


```

13. End For
{Step II: running DBSCAN in MapReduce phase}
14. For each partition p in P do /*select partition in Map phase*/
15.   DBSCANClustering(p); /*running DBSCAN on p in Reduce phase*/
16.   For each point Pts in p do
17.     If Pts.isInner do /*storing result of inner points to local file*/
18.       Output(partition.index,Pts.index+Pts.id);
19.     End if
20.     Else /*storing result of boundary points to HDFS*/
21.       writeFile(partition.index,Pts.index+Pts.id+Pts.isCore);
22.     End else
23.   End for
24. End for
    
```

4 Coterie 轨迹组模式挖掘

由于传统挖掘轨迹组模式 ObjectGrowth 算法^[22]在寻找轨迹闭模式时效率低,因此提出 ClusterGrowth 算法挖掘 closed coteries 模式.图 2 是在 $eps=2$ (邻域半径)和 $Minpts=2$ (邻域内点阈值)条件下对象轨迹和聚类图.

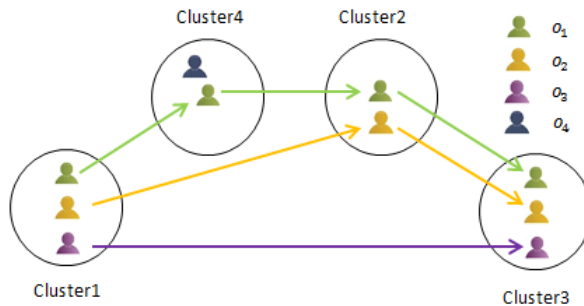


Fig.2 Object trajectories and clusters

图 2 对象轨迹和聚类

由图 2 得知:轨迹对象 o_1 聚类轨迹为 $\{C_1, C_4, C_2, C_3\}$, 聚类 C_1 轨迹对象有 $\{o_1, o_2, o_3\}$.图 3 是在图 2 条件下生成的 ClusterGrowth 搜索树,图中根节点具有所有 Leaguer 轨迹对象 $\{o_1, o_2, o_3, o_4\}$.之后,需要将所有 Cluster 进行前序有序全排列,每个节点被聚类路线 $\{C_x, \dots, C_y\}$ 和拥有聚类路线中所有轨迹对象 $\{o_x, \dots, o_y\}$ 标记.由于当聚类数量较大时搜索空间较大,为了有效地找出 closed coteries,使用前验剪枝和后验剪枝两种剪枝策略压缩搜索空间.

规则 1(前验剪枝). 对于一个 Leaguer 集,若 $L=\{o_x, \dots, o_y\} \subseteq O_{tra}$ 和 $|L| \leq \min_o$, 则 Leaguer 集满足前验剪枝条件.

当 ClusterGrowth 搜索树构建完成后,深度优先搜索树,若树节点轨迹对象 $Leaguer=\{o_x, \dots, o_y\}$ 中对象个数少于阈值 \min_o , 节点被剪枝.例图 3 中,设 $\min_o=2$, 则黄色节点被剪枝.

规则 2(后验剪枝). 对于一个聚类集 $\{C_x, \dots, C_y\} \subseteq C_{tra}$, 如果存在聚类 $C_m \subseteq C_{tra}$ 并且 $m < y$, 使得 $L(\{C_x, \dots, C_y\}) \subseteq L(\{C_x, \dots, C_m, C_y\})$, 则聚类集 $\{C_x, \dots, C_y\}$ 在聚类搜索空间被剪枝.

尽管前验剪枝压缩了搜索空间,但整个搜索空间仍然巨大.因此,使用后验剪枝进一步压缩搜索空间.后验剪枝的思想是:节点中轨迹对象数量不多于节点、子节点轨迹对象数量,则节点被剪枝.原因在于聚类数量增加,而轨迹对象没有减少,即,当前轨迹路线具有更大包容性.图 3 蓝色节点为后验剪枝节点.

规则 3(检验规则). 对于一个聚类集 $\{C_x, \dots, C_y\} \subseteq C_{tra}$, 如果存在聚类 $C_m \subseteq C_{tra}$ 并且 $m > y$, 使得 $L(\{C_x, \dots, C_y\}) \subseteq L(\{C_x, \dots, C_y, C_m\})$, 则聚类集 $\{C_x, \dots, C_y\}$ 在聚类搜索空间被剪枝.

除了上述两种剪枝策略外,还使用检验规则验证 *coterie* 是否为 *closed coterie*.检验规则思想与后验剪枝策略思想大致相同,由于后验剪枝只能在父节点和子节点之间进行剪枝条件判断,而检验规则防止了其他节点之间剪枝遗漏,如图 3 橙色节点,由于不满足先验剪枝和后验剪枝条件被标记为 *closed coterie*,但检验规则可以将橙色节点剪枝.图 3 中灰色节点为最终 *closed coterie*s.

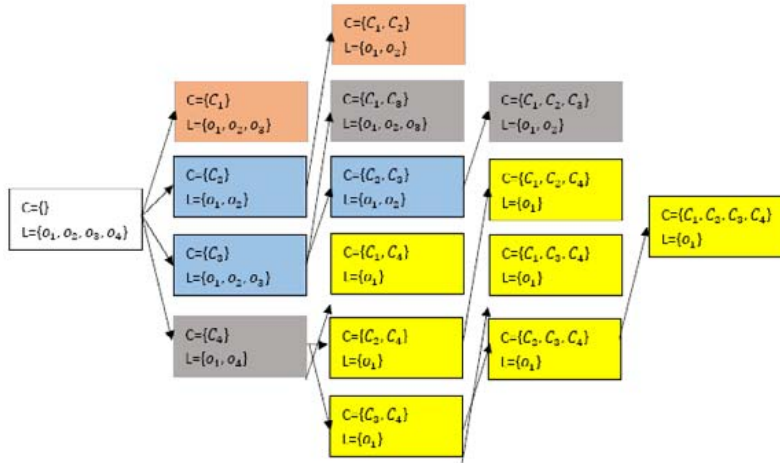


Fig.3 ClusterGrowth algorithm

图 3 ClusterGrowth 算法

5 基于语义的个性化旅游路线推荐

传统组模式挖掘存在语义信息缺失问题,降低了个性化旅游路线推荐质量,因此提出两种基于语义的推荐策略 DRSS 和 CRSS,并且使用了 *tf-idf* 技术结合余弦相似度度量语义相关性.

tf-idf 是一种用于信息检索与数据挖掘的常用加权技术,主要思想是:某个词或短语在一篇文章中出现频率高,并且在其他文章很少出现,则认为此词或短语具有良好的类别区分能力.在一份给定文件中,词频(*tf*)指的是某一个给定词语在文件中出现的频率.对于在某一特定文件中的词语,重要性表示为

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{1}$$

其中, $tf_{i,j}$ 表示在文件 j 中词语 i 的词频重要性, $n_{i,j}$ 表示词语 i 在文件 j 中出现的次数, $\sum_k n_{k,j}$ 表示文件 j 中的所有词语的出现次数之和.

逆向文件频率(*idf*)是词语普遍重要性度量.词语 *idf* 可以由总文件数目和词语出现文件数目衡量,同时防止词语不存在于语料库,*idf* 表示为

$$idf_i = \log \frac{|D|}{1+|\{j:t_i \in d_j\}|} \tag{2}$$

其中, idf_i 表示词语 i 的逆向文件频率, $|D|$ 表示语料库中的文件总数, $|\{j:t_i \in d_j\}|$ 表示包含词语 i 的文件数目.由公式(1)和公式(2)可得 *tf-idf* 为

$$tf-idf_{i,j} = tf_{i,j} \times idf_{i,j} \tag{3}$$

公式(3)可知,高词语频率与低文件频率可以产生高权重的 *tf-idf*.即,*tf-idf* 可以过滤掉常见词语,保留重要的词语.

余弦相似性通过 *tf-idf* 产生的词语生成词频向量,然后由向量余弦夹角计算公式求得余弦相似性:

$$\cos \theta = \frac{\sum_{i=1}^n (A_i \times B_i)}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}} \quad (4)$$

余弦值越接近 1,就表明两个向量越相似.通过将 tf-idf 技术与余弦相似性结合,计算语义相似性.

5.1 基于语义的距离敏感推荐策略(DRSS)

距离对于旅游者选择旅游路线起着十分重要的作用,旅游者会尽量保证在不影响旅行地点前提下寻找最短旅行线路.因此,将语义信息与距离信息结合起来实现旅行路线推荐,可以提高个性化推荐质量,称为 DRSS.

首先,在 closed coteries 中提取每条轨迹语义信息和距离信息.距离因素与推荐系数成反比,距离越长,旅游者选择此路线几率越小.而旅游者输入语义与轨迹语义相关性越大越容易被旅游者选择,即,语义相关性与推荐系数成正比.因此提出距离得分函数:

$$Dscore_{i,j} = 1 - \frac{Dis_{tra_{i,j}}}{\max_{k=traN_j} \{Dis_{tra_{1,j}}, Dis_{tra_{2,j}}, \dots, Dis_{tra_{k,j}}\}} \quad (5)$$

其中, $Dis_{tra_{i,j}}$ 表示第 j 个 closed coteries 的第 i 条轨迹距离, $Dscore_{i,j}$ 表示第 j 个 closed coteries 的第 i 条轨迹距离得分, $traN_j$ 表示第 j 个 closed coteries 轨迹数量.

由公式(4)可得,语义相似度得分函数表示为

$$Score_{tra_{i,j}} = \cos \theta = \frac{\sum_{i=1}^n (A_i \times B_i)}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}} \quad (6)$$

其中, $Score_{tra_{i,j}}$ 表示第 j 个 closed coteries 中第 i 条轨迹语义相似度得分.由公式(5)和公式(6),可以将 DRSS 的推荐得分函数表示为

$$score_{tra_{i,j}} = \alpha \times Dscore_{tra_{i,j}} + (1 - \alpha) \times Sscore_{tra_{i,j}} \quad (7)$$

其中, α 为距离因素所占比重,设定为 0.5; $Dscore_{tra_{i,j}}$ 为距离因素得分; $Sscore_{tra_{i,j}}$ 为语义因素得分; $score_{tra_{i,j}}$ 为 DRSS 最终得分.

5.2 基于语义的从众性推荐策略(CRSS)

在基于组模式个性化旅游路线推荐中,从众性在个性化旅游路线推荐中起着重要作用.当旅游者来到一个陌生城市,旅游计划会更倾向于游客多的旅游路线;同时,从众性因素和语义相关性均与推荐系数成正比.因此提出基于语义与从众性推荐策略,称为 CRSS.

从众性的旅游路线得分利用计算 PageRank 思想,即,计算起点到终点之间每个相邻聚类得分.相邻聚类之间得分被相邻聚类序列和包含在聚类中兴趣度两个因素影响.因为每个聚类中可能存在多个旅游者,则旅游者在选择路径时,路线在聚类中的概率也会影响路线选择.相邻聚类之间,得分函数主要与如下 5 部分有关:(1) 聚类 C_x 的兴趣得分;(2) 聚类 C_y 的兴趣得分;(3) 经过路径 $C_x \rightarrow C_y$ 的旅游者人数;(4) 聚类 C_x 的出度概率;(5) 聚类 C_y 的入度概率.具体表示如下:

$$score_{C_x C_y} = \sum_{tra_{C_x C_y} \in tra_{C_{start} C_{end}}} (I_{C_x} \times Out_{C_x C_y} + I_{C_y} \times In_{C_x C_y}) \quad (8)$$

其中, $score_{C_x C_y}$ 表示 $C_x \rightarrow C_y$ 的得分, $tra_{C_x C_y}$ 表示 $C_x \rightarrow C_y$ 的轨迹, $tra_{C_{start} C_{end}}$ 表示符合起点和终点要求的 closed coteries 中 $C_{start} \rightarrow C_{end}$ 的轨迹, $Out_{C_x C_y}$ 表示聚类 C_x 的出度率(出度到 C_y 的路线占有出度的比率), $In_{C_x C_y}$ 表示聚类 C_y 的入度率, I_{C_x} 表示聚类 C_x 的兴趣得分, I_{C_y} 表示聚类 C_y 的兴趣得分.为便于计算, I_{C_x} 和 I_{C_y} 均设置为 1.

如图 2 所示,假定旅游者要求起始位置为 $v_{start} = \{C_1\} \in C_{tra}$, 终点位置为 $v_{end} = \{C_3\} \in C_{tra}$, 设图 2 的 o_1 和 o_2 轨迹为 closed coterie, 则两条轨迹的得分为

$$score_{c_1c_4c_2c_3} = score_{c_1c_4} + score_{c_4c_2} + score_{c_2c_3} = (0.5 + 1) + (1 + 0.5) + 2 \times (1 + 1) = 7,$$

$$score_{c_1c_2c_3} = score_{c_1c_2} + score_{c_2c_3}.$$

为了从众性因素与语义因素结合,标准化从众性得分:

$$Cscore_{tra_i,j} = \frac{score_{C_{tra_i,j}:startC_{tra_i,j}:end}}{\sum_{k=1}^{traN_j} score_{C_{tra_k,j}:startC_{tra_k,j}:end}} \quad (9)$$

其中, $Cscore_{tra_i,j}$ 表示第 j 个 closed coteries 的第 i 条轨迹的从众性得分, $score_{C_{tra_i,j}:start}$ 表示第 j 个 closed coteries 的第 i 条轨迹的 $start$ 聚类, $score_{C_{tra_i,j}:end}$ 表示第 j 个 closed coteries 的第 i 条轨迹的 end 聚类, $traN_j$ 表示第 j 个 closed coteries 的符合旅游者起始终止条件的轨迹总数.根据公式(6)和公式(9)可得出,CRSS 推荐策略得分函数为

$$score_{tra_i,j} = \beta \times Cscore_{tra_i,j} + (1 - \beta) \times Sscore_{tra_i,j} \quad (10)$$

其中, β 为从众性得分占比,设定为 0.5; $Cscore_{tra_i,j}$ 为从众性得分; $Sscore_{tra_i,j}$ 为语义得分; $score_{tra_i,j}$ 为 CRSS 得分.

6 实验与结果

在实验阶段,通过算法执行效率和两种推荐策略的推荐结果展示实验结果.实验数据集为 Sydney, Melbourne, Brisbane, Perth 和 Darwin 这 5 座城市的 Instagram 数据.由于澳大利亚在 9 月~次年 2 月为旅游旺季,并且推荐算法涉及用户语义信息,因此提取了 2012 年 9 月~2013 年 2 月并过滤掉缺乏语义信息的数据,最终得到 113 366 个用户数据和 851 888 条数据.所有算法均由 Java 实现.

6.1 算法效率

在聚类阶段,DBSCAN 算法有 eps 和 $Minpts$ 这两个参数,输入数据量的规模也会对算法的运行时间产生影响.因此,对 DBSCAN 算法和 nMR-DBSCAN 算法分别在 $eps, Minpts$ 和输入数据量的规模上进行运行时间对比,如图 4 所示.

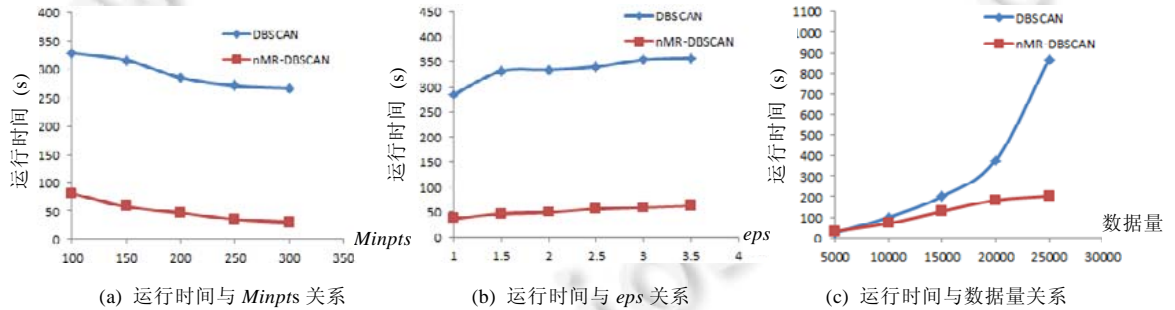


Fig.4 Performance comparison on DBSCAN and nMR-DBSCAN

图 4 DBSCAN 和 nMR-DBSCAN 算法性能比较

由图 4(a)可以看出,两种算法的运行时间随着邻域内元素个数阈值 $Minpts$ 的增加而减少.原因是 $Minpts$ 增大会导致向外扩散的搜索元素减少.由图 4(b)可以看出,DBSCAN 算法和 nMR-DBSCAN 算法的运行时间都随着邻域半径 eps 的增大而逐渐增大.原因是 eps 增大导致额外向外扩散的搜索元素增加.图 4(c)表明:随着数据量的增大,DBSCAN 和 nMR-DBSCAN 算法的执行时间逐渐增加,并且 nMR-DBSCAN 算法的运行时间相比于 DBSCAN 算法有了明显的降低.总之,随着 $eps, Minpts$ 和数据量的变化,nMR-DBSCAN 算法相比于 DBSCAN 算法均有明显的运行时间优势.

在挖掘 closed coteries 阶段,根据不同参数对比了 ClusterGrowth 算法和 ObjectGrowth 算法的运行时间.ObjectGrowth 算法挑选数量更多的对象作为组合项,而 ClusterGrowth 算法挑选更少的聚类作为组合项,因此后者运行时间更少.图 5 表明了随着 min_c 和 min_o 的变化,ClusterGrowth 算法的执行时间始终少于 ObjectGrowth 算法.

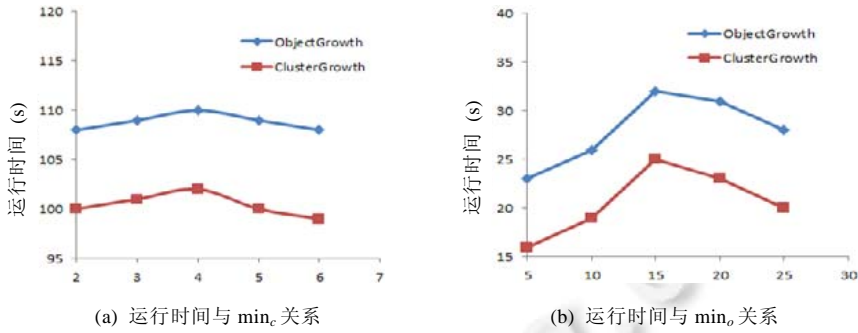


Fig.5 Performance comparison on ObjectGrowth and ClusterGrowth
图 5 ObjectGrowth 和 ClusterGrowth 算法性能比较

在推荐旅游路线阶段, DRSS 和 CRSS 的运行时间在固定 $min_c=3$ 和 $min_o=2$ 的前提下进行实验, 数据量从 5 000~25 000. 图 6 表明: 随着数据量的增大, DRSS 和 CRSS 的运行时间逐渐增加.

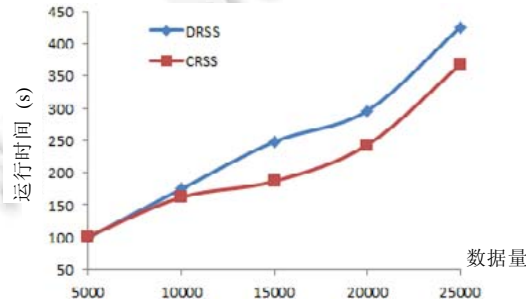


Fig.6 Performance comparison on DRSS and CRSS
图 6 DRSS 和 CRSS 算法性能比较

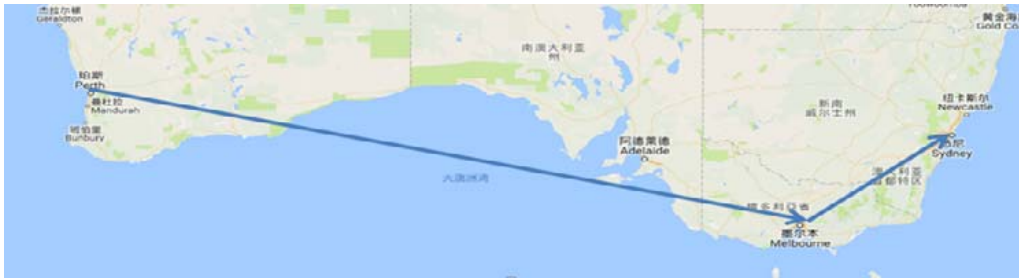
6.2 旅游路线的推荐

在基于组模式个性化旅游路线推荐阶段, 提出了 DRSS 和 CRSS 两种推荐策略. 旅游者可以提供旅行的出发地和目的地、途径旅游地点及语义信息. 通过对所有 closed coteries 的分析处理, 可得到满足旅游者经过地点要求的 closed coteries. 然后, 按照 DRSS 和 CRSS 的得分计算公式计算 closed coteries 组轨迹中每条轨迹的得分, 将相对满足旅游者意愿的路线推荐给旅游者. 同时, 由于 DBSCAN 聚类算法及 ClusterGrowth 算法具有参数敏感性, 因此将参数 $eps, minpts, min_c$ 和 min_o 固定设置为 0.5, 100, 2 和 2 进行实验. 通过展示路线经过的准确地点, 完善地展示推荐路线, 防止由于地图比例尺的影响无法详细获取路线信息.

个性化旅游路线推荐以从 Perth 到 Melbourne、最后到达 Sydney 的路线及 music 关键词作为输入进行推荐展示. 图 7 展示了基于 closed coteries 的 DRSS 和 CRSS 推荐路线.

图 7(a) 展示了城市之间旅游路线推荐, 由于比例尺大, 将 DRSS 和 CRSS 城市之间的轨迹推荐路线放入图 7(a) 中. 图 7(b)~图 7(d) 展示了基于 DRSS 路线推荐的准确位置. 图 7(b) 展示了旅游者在 Perth 的 The Hen House Live 上传 Good music in perth 文本信息, 说明在 Perth 的 Hen House Live 听了一场音乐会. 图 7(c) 展示了推荐轨迹 Melbourne 位置的 The Bottom End 上传了 Melb... 文本信息, 说明旅游者在 Melbourne 去了 Bottom End. 图 7(d) 展示了推荐轨迹在 Sydney 的位置为 Sydney Opera House, 上传了 I love music... 文本信息, 说明旅游者到达 Sydney 去了 Sydney Opera House 听音乐. 图 7(b)~图 7(d) 这条轨迹是将距离和语义结合得到的推荐路线, 能够看出, 这条轨迹符合旅游者路线地点和音乐的要求. 图 7(e)~图 7(g) 展示了基于 CRSS 的路线推荐准确位置. 图 7(e) 显示, 旅游者在 Perth 的 Hyde Park 上传了 Me and Burger 文本信息, 说明在 Perth 的 Hyde Park 游玩. 图 7(f) 展示了推荐轨迹在 Melbourne 的 Arts Center Melbourne 上传了 sweet music 文本信息, 说明旅游者到达 Melbourne 后

去了 Arts Center Melbourne 听音乐.图 7(g)展示了旅游者在 Sydney 的 St mary’s Cathedral 上传了 beautiful...文本信息,即,去了 St mary’s Cathedral.



(a) SDRS 和 SCRS 的推荐路线



(b) DRSS 轨迹起点



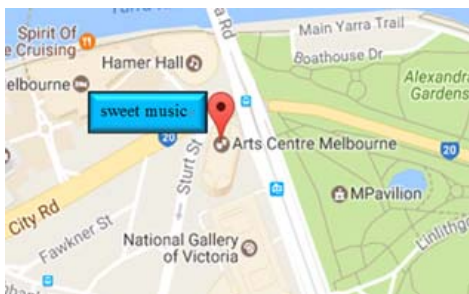
(c) DRSS 轨迹通过点



(d) DRSS 轨迹终点



(e) CRSS 轨迹起点



(f) CRSS 轨迹通过点



(g) CRSS 轨迹终点

Fig.7 Recommending routes by SDRS and SCRS based on closed coterie patterns

图 7 基于 closed coterie 模式的 SDRS 和 SCRS 的路线推荐

从旅游路线推荐图中可以看出,分别将距离因素和从众因素与语义信息结合起来进行路线推荐.以往的推荐 celv 均只考虑单一的影响因素,并且未考虑社交网语义信息对旅游路线推荐的影响.从旅游路线推荐图可以看出:推荐的路线与用户输入的信息进行了信息匹配,相比于传统的只考虑距离因素或者从众因素,提出的推荐

策略会更贴近用户的需求,也会增加用户体验和推荐信息的完整性.

7 总 结

在 coterie 组模式挖掘阶段,为了提高寻找 closed coteries 的效率,使用了 ClusterGrowth 算法,并采用前验剪枝和后验剪枝压缩搜索空间.最后,提出验证规则找出 closed coteries.实验结果表明,ClusterGrowth 算法的效率高于 ObjectGrowth 算法.

在基于组模式的个性化旅游路线推荐阶段,传统的组模式挖掘由于语义信息的缺失导致了个性化推荐不完善,因此提出基于语义的 DRSS 和 CRSS 两种推荐策略,完成个性化旅游路线的推荐.实验展示了 DRSS 和 CRSS 推荐的旅游路线.

在聚类处理阶段,为了提高聚类算法的效率,将 DBSCAN 聚类算法与 Hadoop 平台下的 MapReduce 框架结合起来,并提出了 nPRBP 算法提高分区效率.实验结果表明,nMR-DBSCAN 算法提高了聚类算法的效率.

References:

- [1] Dai BR, Lin IC. Efficient Map/Reduce-based DBSCAN algorithm with optimized data partition. In: Proc. of the IEEE CLOUD. 2012. 59–66. [doi: 10.1109/CLOUD.2012.42]
- [2] Nie LQ, Song XM, Chua TS. Learning from multiple social networks. In: Proc. of the Synthesis Lectures on Information Concepts, Retrieval, and Services. Morgan & Claypool Publishers, 2016. [doi: 10.2200/S00714ED1V01Y201603ICR048]
- [3] Zheng Y, Zhang LZ, Xie X, Ma WY. Mining interesting locations and travel sequences from GPS trajectories. In: Proc. of the WWW. 2009. 791–800. [doi: 10.1145/1526709.1526816]
- [4] Bastani F, Xie X, Huang Y, Powell JW. A greener transportation mode: Flexible routes discovery from GPS trajectory data. In: Proc. of the GIS. 2011. 405–408. [doi: 10.1145/2093973.2094034]
- [5] Savage NS, Nishimura S, Chávez NE, Yan XF. Frequent trajectory mining on GPS data. In: Proc. of the LocWeb. 2010. 3. [doi: 10.1145/1899662.1899665]
- [6] Yin PF, Ye M, Lee WC, Li ZH. Mining GPS data for trajectory recommendation. In: Proc. of the 18th Pacific-Asia Conf. on PAKDD. Springer-Verlag, 2014. 50–61. [doi: 10.1007/978-3-319-06605-9_5]
- [7] Eldawy A, Mokbel MF, Al-Harathi S, Alzaidy A, Tarek K, Ghani S. SHAHED: A MapReduce-based system for querying and visualizing spatio-temporal satellite data. In: Proc. of the ICDE. 2015. 1585–1596. [doi: 10.1109/ICDE.2015.7113427]
- [8] Tsai CY, Lai BH. A location-item-time sequential pattern mining algorithm for route recommendation. Knowl.-Based Syst., 2015, 73:97–110. [doi: 10.1016/j.knosys.2014.09.012]
- [9] Pires CSG, de Aguiar MS, Paulo R. Ferreira: Mobile ACORoute-route recommendation based on communication by pheromones. Polibits, 2015,51:27–32. [doi: 10.17562/PB-51-4]
- [10] Dai J, Yang B, Guo CJ, Ding ZM. Personalized route recommendation using big trajectory data. In: Proc. of the ICDE. 2015. 543–554. [doi: 10.1109/ICDE.2015.7113313]
- [11] Shen Y, Zhao LG, Fan J. Analysis and visualization for hot spot based route recommendation using short-dated taxi GPS traces. Information, 2015,6(2):134–151. [doi: 10.3390/info6020134]
- [12] Wang MM, Zuo WL, Wang Y. An improved density peaks-based clustering method for social circle discovery in social networks. Neurocomputing, 2016,179:219–227. [doi: 10.1016/j.neucom.2015.11.091]
- [13] He YB, Tan HY, Luo WM, Feng SZ, Fan JP. MR-DBSCAN: A scalable MapReduce-based DBSCAN algorithm for heavily skewed data. Frontiers of Computer Science, 2014,8(1):83–99. [doi: 10.1007/s11704-013-3158-3]
- [14] Zheng K, Zheng Y, Yuan N, Shang S. On discovery of gathering patterns from trajectories. In: Proc. of the IEEE 2013. 2013. 242–253. [doi: 10.1109/ICDE.2013.6544829]
- [15] Vieira MR, Bakalov P, Tsotras VJ. On-Line discovery of flock patterns in spatio-temporal data. In: Proc. of the GIS. 2009. 286–295. [doi: 10.1145/1653771.1653812]
- [16] Jeung HY, Yiu ML, Zhou XF, Jensen CS, Shen HT. Discovery of convoys in trajectory databases. Computer Science, 2010,1(1): 1068–1080. [doi: 10.14778/1453856.1453971]

- [17] Li YX, Bailey J, Kulik L. Efficient mining of platoon patterns in trajectory databases. *Data Knowledge Engineering*, 2015,100: 167–187. [doi: 10.1016/j.datak.2015.02.001]
- [18] Fan Q, Zhang DX, Wu HY, Tan KL. A general and parallel platform for mining co-movement patterns over large-scale trajectories. *PVLDB*, 2016,10(4):313–324. [doi: 10.14778/3025111.3025114]
- [19] Liu HL, Li JH, Peng J. A novel recommendation system for the personalized smart tourism route: Design and implementation. In: *Proc. of the ICCI*CC2015*. 2015. 291–296. [doi: 10.1109/ICCI-CC.2015.7259400]
- [20] Hasuike T, Katagiri H, Tsubaki H, Tsuda H. A route recommendation system for sightseeing with network optimization and conditional probability. In: *Proc. of the SMC*. 2015. 2672–2677. [doi: 10.1109/SMC.2015.467]
- [21] Wen YT, Cho KJ, Peng WC, Yeo JY, Hwang SW. KSTR: Keyword-Aware skyline travel route recommendation. In: *Proc. of the ICDM*. 2015. 449–458. [doi: 10.1109/ICDM.2015.37]
- [22] Li Z, Ding B, Han J, Kays R. Swarm: Mining relaxed temporal moving object clusters. *Proc. of the VLDB Endowment*, 2010,3(1): 723–734. [doi: 10.14778/1920841.1920934]



李晓旭(1993—),男,辽宁沈阳人,硕士,主要研究领域为轨迹挖掘,社交网.



张文超(1992—),男,硕士,主要研究领域为数据挖掘,机器学习,社交网,超图.



于亚新(1971—),女,博士,副教授,CCF 专业会员,主要研究领域为大数据挖掘,轨迹挖掘,社交网数据分析.



王磊(1992—),男,硕士,主要研究领域为数据挖掘,社交网,机器学习,网页挖掘.