


```

15:   将  $e_i$  加入到  $\mathcal{P}(u_j)$  中;
16:   While ( $\mathcal{P}(u_j)$  的行程开销超过  $u_j$  的行程预算)
17:     从  $\mathcal{P}(u_j)$  中删除偏爱程度最差的事件  $e_k$ ;
18:     If ( $e_i$  与  $e_k$  是同一个事件)
19:       Break;
20:     End If
21:     将  $e_k$  加入  $conflict\_events$ ;
22:   End While
23:   If ( $e_i$  被从  $\mathcal{P}(u_j)$  的行程中删除)
24:     For (在  $conflict\_events$  中的每个事件  $e_k$ )
25:       将  $e_k$  加入  $\mathcal{P}(u_j)$ , 将  $e_k$  的状态变为 inactive;
26:     End For
27:   Else If (事件  $e_i$  的人数超过其上界)
28:     找到目前事件  $e_i$  偏好程度最低的用户  $u_k$ , 并将  $e_i$  从  $\mathcal{P}(u_k)$  中删除;
29:   End If
30: End For

```

算法 3 首先将所有可能的用户和事件组合按照互相偏序加和,按照和的递增排序,和越小意味着用户和事件之间的偏好程度越高(第 1 行).按顺序取出每一个组合,记 (e_i, u_j) (第 3 行),如果 e_i 被安排的用户数量已经达到了其人数上界且更偏好于当前被安排的用户,则证明 u_j 无法被安排到该事件,于是跳出循环(第 3 行~第 5 行);否则,声明一个集合 $conflict_events$ 来存储与 e_i 冲突的事件,并初始化为空(第 6 行).首先判断 $\mathcal{P}(u_j)$ 中是否有与 e_i 冲突的事件,如果有(设为 e_k),则判断在 $PU(u_j)$ 中 u_j 更喜欢 e_i 还是 e_k :如果用户更喜欢 e_k ,则证明 e_i 无法加入到 $\mathcal{P}(u_j)$ 中,并跳出循环(第 8 行~第 10 行);否则,将 e_k 从 $\mathcal{P}(u_j)$ 中移除,并加入与 e_i 冲突的 $conflict_events$ 集合中(第 11 行~第 14 行).以上证明 e_i 与 $\mathcal{P}(u_j)$ 中的所有事件不冲突,于是将其加入 $\mathcal{P}(u_j)$ 中(第 15 行).如果加入 e_i 会使得 $\mathcal{P}(u_j)$ 的行程开销超过 u_j 的行程预算,则不断的删除 $\mathcal{P}(u_j)$ 偏好最差的事件(第 16 行~第 22 行).如果 e_i 被从 $\mathcal{P}(u_j)$ 中删除,则证明 e_i 不能加入 $\mathcal{P}(u_j)$ 中,那么判断 $conflict_events$ 中的每个事件(依然按照效用值从大到小的顺序)是否有合适的事件 e_k 加入 $\mathcal{P}(u_j)$:如果可以,则加入 e_k (第 23 行~第 26 行).最后,如果 e_i 分配的用户超过人数上界,则删除 e_i 中偏好程度最低的用户,并将 e_i 从 $\mathcal{P}(u_k)$ 中删除(第 27 行~第 30 行).下面用例 4 对算法 3 的过程加以说明.

例 4:如例 1 中,事件和用户的位置如图 2 所示,用户对事件的偏好效用值见表 1,事件对用户的偏好效用值见表 2.根据算法 3,所有用户和事件的组合排序结果为 $(e_2, u_5), (e_3, u_5), (e_1, u_4), \dots, (e_1, u_3)$.因为 e_2 对 u_5 的偏序是 2, u_5 对 e_2 的偏序是 1, 和为 3, (e_2, u_5) 排在最前面; (e_3, u_5) 的和也为 3, 但是 $e_3 > e_2$, 所以排在第 2 个位置; (e_1, u_4) 的和为 4, 排在第 3 个位置;以此类推, (e_1, u_3) 的和为 7, 排在最后一个.首先考察组合 (e_2, u_5) , 将 e_2 加入 $\mathcal{P}(u_5)$ 中.接下来分别考察 (e_3, u_5) 和 (e_1, u_4) , 分别将 e_3 和 e_2 加入 $\mathcal{P}(u_5)$ 和 $\mathcal{P}(u_4)$ 中.当考察 (e_1, u_5) 时, 将 e_1 加入 $\mathcal{P}(u_5)$ 中会超过 u_5 的预算, 所以 e_1 无法加入到 $\mathcal{P}(u_5)$ 中.接下来将 e_2 加入到 $\mathcal{P}(u_3)$ 中, 将 e_2 加入到 $\mathcal{P}(u_3)$ 中.考察 (e_1, u_3) , e_1 已经达到人数上界, 且更偏爱 u_4 , 所以 e_1 加入 $\mathcal{P}(u_3)$ 中.考察 (e_2, u_1) , e_2 已经达到人数上界, 但 e_2 更偏爱 u_1 , 将 e_2 加入到 $\mathcal{P}(u_1)$ 中, 将 e_2 从 $\mathcal{P}(u_3)$ 中移除.下一组, 将 e_3 加入到 $\mathcal{P}(u_3)$ 中. e_3 无法加入到 $\mathcal{P}(u_4)$ 中, 因为将超过 u_4 的预算. e_1 无法加入到 $\mathcal{P}(u_1)$ 中, 因为 e_1 更加偏爱当前的 u_4 . 同理, e_2 无法加入到 $\mathcal{P}(u_2)$ 中. 接下来考察 (e_3, u_1) , 由于加入 e_3 将超过 u_1 的预算, 所以 e_3 无法加入到 $\mathcal{P}(u_1)$ 中. 将 e_3 加入到 $\mathcal{P}(u_2)$ 中. 由于将会超过 u_4 的预算, e_3 无法加入到 $\mathcal{P}(u_4)$ 中. 最后, e_1 无法加入到 $\mathcal{P}(u_3)$ 中, 因为 e_1 更加偏爱当前的 u_3 . 此时, 所有的组合遍历完毕, 算法终止.

算法正确性分析:算法在第 3 行保障当前事件所选择的用户是其最喜欢的,而一旦事件不得不放弃选择其最喜欢的用户时,算法的第 9 行保障该用户选择到的事件是其最喜欢的.因此,至少保障用户或事件一方满足其最喜欢的选择,因此不会出现阻塞对.同时,每当在计划中加入一个事件或者用户时,均会判断是否满足事件人数上界、用户行程开销、事件与计划中是否存在冲突这些所有约束条件是否满足.当且仅当所有的约束条件均

满足时,才会插入该事件或者用户.因此,算法 3 可以满足定义 2 中的所有条件,所以算法 3 是正确的.

算法复杂度分析:算法在第 2 行需要遍历所有可能组合,时间复杂度为 $O(mn)$;其余的计算复杂度为 $O(1)$.因此,算法的时间复杂度为 $O(mn)$,空间复杂度也为 $O(mn)$.

4.3 空间剪枝方法

不难发现,对于任意一个用户 u_i ,他所能参加的最远的事件,其举办地与 u_i 所在位置的距离不能超过 $B_i/2$.因此, u_i 可以参加的事件,其举办地必须在以 u_i 所在位置 l_{u_i} 为圆心、 $B_i/2$ 为半径所在的圆内.于是,在算法运行之前,可以对数据集进行预处理.对每个用户 u_i 来说,以 l_{u_i} 为圆心、 $B_i/2$ 为半径画一个圆,将 l_{e_j} 不在圆内的每个事件 e_j 彼此的效用值 $p_u(u_i, e_j)$ 和 $p_e(e_j, u_i)$ 均置为 0,即从彼此效用值表中删除.利用这种方法可以大量减少两个效用值表的存储,并节省算法在遍历两个效用值表时所用的时间.

5 实验及分析

本节给出算法的实验结果,包括实验环境、算法的计算时间、内存消耗、获得的总体效用值.另外,将本文提出的稳态规划问题定义与现有工作^[6]所提出的问题定义(即忽略定义 2 中的条件(4))及算法进行比较,证明本文提出问题定义及相应算法的必要性.

5.1 数据集及实验环境

本文算法用 C++ 及其 STL 库所实现.实验在一台服务器上运行,该服务器的操作系统为 Linux Fedora 16 (Linux 3.6.11-4.fc16*86 62 GNOME 3.2.1),处理器为 Intel(R)Xeon(R) CPU E5-2620 0@2.00GHz,内存 64GB.内存开销利用 Linux 系统函数进行测试.

本实验所使用的数据集为 Meetup 数据集^[1],数据集中,每个用户有一个标签文件和位置文件.标签文件记录用户在平台上注册时所选择的兴趣标签;位置文件记录用户位置的经纬度坐标.对于每个事件,有一个位置文件以及一个事件群组文件.在 Meetup 中,事件是由事件群所创建的,事件群文件记录着包含在群中的事件,标签文件记录着每个群组的兴趣标签.位置文件记录着每个事件举办地的经纬度.利用用户的表情文件、事件的标签文件、群组的标签文件,可以计算用户对事件感兴趣程度的效用值^[1,3].事件对用户的感兴趣程度,由用户参加事件的历史记录所计算.设 e 为用户 u 参加过的历史事件, e 的标签集为 $\{L_1, L_2, \dots, L_n\}$,则每个拥有 $\{L_1, L_2, \dots, L_n\}$ 中至少其中一个表情的事件对 u 的好感度+1.统计所有历史事件后,将每个事件对用户的好感度进行归一化.其他参数的产生方式与文献[6]中相同.表 3 中总结了真实数据集中的各参数,其中, n 为数据集中的用户数量, m 为数据集中的事件数量, η 为事件参与人数的上界,冲突率表示冲突事件占事件总数的百分比.

Table 3 Real datasets

表 3 真实数据集

城市名称	n	m	η 的均值	冲突率
北京	113	16	50	0.25
奥克兰	569	37	50	0.25
夏威夷	2 967	817	50	0.25
阿拉斯加	2 394	4 156	50	0.25
香港	3 528	1 324	50	0.25
新加坡	9 893	4 257	50	0.25
温哥华	16 095	11 536	50	0.25

为了测试算法的可扩展性,从温哥华真实数据中截取了一定数量的用户和事件,生成几个合成数据集.表 4 列出了合成数据集中用户数量和事件数量的变化情况,其中加粗的为默认值.

本实验对比本文的算法 1~算法 3 以及文献[6]中的算法.以下,算法 1 记为事件稳定规划,算法 2 记为用户稳定规划,算法 3 记为改进稳定规划,文献[6]中的算法记为单边规划.算法 1~算法 3 均经过第 4.3 节空间剪枝方法的预处理.在真实数据集上,测试以上 4 个算法的总效用值、运行时间、所需内存以及“单边规划”算法所得结果

中阻塞对数量所占的百分比.在可扩展性测试中,测试以上 4 种算法在用户数量 n 和事件数量 m 变化时,运行时间和总效用值的变化情况.在无特殊说明的情况下, n 和 m 的值取默认值.

Table 4 Synthetic datasets

表 4 合成数据集

参数名称	设置值
n	20, 50 , 100, 200, 500
m	200, 500, 1000, 5000

5.2 真实数据集上的实验结果

4 种算法在真实数据集上的实验结果见表 5~表 7.表 5 显示各算法在真实数据集上获得的总效用值.表 6 显示单边规划算法在真实数据集上运行结果中阻塞对所占的百分比.表 7 显示各算法在真实数据集上运行所需的时间开销和内存开销.

Table 5 Results of total utility over real datasets

表 5 真实数据集上对总效用值的测试结果

城市名称	单边规划	事件稳定规划	用户稳定规划	改进稳定规划
北京	212.6	218.9	210.3	219.8
奥克兰	2 348.3	2 439.7	2 436.8	2 442.2
夏威夷	11 649.8	1 237.8	12 297.2	12 416.7
阿拉斯加	10 108.5	10 198.3	10 239.6	10 367.3
香港	18 538.2	17 936.5	18 773.2	18 933.2
新加坡	50 175.2	47 682.3	46 811.0	52 927.8
温哥华	31 357.0	69 384.7	69 001.0	70 789.0

Table 6 Results of percentage of block pairs over real datasets

表 6 真实数据集上单边规划阻塞对数量百分比

城市名称	单边规划阻塞对数量百分比(%)
北京	39.31
奥克兰	61.14
夏威夷	32.35
阿拉斯加	26.38
香港	52.32
新加坡	68.73
温哥华	47.43

Table 7 Resultsof time and memory cost over real datasets

表 7 真实数据集上对时间和内存开销的测试结果

城市名称		北京	奥克兰	夏威夷	阿拉斯加	香港	新加坡	温哥华
单边规划	计算时间(s)	0.25	0.37	2	10	5	37	114
	消耗内存(MB)	98.72	2 543.3	12 347.9	11 325.4	18 970.3	47 250.8	69 472.7
事件稳定规划	计算时间(s)	0.13	1	64	214	383	10 088	43 931
	消耗内存(MB)	96.10	0.18	0.02	23.40	40.95	368.43	1 630.93
用户稳定规划	计算时间(s)	0.21	2.47	85.63	227	452	10 458	44 295
	消耗内存(MB)	97.85	0.18	0.02	25.40	41.37	379.46	1 689.25
改进稳定规划	计算时间(s)	0.09	0.15	12	33	27	395	1 436
	消耗内存(MB)	104.35	0.29	0.18	42.50	51.37	368.43	1 689.25

由表 5 可以看出:改进稳定规划算法所获得的总效用值是最高的,在温哥华数据集上,其效用值甚至比单边规划算法高出 2 倍以上.事件稳定规划算法和用户稳定规划算法所获得的总效用值在数据集北京、香港、新加坡上比单边规划低.这说明现有的单边规划算法不考虑事件与用户双方的效用偏好,它得到的总效用值是比较随机的.而考虑了双边效用偏好的算法在大部分情况下,也可以获得较高的效用值.事件稳定规划算法和用户稳定规划两个算法偶尔得到效用值低的情况,是它们在双方均取较低效用值的情况下所达到的稳态规划.而改进

算法较好地规避了这一情况,因此在保证稳态规划下,也可以获得较高的总效用值,在数据量大的情况下尤为显著.该实验证明了本文提出稳态规划的合理性和必要性.

由表 6 可以看出:单边规划算法在真实数据集上所得到的计算结果中,阻塞对的数量不在少数,在数据集奥克兰和新加坡上,甚至高达 61.14% 和 68.73%,在 3 个稳态规划算法中,阻塞对的数量为 0.根据表 5 和表 6 二者综合的结果可以看出:单边规划算法既不能保证事件和用户双方共同达到满意,在总效用值上也不如考虑双方偏好效用的稳态算法.由此可以进一步的证明,本文提出的稳态规划问题的合理性和必要性.

由表 7 可以看出,单边规划算法和双边稳态算法的内存开销差别不大.而在时间开销上,双边稳态规划算法比单边算法要高一些.这是因为,双边稳态算法需要一定的时间来保障匹配结果中没有阻塞对的存在,因此需要更多的时间开销.另外,虽然本文提出的 3 种算法在时间复杂度上是一样的,但平均来说,改进稳态算法比两个基础算法所花费的时间要低一些.这是因为改进算法同时考虑双方共同的选择,因此当遇到阻塞对出现时,所需调整的空间就小一些.因此,改进算法在实验中的时间开销会比两个基础算法要小,但比单边规划算法要大一些.

5.3 合成数据集上的实验结果

本节分析本文提出的 3 种算法在合成数据集上的可扩展性,并与单边规划算法进行比较.实验结果如图 3 所示.当用户数量 n 变化时, m 取默认值 5 000;当事件数量 m 变化时, n 取默认值 50.

图 3 所示为单边规划算法在合成数据集上阻塞对所占的百分比,可以看出:随着 n 与 m 变大,单边规划算法在合成数据集上阻塞对所占的百分比基本上在增高,当然也存在下降的情况.这是因为单边规划算法完全不考虑双方稳态平衡的问题,产生阻塞对的情况基本上是随机的.而随着数据量的增大,产生阻塞对的概率就会增大,因此阻塞对百分比基本会呈现上升趋势.

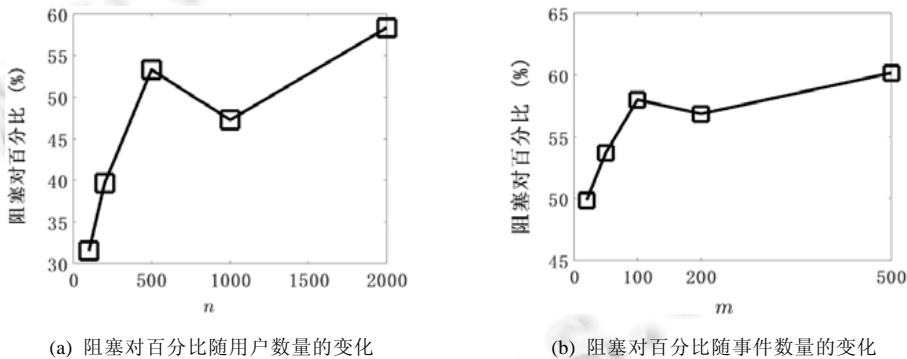


Fig.3 Percentage of block pairs in the results of unilateral planning algorithms over synthetic datasets

图 3 单边规划算法在合成数据集上阻塞对所占的百分比

图 4 所示为时间开销随用户数量和事件数量的变化情况,从图中可以看出,4 种算法的时间开销均随用户数量和事件数量的增加而增加.该现象显然是合理的,因为随着数据量的增加,算法在检索遍历事件和用户之间效用值的过程所耗费的时间就要增加.这是因为双边稳态算法需要一定的时间来保障匹配结果中没有阻塞对的存在,因此需要更多的时间开销.另外,单边规划的时间开销比较低,随着用户和事件数量的增大而增加的幅度也比较小.事件稳态规划和用户稳态规划两种基础算法所耗费的时间最长,二者之间的差别不大.改进稳态规划的时间开销介于单边规划和两个基础算法之间.这是因为改进算法同时考虑双方共同的选择,因此当遇到阻塞对出现时,所需调整的空间就小一些.

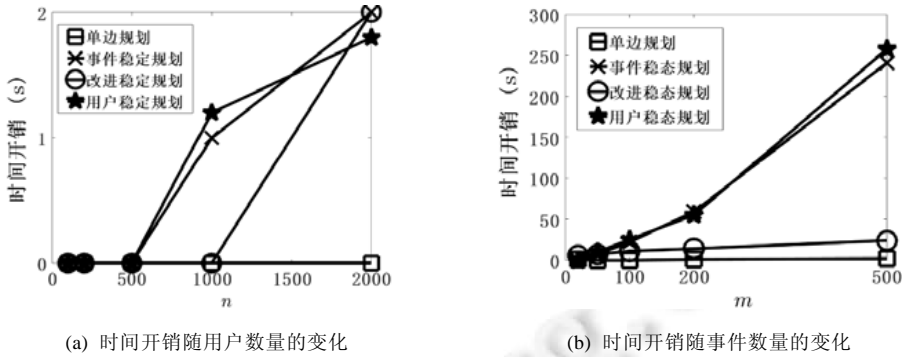


Fig.4 Time cost with reference to the number of users and events
图 4 时间开销随用户数量和事件数量的变化

图 5 所示为 4 种算法的总效用值随用户数量和事件数量的变化情况,可以看出,4 种算法的总效用值均随用户数量和事件数量的增加而增加.该现象显然是合理的,因为随着数据量的增加,形成匹配的用户和事件的数量就会增加,因此总体的效用值必然增加.改进稳定规划算法所获得的总效用值是最高的;事件稳定规划算法和用户稳定规划算法所获得的总效用值次之,但并不明显;单边规划的总效用值最低.由此可见:现有的单边规划定义和算法无论在考虑事件和用户的双方满意度上,还是总体效用值上,均不如考虑双边稳态的定义和算法达到的效果好,因此也佐证了本文提出的双边稳态问题的合理性.

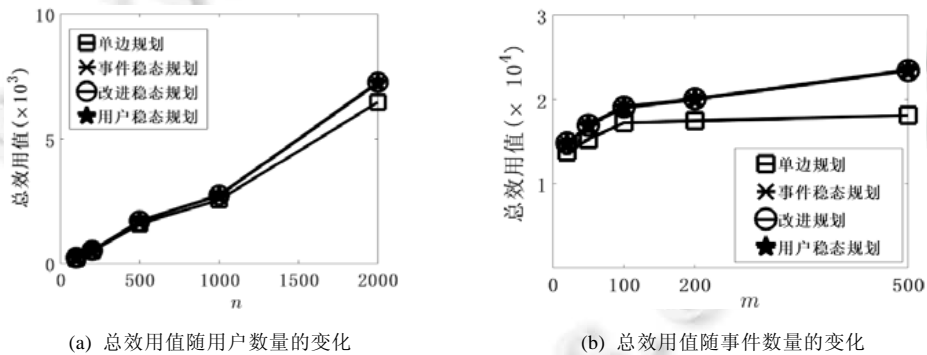


Fig.5 Total utility with reference to the number of users and events
图 5 总效用值随用户数量和事件数量的变化

6 结论及展望

本文重点研究社交网络中为用户规划感兴趣事件这一经典问题,从用户和事件两个角度进行偏好考虑,提出一种双边偏好稳态规划算法,解决了社交网络中这一双向选择难题.为了验证该双边偏好算法的有效性,设计了两种基础算法与一种改进算法,并通过实验进行了对比.实验结果显示,双边稳态规划对比现有的单边规划无论从总效用值上,还是满足事件与用户双方偏好上,均没有双边稳态规划的效果好.从而为主办者和参与者提供了一种高效、高质量的双向匹配事件推送模式,有效解决了这一经典问题中的双向选择难题,有助于提高社交平台的用户满意度与价值提升.

未来工作中,可以考虑事件与用户中各个参数的动态变化的可能,提供支持动态变化的双边稳态规划算法.

References:

- [1] Liu X, He Q, Tian Y, Lee WC, McPherson J, Han J. Event based social networks: Linking the online and offline social worlds. In: Proc. of the 18th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. Beijing: ACM Press, 2012. 1032–1040.
- [2] Cheng Y, Yuan Y, Chen L, Giraud-Carrier C, Wang G. Complex event-participant planning and its incremental variant. In: Proc. of the 2017 IEEE 33rd Int'l Conf. on Data Engineering. San Diego: IEEE, 2017. 859–870.
- [3] She J, Tong Y, Chen L, Cao CC. Conflict-aware event participant arrangement. In: Proc. of the 2015 IEEE 31st Int'l Conf. on Data Engineering. Seoul: IEEE, 2015. 735–746.
- [4] She J, Tong Y, Chen L, Cao CC. Conflict-aware event-participant arrangement and its variant for online setting. IEEE Trans. on Knowledge and Data Engineering, 2016,28(9):2281–2295.
- [5] Li K, Lu W, Bhagat S, Lakshmanan LV, Yu C. On social event organization. In: Proc. of the 20th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. New York: ACM Press, 2014. 1206–1215.
- [6] She J, Tong Y, Chen L. Utility-aware social event-participant planning. In: Proc. of the 2015 ACM SIGMOD Int'l Conf. on Management of Data. Melbourne: ACM Press, 2015. 1629–1643.
- [7] Li SJ, Zhao HY. Foundation of Economics. Beijing: Tsinghua University Press, 2015 (in Chinese).
- [8] Minkov E, Charrow B, Ledlie J, Teller S, Jaakkola T. Collaborative future event recommendation. In: Proc. of the 19th ACM Int'l Conf. on Information and Knowledge Management. Toronto: ACM Press, 2010. 819–828.
- [9] Liao G, Zhao Y, Xie S, Yu PS. An effective latent networks fusion based model for event recommendation in offline ephemeral social networks. In: Proc. of the 22nd ACM Int'l Conf. on Information & Knowledge Management. San Francisco: ACM Press, 2013. 1655–1660.
- [10] Khrouf H, Troncy R. Hybrid event recommendation using linked data and user diversity. In: Proc. of the 7th ACM Conf. on Recommender Systems. Hong Kong: ACM Press, 2013. 185–192.
- [11] Sun YC, Chen CC. A novel social event recommendation method based on social and collaborative friendships. In: Proc. of the Int'l Conf. on Social Informatics. Kyoto: Springer-Verlag, 2013. 109–118.
- [12] Chen C, Zhang D, Guo B, Ma X, Pan G, Wu Z. Trippanner: Personalized trip planning leveraging heterogeneous crowdsourced digital footprints. IEEE Trans. on Intelligent Transportation Systems, 2015,16(3):1259–1273.
- [13] Lu EHC, Chen CY, Tseng VS. Personalized trip recommendation with multiple constraints by mining user check-in behaviors. In: Proc. of the 20th Int'l Conf. on Advances in Geographic Information Systems. Redondo Beach: ACM Press, 2012. 209–218.
- [14] Zhang W, Wang J, Feng W. Combining latent factor model with location features for event-based group recommendation. In: Proc. of the 19th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. Chicago: ACM Press, 2013. 910–918.
- [15] Du R, Yu Z, Mei T, Wang Z, Wang Z, Guo B. Predicting Activity Attendance in Event-based Social Networks: Content, Context and Social Influence. Seattle: ACM Press, 2014. 425–434.
- [16] Pham TAN, Li X, Cong G, Zhang Z. A general graph-based model for recommendation in event-based social networks. In: Proc. of the 2015 IEEE 31st Int'l Conf. on Data Engineering. Seoul: IEEE, 2015. 567–578.
- [17] Feng K, Cong G, Bhowmick SS, Ma S. In search of influential event organizers in online social networks. In: Proc. of the 2014 ACM SIGMOD Int'l Conf. on Management of Data. Snowbird: ACM Press, 2014. 63–74.
- [18] Kempe D, Kleinberg J, Tardos E. Maximizing the spread of influence through a social network. In: Proc. of the 9th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. Washington: ACM Press, 2003. 137–146.
- [19] Lappas T, Liu K, Terzi E. Finding a team of experts in social networks. In: Proc. of the 15th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. Paris: ACM Press, 2009. 467–476.
- [20] Iwama K, Miyazaki S. A survey of the stable marriage problem and its variants. In: Proc. of the Int'l Conf. on Informatics Education and Research for Knowledge-Circulating Society. Kyoto: IEEE, 2008. 131–136.
- [21] Gusfield D, Irving RW. The Stable Marriage Problem: Structure and Algorithms. Cambridge: MIT Press, 1989.
- [22] Malhotra VS. On the stability of multiple partner stable marriages with ties. In: Proc. of the European Symp. on Algorithms. Berlin: Springer-Verlag, 2004. 508–519.
- [23] Kujansuu E, Lindberg T, Mäkinen E. The stable room mates problem and chess tournament pairings. Divulgaciones Matemáticas, 1999,7(1):19–28.

- [24] Qiao SJ, Han N, Zhou JL, Li RH, Jin CQ, Gutierrez LA. SocialMix: A familiarity-based and preference-aware location suggestion approach. *Engineering Applications of Artificial Intelligence*, 2018,68:192–204.
- [25] Qiao SJ, Han N, Gao YJ, Li RH, Huang JB, Guo J, Gutierrez LA, Wu XD. A fast parallel community discovery model on complex networks through approximate optimization. *IEEE Trans. on Knowledge and Data Engineering*, 2018,30(9):1638–1651.
- [26] Qiao SJ, Han N, *et al.* TraPlan: An effective three-in-one trajectory prediction model in transportation networks. *IEEE Trans. on Intelligent Transportation Systems*, 2015,16(3):1188–1198.
- [27] Qiao SJ, Shen DY, Wang XT, Han N, Zhu W. A self-adaptive parameter selection trajectory prediction approach via hidden Markov models. *IEEE Trans. on Intelligent Transportation Systems*, 2015,16(1):284–296.

附中文参考文献:

- [7] 李世杰,赵岩红.经济学基础.北京:清华大学出版社,2015.



成雨蓉(1989—),女,山西清徐人,博士,CCF 专业会员,主要研究领域为图数据查询处理与分析,时空众包数据分析,社交网络数据分析.



李博扬(1992—),男,学士,主要研究领域为社交网络数据分析,时空数据分析,机器学习.



王国仁(1966—),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为不确定数据管理,数据密集型计算,可视媒体数据管理与分析,非结构化数据管理,分布式查询处理与优化技术(主要包括传感器网络和 P2P 对等计算),生物信息学.



袁野(1981—),男,博士,教授,博士生导师,CCF 专业会员,主要研究领域为云计算,大数据管理(包括图数据管理,不确定数据管理,数据隐私保护),P2P 计算.