说明: 当采样率较高时, 由于两点之间距离较近, 满足条件的 Skyline 路径数量较少, 因此增加  $k_2$  对精度影响小; 采样率越低的轨迹, 相邻两采样点之间距离越远, 潜在的 Skyline 数量越多, 随着  $k_2$  的增加, 路网匹配精度有明显的提升。

图 9(b)表明,与  $k_2$  相比,求解 Skyline 路径与距离远近对运行时间影响更大,采样率越低,距离就越远,求解 Skyline 路径需要的计算量也就越多,而增加  $k_2$  主要影响全局匹配的动态规划算法运行效率.

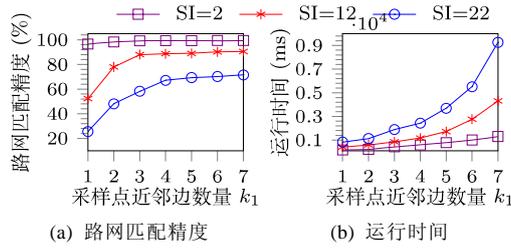


Fig.8 Effect of varying  $k_1$

图 8 改变变量  $k_1$  的效果

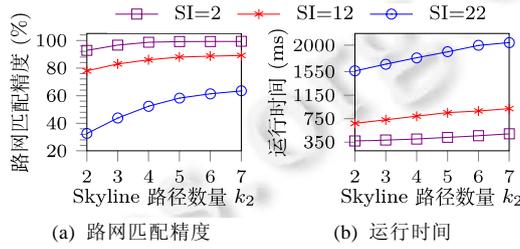


Fig.9 Effect of varying  $k_2$

图 9 改变变量  $k_2$  的效果

• 全局路径数量效果

如图 10(a)所示,随着全局路径数量  $k_3$  增加,路网匹配精度都有所提升,采样率越低,路网匹配精度提高越明显.当  $k_3$  从 2 增大到 10 时,路网匹配精确度也在逐步提高;但当  $k_3$  超过 10 时,路网匹配精确度就不再有所增加.这主要是由于足够大的  $k_3$  值,已经可以覆盖轨迹的实际路径.而图 10(b)表明,随着  $k_3$  的增大,各采样率轨迹路网匹配运行时间都有所增长,采样率越高,轨迹的路网匹配运行时间受  $k_3$  影响越大;采样率越低,影响越小.这主要是由于等距离的轨迹,高采样轨迹点较多,使得动态规划中出现较多阶段;而低采样轨迹阶段较少,且在同阶段中局部边数量固定,则较多阶段比较少阶段受到  $k_3$  的影响要大.

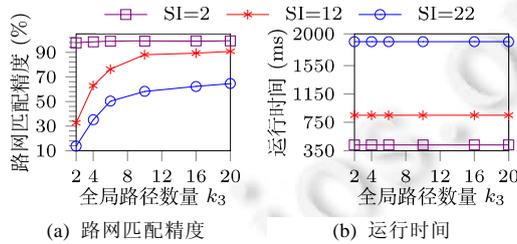


Fig.10 Effect of varying  $k_3$

图 10 改变变量  $k_3$  的效果

6 结 论

本文提出了在多属性动态网络中低采样轨迹路网匹配的问题,考虑到不同用户选择路径时的个性化差异,我们构造了基于用户驾驶偏好的低采样轨迹个性化路网匹配算法,通过从历史高采样轨迹中学习到的驾驶偏好来指导低采样轨迹的路网匹配.在驾驶偏好提取和路网匹配过程中,采用多目标优化思想查找局部 Skyline 路

径作为局部匹配的候选路径.该方法在采样时间间隔为 20min 时也能保证路网匹配精度在 65%以上.我们基于真实数据集做了充分实验,实验结果表明,该方法是有效和高效的.在将来的工作中,我们着重考虑道路网络边界的动态属性服从一定概率分布的情形,并在用户驾驶偏好模型中引入不确定性和动态性,从而提高路网匹配的精度,并设计更加高效的算法,提高路网匹配运行效率.

**致谢** 感谢各位审稿专家,他们给出的意见和建议对提高本文质量起到了关键作用.在此,我们也向对本文工作给予支持和帮助的老师 and 同学表示感谢.

#### References:

- [1] Newson P, Krumm J. Hidden Markov map matching through noise and sparseness. In: Proc. of the GIS 2009. 2009. 336–343.
- [2] Prithu B, Sayan R, Sriram R. Inferring uncertain trajectories from partial observations. In: Proc. of the ICDM 2014. 2014. 30–39.
- [3] Wu H, Mao JY, Sun WW, Zheng BH, Zhang HY, Chen ZY, Wang W. Probabilistic robust route recovery with spatio-temporal dynamics. In: Proc. of the 22nd ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. 2016. 1915–1924.
- [4] Zheng K, Zheng Y, Xie X, Zhou XF. Reducing uncertainty of low-sampling-rate trajectories. In: Proc. of the 28th Int'l Conf. on Data Engineering. 2012. 1144–1155.
- [5] Ceikute V, Jensen CS. Routing service quality local driver behavior versus routing services. In: Proc. of the 14th Int'l Conf. on Mobile Data Management. 2013. 97–106.
- [6] White CE, Bernstein D, Alain LK. Some map-matching algorithms for personal navigation assistants. In: Proc. of the Transportation Research Part C8. 2000. 91–108.
- [7] Dai J, Ding ZM, Xu J. Context-Based moving object trajectory uncertainty reduction and ranking in road network. Journal of Computer Science and Technology, 2016,31(1):167–184.
- [8] Meng Y. Improved Positioning of land vehicle in ITS using digital map and other accessory information [Ph.D. Thesis]. Hong Kong: Department of Land Surveying & Geo-Informatics, Hong Kong Polytechnic University, 2006.
- [9] Yin HB, Wolfson O. A weight-based map matching method in moving objects databases. In: Proc. of the Int'l Conf. on Scientific and Statistical Database Management (SSDBM 2004), Vol.16. 2004. 437–410.
- [10] Yuan J, Zheng Y, Zhang C, Xie X, Sun GZ. An interactive-voting based map matching algorithm. In Proc. of the 11th Int'l Conf. on Mobile Data Management (MDM 2010). 2010. 43–52.
- [11] Lou Y, Zhang C, Zheng Y, Xie X, Wang W, Huang Y. Map-Matching for low-sampling-rate GPS trajectories. In: Proc. of the 17th ACM SIGSPATIAL Int'l Conf. on Advances in Geographic Information System. 2009. 352–361.
- [12] Raymond R, Morimura T, Osogami T, Hirose N. Map matching with hidden Markov model on sampled road network. In: Proc. of the 21st Int'l Conf. on Pattern Recognition (ICPR 2012). 2012. 2242–2245.
- [13] Goh CY, Dauwels J, Mitrovic N, Asif MT, Oran A, Jaillet P. Online map-matching based on Hidden Markov model for real-time traffic sensing applications. In: Proc. of the 15th Int'l Conf. on Intelligent Transportation Systems. 2012. 776–781.
- [14] Ochieng WY, Quddus MA, Noland RB. Map-Matching in complex urban road networks. Journal of Cartography, 2004,55(2):1–18.
- [15] Brakatsoulas R, Pfoser D, Salas R, Wenk C. On map-matching vehicle tracking data. In: Proc. of the VLDB 2005. 2005. 852–864.
- [16] González MC, Hidalgo CA, *et al.* Understanding individual human mobility patterns. Nature, 2008,453(7196):779–782.
- [17] Li Y, Huang QX, Michael K, Zhang L, Leonidas G. Large-Scale joint map matching of GPS traces. In: Proc. of the 21st ACM SIGSPATIAL Int'l Conf. on Advances in Geographic Information Systems (GIS 2013). 2013. 214–223.
- [18] Javanmard A, Haridasan M, Zhang L. Multi-Track map matching. In: Proc. of the GIS 2012. 2012. 394–397.
- [19] Chang KP, Wei LY, Ling Y, Yeh MY, Peng WC. Discovering personalized routes from trajectories. In: Proc of the 3rd ACM SIGSPATIAL Int'l Workshop on Location-Based Social Networks. 2011. 33–40.
- [20] Silva ER, Souza C, Paiva AC. Personalized path finding in road networks. In: Proc. of the NCM. 2008. 586–591.
- [21] Balteanu A, Jossè G, Schubert M. Mining driving preferences in multi-cost networks. In: Proc. of the SSTD. 2013. 74–91.
- [22] Yang B, Guo CJ, Ma Y, Jensen CS. Toward personalized, context-aware routing. The VLDB Journal, 2015,25(2):279–318.
- [23] Dai J, Yang B, Guo CJ, Ding DZ. Personalized route recommendation using big trajectory data. In: Proc. of the Int'l Conf. on Data Engineering. 2015. 543–554.

- [24] Yang YJ, Gao H, Li JZ. Optimal path query based on cost function over multi-cost graphs. Chinese Journal of Computers, 2012, 35(10):2147–2158 (in Chinese with English abstract).
- [25] Wei XJ, Yang J, Li CP, Chen H. Skyline query processing. Ruan Jian Xue Bao/Journal of Software, 2008,19(6):1386–1400 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/19/1386.htm> [doi: 10.3724/SP.J.1001.2008.01386]
- [26] Huang ZH, Xiang Y, Xue YS, Liu XL. An efficient method for processing skyline queries. Journal of Computer Research and Development, 2010,47(11):1947–1953 (in Chinese with English abstract).
- [27] Kriegel HP, Renz M, Schubert M. Route skyline queries: A multi-preference path planning approach. In: Proc. of the Int'l Conf. Data Engineering, 2010. 261–272.
- [28] Shekelyan M, Josse G, Schubert M. Linear path skylines in multicriteria networks. In: Proc. of the ICDE 2015. 2015. 459–470.
- [29] Yang B, Guo CJ, Jensen CS. Travel cost inference from sparse, spatio-temporally correlated time series using Markov models. Proc. of the VLDB Endowment, 2013,6(9):769–780.
- [30] Orda A, Rom R. Shortest-Path and minimum-delay algorithms in networks with time-dependent edge-length. Journal of ACM, 1997,37(3):607–625.
- [31] Ester M, Kriegel HP, Sander J, Xu XW. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proc. of the 2nd Int'l Conf. on Knowledge Discovery and Data Mining. 1996. 226–231.
- [32] Rokach L, Oded M. Clustering Methods. Data Mining and Knowledge Discovery Handbook. Springer-Verlag, 2005. 321–352.
- [33] Bishop CM. Pattern Recognition and Machine Learning. New York: Springer-Verlag, 2006.
- [34] Bentley JL, Kung HT. On the average number of maxima in a set of vectors and applications. Journal of the Association for Computing Machinery, 1978,25(4):536–543.
- [35] Serafini P. Some considerations about computational complexity for multiobjective combinatorial problems. Recent Advances and Historical Development of Vector Optimization, 1986,294:222–231.
- [36] Rakha H, Ahn K, Trani A. Development of VT-micro model for estimating hot stabilized light duty vehicle and truck emission. Transportation Research Part D: Transportation Environment, 2004,9(1):49–74.

#### 附中文参考文献:

- [24] 杨雅君,高宏,李建中.多维代价图模型上最优路径查询问题的研究.计算机学报,2012,35(10):2147–2158.
- [25] 魏小娟,杨婧,李翠平,陈红.Skyline 查询处理.软件学报,2008,19(6):1386–1400. <http://www.jos.org.cn/1000-9825/19/1386.htm> [doi: 10.3724/SP.J.1001.2008.01386]
- [26] 黄振华,向阳,薛永生,刘啸岭.一种处理 Skyline 查询的有效方法.计算机研究与发展,2010,47(11):1947–1953.



高需(1980—),男,河南南阳人,博士,讲师,CCF 专业会员,主要研究领域为时空数据管理,时空数据挖掘,大数据存储、查询和分析.



丁治明(1966—),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为数据库与知识库系统,时空数据管理,物联网,云计算,大数据.



武延军(1979—),男,博士,研究员,博士生导师,CCF 高级会员,主要研究领域为操作系统,系统安全.



陈军成(1980—),男,博士,讲师,CCF 专业会员,主要研究领域为时空数据管理,时空数据挖掘,大数据存储、查询和分析.



郭黎敏(1984—),女,博士,讲师,CCF 专业会员,主要研究领域为时空数据管理,时空数据挖掘,大数据存储、查询和分析.