9 Output F

给定对齐网络 G 和 G'以及初始匹配用户关系 F_0,MCS_INA 的输出是包含 F_0 的用户对应关系 F.首先,算法利用初始匹配用户集合 F_0 初始化输出集合 F(第 1 行),之后,迭代地利用已匹配用户识别其他用户.在每次迭代过程中,主要分为两步:第 1 步,分别从 G 和 G'中选取识别度较高的候选匹配用户集合 C,C'(第 3 行~第 4 行),为降低计算复杂度,该步骤分别针对两个网络 G,G'单独进行处理,选取与已匹配用户关系较近的用户集合;第 2 步,分别针对候选匹配用户集合 C,C'进行用户匹配,借鉴最大公共子图思想,构建匹配用户映射关系 M:$V\to C'$和 M':$V'\to C$(第 5 行~第 6 行),并将 M 与 M'重叠的部分作为匹配结果(第 7 行),加入到输出集合 F 中,并执行下一次循环,若连续两次迭代匹配结果 F 未发生改变,则停止迭代,将 F 作为算法的输出.

本文在第 4.1 节中将深入讨论候选用户集合选取问题;在第 4.2 节中将深入讨论候选用户集合匹配问题.

4.1 候选匹配用户选取策略

为便于描述,本节仅针对 G 中候选匹配用户选取问题进行讨论.给定网络 $G(V,E)$以及已匹配用户映射关系 F,候选匹配用户选取算法 $Candidate(G,F)$的目标是,在 G 中选取与匹配用户集合 V_F 关系紧密的用户群体 C.结合最大公共子图思想,对于用户 k,构建其代价函数如下:

$$\Delta Sco_E(k)=\sum_{i\in V_{\bar{F}}}\Pr(V_{i,k},V'_{F(i),k'})-\alpha\Pr(V_{i,k},\neg V'_{F(i),k'})-\alpha\Pr(\neg V_{i,k},V'_{F(i),k'}) \tag{4}$$

其中,k'为与 k 匹配的用户;$\Delta Sco_E(k)$表示通过匹配用户 k,最终匹配结果 Sco 值提升幅度的期望.由于 k'未知,因此下面着重讨论如何估计$\Delta Sco_E(k)$的取值.为方便起见,本文假设 V_F 中用户以及 k'均匹配正确,即 $\bar{k}=\bar{k}'$, 且对于任意 $i\in V_F$,有 $\bar{i}=\overline{F(i)}$.

对于用户 $i\in V_F$,由于 $\bar{i}=\overline{F(i)}$ [0,1], $\bar{k}=\bar{k}'$, 有 $\Pr(V_{i,k},V'_{F(i),k'})=S_{E'}\Pr(V_{i,k})$,$\Pr(V_{i,k},\neg V'_{F(i),k'})=(1-S_{E'})\Pr(V_{i,k})$, $\Pr(\neg V_{i,k},V'_{F(i),k'})=\Pr(V^*_{\bar{i},\bar{k}})(1-S_E)S_{E'}$,其中,$\Pr(V^*_{\bar{i},\bar{k}})$ 表示 $\bar{i}$ 与 $\bar{k}$ 之间有边的概率,$\Pr(V^*_{\bar{i},\bar{k}})=\Pr(V_{i,k})/S_E$,因此,公式(4)可化简为

$$\Delta Sco_E(k)=\sum_{i\in V_{\bar{F}}}S_{E'}\Pr(V_{i,k})-\alpha(1-S_{E'})\Pr(V_{i,k})-\alpha\Pr(V_{i,k})(1-S_E)S_{E'}/S_E \tag{5}$$

公式(5)中,S_E(本文仅对 S_E 进行分析,$S_{E'}$同理)、$\Pr(V_{i,k})$、α均为未知参数,下面将对这些参数进行分析估计.

首先,S_E 表示网络对齐模型中网络 G^*中的边保留到 G 的概率,可通过以下公式进行估计:

$$S_E=\sum_{i\in V_F,j\in V_F}V_{i,j}V'_{F(i),F(j)}\Big/\sum_{i\in V_F,j\in V_F}V'_{F(i),F(j)} \tag{6}$$

其中,分子部分 $\sum_{i\in V_F,j\in V_F}V_{i,j}V'_{F(i),F(j)}=\sum_{i\in V_F,j\in V_F}\Pr(V^*_{\bar{i},\bar{j}})S_{E'}S_E$ 为 G 与 G'中重叠边数量,分母部分 $\sum_{i\in V_F,j\in V_F}V'_{F(i),F(j)}=\sum_{i\in V_F,j\in V_F}\Pr(V^*_{\bar{i},\bar{j}})S_{E'}$ 为 G'中已匹配用户间边的数量.

其次,$\Pr(V_{i,k})$表示 G 中用户 i 与 k 之间存在边的可能性,可通过 i,k 之间的间接关系进行预测,本文认为 $\Pr(V_{i,k})$与用户 i 与 k 的共同邻居数量相关.

$$\Pr(V_{i,k})=\sum_{p\in V,q\in V}\left(\delta(p,q,|N(i)\cap N(k)|)V_{p,q}\right)\Big/\sum_{p\in V,q\in V}\delta(p,q,|N(i)\cap N(k)|) \tag{7}$$

其中,$N(i)$表示用户 i 的邻居集合,$\delta(p,q,|N(i)\cap N(k)|)$表示用户对 p,q 的共同邻居数量是否等于$|N(i)\cap N(k)|$的判断.若相等,则$\delta(p,q,|N(i)\cap N(k)|)$取值为 1;否则,$\delta(p,q,|N(i)\cap N(k)|)$取值为 0.公式(7)的分子部分表示网络中公共邻居数量为$|N(i)\cap N(k)|$且存在边的节点对数量,分母为网络中公共邻居数量为$|N(i)\cap N(k)|$的节点对数量.显然,若网络中大部分公共邻居数量为$|N(i)\cap N(k)|$的节点对之间存在边,则 $\Pr(V_{i,k})$取值应该较高,否则,$\Pr(V_{i,k})$取值较低.

最后,对于α,依据定理 1,当 $\alpha=\mathrm{Min}(S_V^2S_E,S_{V'}^2S_{E'})\big/(2-2\mathrm{Min}(S_V^2S_E,S_{V'}^2S_{E'}))$ 时,可有效区分匹配用户与非匹配用户.然而,由于参数 S_V 与 $S_{V'}$无法预先确定,故无法准确估计α的取值.因此,在实验过程中,本文逐渐降低α取值,优先计算α较大时的匹配用户.本文令$\alpha=\beta\mathrm{Min}(S_E,S_{E'})/(2-2\beta\mathrm{Min}(S_E,S_{E'}))$,初始情况下,$\beta$=1,随着迭代的进行,逐渐降低$\beta$的取值.当$\beta=(|V_F|/\mathrm{Max}\{|V|,|V'|\})^2$ 时,$\alpha\leqslant\mathrm{Min}(S_V^2S_E,S_{V'}^2S_{E'})\big/(2-2\mathrm{Min}(S_V^2S_E,S_{V'}^2S_{E'}))$, 已小于最优取值,则迭代停止.由此,候选匹配选取算法 $Candidate(G,G',\mathrm{F})$如算法 2 所示.

算法 2. *Candidate*(G,G',F).

输入:G,G',F;

输出:C.

```
1  Compute S_E,S_E',β=1
2  Tmp←∅
3  For i∈V_F do
4     Tmp←N(i)
5  End For
6  For β∈[(|V_F|/Max{|V|,|V'|})^2,1]
7     Compute α
8     For i∈Tmp−V_F do
9        Compute ΔSco_E(i)
10       If ΔSco_E(i)>0
11          C.put(i)                    //C is ordered by ΔSco_E
12       End If
13    End For
14    If C is empty
15       β←β−0.1
16    End If
17 End For
18 Output C
```

在算法 2 中,首先,利用公式(6)对参数 S_E,$S_{E'}$进行估计,并赋值 β=1(第 1 行).之后,利用已识别用户集合 V_F,获取待分析的候选匹配用户集合 *Tmp*(第 2 行~第 5 行),从第 6 行开始,迭代地分析 *Tmp* 中用户是否适合作为候选匹配用户.若对于用户 $i\in Tmp-V_F$,其$\Delta Sco_E(i)>0$,则认为用户 i 适合作为候选匹配用户,并将其放入候选匹配用户集合 C 中(第 9 行~第 12 行),并输出结果集 C.若结果集 C 为空集,则有可能参数β取值过高,降低参数β,并重新计算候选匹配用户集合(第 15 行).在算法 2 中,通过迭代降低参数β,可有效提高算法识别精度,初始情况下,β取值为 1,相对应的α取值较高,候选匹配用户集合选取相对严格.而随着迭代的进行,β取值逐渐降低,进而α取值随之降低,候选匹配用户集合选取逐渐宽松,最终,当β取最低值时,若候选匹配用户集合依然为空,则无适合匹配的用户.

通过算法 2,可获取有序的候选匹配用户集合 C,集合 C 中用户依据与已匹配用户之间的关系强度进行排序,与已匹配用户关系紧密的用户具有较高排名,相反地,关系疏远的用户具有较低排名.另外,对于每个候选匹配用户,计算其代价函数的时间复杂度为$O(D^2)$.因此,在 MCS_INA 中,候选匹配选取算法 *Candidate*(G,G',F)的时间复杂度为 $O(D^2|V|+D'^2|V'|)$.

4.2 用户匹配策略

给定 G 中候选匹配用户集合 C,用户匹配算法 *Match*(G,G',F,C)的目标是,构建映射函数 $M':V'\to C$.由于最大公共子图问题为 NP 完全问题,为降低计算复杂度,本节利用第 4.1 节中候选匹配用户排名,结合贪婪思想,提出近似求解算法.

对于候选匹配用户集合 C 中任意用户 $k\in C$,令 k'为 G'中未匹配用户,借鉴最大公共子图思想(见公式(1)),则 k'与 k 的匹配度可表示为

$$\Delta Sco(k,k')=\sum_{i\in V_F}V_{i,k}V'_{F(i),k'}-\alpha\left|V_{i,k}-V'_{F(i),k'}\right| \tag{8}$$

公式(8)表示,若匹配用户 k 与 k',可提升匹配结果 *Sco* 得分$\Delta Sco(k,k')$.

至此,用户匹配算法 $Match(G,G',F,C)$如算法 3 所示.

算法 3. $Match(G,G',F,C)$.

输入:G,G',F,C;

输出:M.

```
1  M←∅
2  For k∈C                    //C is ordered by ΔSco_E
3     Tmp←∅, k'←null
4     For i∈N(k)∩V_F
5        Tmp←N(F(i))
6     End For
7     For t'∈Tmp−V'_F
8        If ΔSco(k,t')>ΔSco(k,k') & ΔSco(k,t')>0
9           k'←t'
10       End If
11    End For
12    M←(k,k')
13 End For
14 Output M
```

算法 3 中采用贪婪思想有序地对用户集合 C 中用户进行识别,优先识别与已匹配用户关系紧密的用户,可有效降低识别错误的发生.

在算法 3 中,对于每个候选匹配用户 k,其对应的 Tmp 集合中用户的个数为 $O(DD')$,而对于 Tmp 中每个用户 t',计算 k 与 t'匹配度的时间复杂度为 $O(D+D')$,因此,识别每个候选匹配用户 k 的时间复杂度为 $O(DD'(D+D'))$,且在 MCS_INA 中,用户匹配算法 $Match(G,G',F,C)$的时间复杂度为 $O(DD'(D+D')(|V|+|V'|))$.

综上,算法 MCS_INA 的时间复杂度为 $O(DD'(D+D')(|V|+|V'|))$.

5 实 验

5.1 实验环境

实验环境:本文采用 Java 编程语言实现相关算法,实验主机采用 Intel i5-4590 处理器,主频 3.30GHz,8GB 内存,操作系统为 64 位 Windows 7.

数据集:本文所使用数据集见表 1.首先,Facebook 表示匿名化的真实 Facebook 数据集,其中,第 1 个网络(FL)为 Facebook 新奥尔良市用户关系网络,另一个网络(FW)为 Facebook 新奥尔良市用户信息墙交互网络,其中,重叠的用户数量为25 538,重叠的边的数量为151 580,该数据集可参考文献[25];其次,本文利用真实社交网络生成部分数据集,其中,Twitter 表示真实 Twitter 用户数据集,$T_{1.0}$ 表示原始的 Twitter 数据规模,$T_{0.8}$ 表示 $T_{1.0}$ 中 80%的边以及 80%的节点被保留到数据集 $T_{0.8}$ 中,$T_{0.7}$ 和 $T_{0.9}$ 同理.在生成 $T_{0.7}$、$T_{0.8}$ 和 $T_{0.9}$ 时,均采用随机概率保留点和边.另外,本文采用不同随机图生成算法生成合成数据集 ER 和 PA,其中,ER 表示该网络分布满足 ER 随机图模型[23],PA 表示该网络节点关系分布满足幂律分布[26],所有随机图均通过 igraph 生成.最后,本文随机地选取匹配用户作为已知匹配用户,该方式将有效降低识别瓶颈的发生,若已知匹配用户集中于单一社区内,将造成社区外部节点识别准确率的下降.

对比算法:由于启发式的解决方法适用性较低,与本文研究内容有差异;基于网络表示的网络对齐算法,需要大量训练数据,而本文方法仅基于预先匹配的少量用户节点数量(占比通常为 10%以下),两种方法环境不同.为此,本文仅选取与本文研究方法密切相关的两种经典算法 CN 和 CNR 进行比较:(1) CN[8]:CN 算法为迭代算法,每次迭代过程中,选取共同邻居数量最多的用户对作为匹配用户;(2) CNR[9]:与 CN 算法类似,但 CNR 算法在

每次迭代过程中优先匹配度数较高的用户;(3) MCS_INA:本文提出的算法.

效果评价:本文采用准确率(precision)、召回率(recall)、*F*-measure 以及运行时间(runtime)这 4 个方面进行评估.

Table 1 Datasets

表 1 数据集

数据集		节点数	边数
Facebook	FL (FacebookLinks)	49 247	797 470
	FW (FacebookWall)	25 622	157 236
Twitter	$T_{1.0}$	11 473	191 626
	$T_{0.9}$	10 331	138 970
	$T_{0.8}$	9 149	97 360
	$T_{0.7}$	8 008	66 578
ER		20 000	400 000
PA		20 000	399 791

5.2 真实数据集中的实验效果

首先,为比较不同算法在不同数据集中的识别准确率,该组实验采用 FL&FW、$T_{0.7}$&$T_{0.8}$、$T_{0.7}$&$T_{0.9}$ 和 $T_{0.8}$&$T_{0.9}$ 这 4 个数据集,对于每组数据集,随机地选取 10%的匹配用户作为已知,实验结果如图 2 所示.在 3 种不同算法中,本文算法 MCS_INA 的准确率最高,而 CN 算法准确率最低.另外,对比 Twitter 的 3 组数据集,3 种算法在数据集 $T_{0.8}$&$T_{0.9}$ 上具有较高准确率,而在 $T_{0.7}$&$T_{0.8}$ 数据集上具有较低准确率.原因是,$T_{0.8}$&$T_{0.9}$ 重叠用户数量以及重叠边较多,期望情况下其重叠边为 $T_{1.0}$ 边数量的 37%,而 $T_{0.7}$&$T_{0.8}$ 的重叠边比率仅为 17%.因此,$T_{0.8}$&$T_{0.9}$ 相对更容易识别.

其次,对比不同算法在不同数据集中的召回率,如图 3 所示.在 3 种算法中,本文算法 MCS_INA 的召回率依然最高,其次为 CNR,算法 CN 召回率最低;对比图 2 中的准确率,算法 CNR 在数据集 $T_{0.7}$&$T_{0.8}$、$T_{0.7}$&$T_{0.9}$、$T_{0.8}$&$T_{0.9}$ 中的召回率略高于准确率,这是由于 Twitter 数据集中包含大量单独存在于单个网络中的用户,算法 CNR 错误地识别了这一部分用户.而在数据集 FL&FW 中,算法 CNR 的准确率略高于召回率,这是由于 FW 数据集几乎为 FL 数据集的子集.

最后,综合准确率与召回率,*F*-measure 的比较结果如图 4 所示,MCS_INA 的综合性能明显优于算法 CN 和 CNR.

为测试不同算法在不同数据集中的运行时间,本组实验记录了不同算法的运行时间,见表 2.由表 2 可知,算法 CN 的运行时间最短,其次为 MCS_INA,CNR 的运行时间最长.虽然算法 CN 具有最短的运行时间,但综合图 4 中 *F*-measure 的比较结果来看,MCS_INA 依然具有最高的综合性能.另外,对于算法 CNR,无论从算法执行时间还是算法精度,MCS_INA 均优于 CNR.

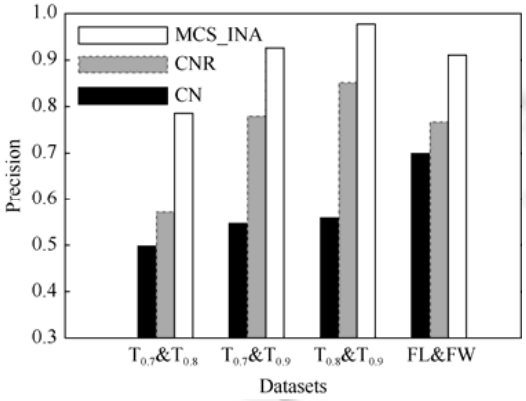

Fig.2 Precision on real-world datasets

图 2 真实数据集中的准确率比较结果

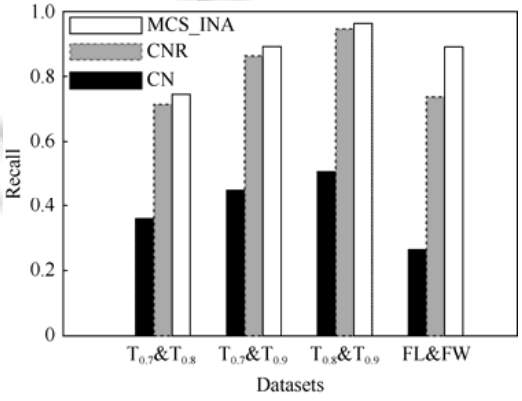

Fig.3 Recall on real-world datasets

图 3 真实数据集中的召回率比较结果

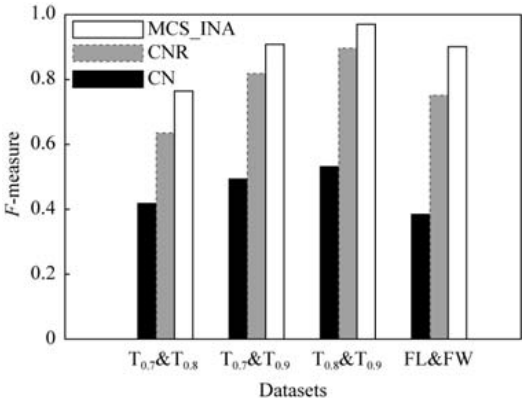

Fig.4 *F*-measure on real-world datasets

图 4 真实数据集中的 *F*-measure 比较结果

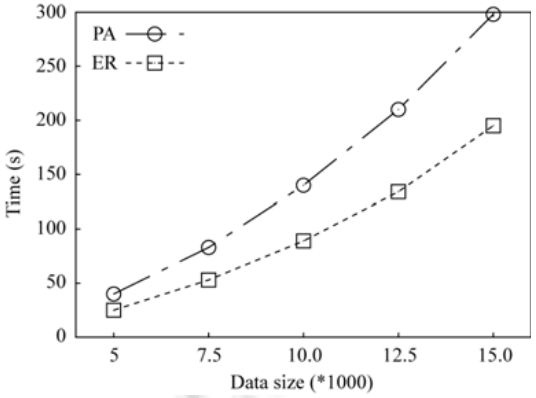

Fig.5 Running time regarding the nodes

图 5 运行时间随节点数变化实验图

Table 2 Runnin time on real-world datasets (min)

表 2 真实数据集中运行时间比较结果 (分钟)

	$T_{0.7}$&$T_{0.8}$	$T_{0.7}$&$T_{0.9}$	$T_{0.8}$&$T_{0.9}$	FL&FW
CN	0.53	0.58	0.65	2.43
CNR	1.32	1.95	2.27	10.21
MCS_INA	0.95	1.22	1.43	7.55

5.3 合成数据集中的实验效果

在第 5.2 节中,本文针对真实数据集进行了实验,虽然在真实数据集中算法 MCS_INA 具有较优性能,但并不代表在所有数据集中算法 MCS_INA 均表现优异,为此,在第 5.3 节中,本文利用不同类型的合成数据集,测试算法的性能.

1) MCS_INA 在不同类型网络中的性能实验

为验证算法 MCS_INA 在不同类型数据集中的表现,本节分别在 ER 数据集与 PA 数据集中测试 MCS_INA 算法的性能,见表 3 和表 4.以 ER 数据集为例,数据集 $ER_{0.5}$、$ER_{0.6}$、$ER_{0.7}$、$ER_{0.8}$、$ER_{0.9}$ 分别表示从数据集 ER 中以概率[0.5,0.6,0.7,0.8,0.9]提取点和边,对于每组数据集,本实验选取 10%的用户作为已知匹配用户.由表 3 和表 4 可知,当数据集重叠部分较大时($ER_{0.6}$&$ER_{0.7}$、$ER_{0.7}$&$ER_{0.8}$、$ER_{0.8}$&$ER_{0.9}$、$PA_{0.7}$&$PA_{0.8}$、$PA_{0.8}$&$PA_{0.9}$),MCS_INA 具有较高的准确率与召回率,而当数据集重叠部分较小时($ER_{0.5}$&$ER_{0.6}$、$PA_{0.5}$&$PA_{0.6}$、$PA_{0.6}$&$PA_{0.7}$),MCS_INA 具有较低的准确率与召回率,其原因是,当数据集重叠部分较小时,非匹配用户之间相似性相对较强,从而错误地匹配非匹配用户,降低了准确率与召回率.另外,对比表 3 与表 4,MCS_INA 在 ER 数据集中的表现明显强于 PA 数据集,其原因是,ER 数据集中用户之间相似程度较低,而 PA 数据集中,尤其是度数较低用户之间,相似程度较高,当删除部分用户以后,MCS_INA 错误地将这些相似度较高的用户进行匹配,从而降低了准确率与召回率.

Table 3 Performance of MCS_INA on synthetic ER datasets

表 3 MCS_INA 在合成 ER 数据集中的运行结果

	$ER_{0.5}$&$ER_{0.6}$	$ER_{0.6}$&$ER_{0.7}$	$ER_{0.7}$&$ER_{0.8}$	$ER_{0.8}$&$ER_{0.9}$
Precision	0.49	0.98	1.0	1.0
Recall	0.38	0.96	0.99	1.0
F-measure	0.42	0.97	0.99	1.0

Table 4 Performance of MCS_INA on synthetic PA datasets

表 4 MCS_INA 在合成 PA 数据集中的运行结果

	$PA_{0.5}$&$PA_{0.6}$	$PA_{0.6}$&$PA_{0.7}$	$PA_{0.7}$&$PA_{0.8}$	$PA_{0.8}$&$PA_{0.9}$
Precision	0.27	0.81	0.95	0.98
Recall	0.35	0.87	0.96	0.99
F-measure	0.30	0.83	0.95	0.98

2) MCS_INA 运行时间随网络规模变化的实验

为测试 MCS_INA 运行时间随网络规模的变化趋势,本实验利用 ER 与 PA 数据集进行实验.首先,固定合成网络平均度数为 15,变化网络中节点数量,生成不同的原始网络.之后,采用参数 S_E=$S_{E'}$=S_V=$S_{V'}$=0.8,生成对齐网络,实验结果如图 5 所示.横轴表示生成网络中节点数量,纵轴表示算法运行时间,可知,随着网络中节点数量的增多,MCS_INA 算法的运行时间随网络中节点数量的增加基本呈线性增长.然后,固定合成网络节点数量为 5 000,变化网络中平均节点度数,生成不同的原始网络.之后,采用参数 S_E=$S_{E'}$=S_V=$S_{V'}$=0.8,生成对齐网络,实验结果如图 6 所示.通过实验可知,MCS_INA 算法的运行时间随网络中节点度数的增加呈指数型增长,且 MCS_INA 处理 ER 数据集的能力要高于处理 PA 数据集的能力.

3) MCS_INA 性能随已知匹配用户数量变化的实验

本实验数据集采用 $ER_{0.8}$&$ER_{0.8}$,即依据 ER 数据集,采用参数 S_E=$S_{E'}$=S_V=$S_{V'}$=0.8 生成两组不同 $ER_{0.8}$ 并对其进行匹配,本实验随机抽取不同数量百分比的用户作为已知匹配用户对,实验结果如图 7 所示.由图 7 可知,随着已知匹配用户的减少,实验准确率与召回率逐渐降低,当已知匹配用户数量减少至 0.3%时,准确率与召回率实现断崖式降低.这是由于,当已知匹配用户数量降低至 0.3%时,这些已知匹配用户之间几乎不存在直接关系,从而使得 MCS_INA 的准确率与召回率基本降至 0.

4) 参数分析

首先,对自适应参数α进行实验分析,如图 8 所示,1-MCS_INA 表示在每次迭代中不对参数α进行估计,并设定α为 1,同理于 0.5-MCS_INA.该实验分别在 3 个不同数据集 $ER_{0.7}$&$ER_{0.7}$、$ER_{0.8}$&$ER_{0.8}$、$ER_{0.9}$&$ER_{0.9}$ 中运行 MCS_INA、1-MCS_INA 和 0.5-MCS_INA.由图 8 可知,通过自适应调节参数α,在 3 个不同数据集中均取得最优性能.另外,0.5-MCS_INA 的表现优于 1-MCS_INA,其原因是,对于 1-MCS_INA,其节点对相似性函数(见公式(8))中参数α过大,很多匹配用户无法达到阈值,使得召回率降低.

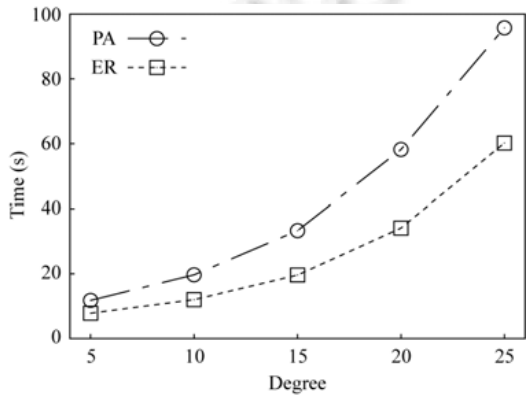

Fig.6 Running time regarding node degree

图 6 运行时间随节点度数变化实验图

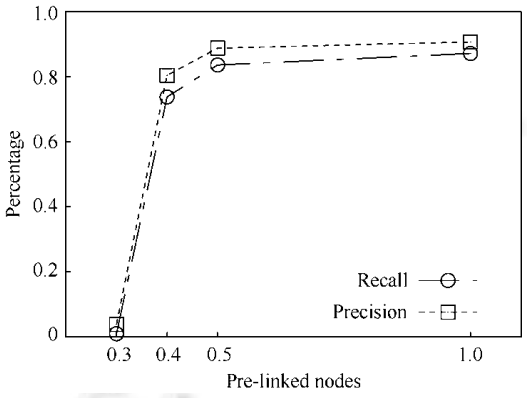

Fig.7 Performance regarding pre-linked nodes

图 7 性能随已知匹配用户变化实验图

其次,针对每次迭代过程中参数α、S_E 和 $S_{E'}$的估计准确性进行分析,本实验采用数据集 $ER_{0.8}$&$ER_{0.8}$,并记录每次迭代过程中 3 个参数值的大小,如图 9 所示.对于参数 S_E 和 $S_{E'}$,其取值随迭代过程逐渐降低,并维持在 0.8 左右.对于参数α,其波动范围较大,在前几次迭代过程中,参数α的取值范围较大,优先对识别度较高的用户进行识别,之后,参数α的取值随迭代过程逐渐降低,并最终稳定在 0.4 左右.通过理论计算参数α可知,当参数α理论取值 0.52 时最优(通过定理 1 可知),之所以会导致实际参数取值与理论取值不一致的情况,是因为在实际情况下,通常有少部分匹配用户,其结构相似性较低,需适量降低参数α的取值.

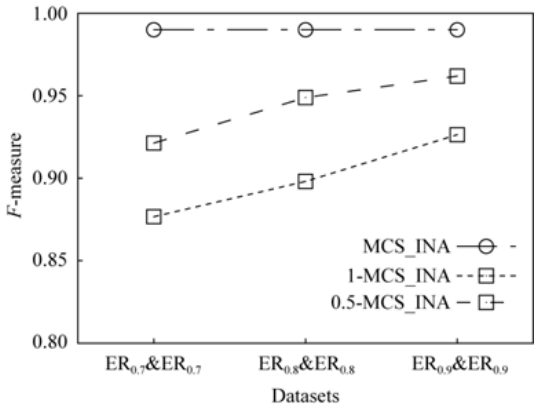

Fig.8 Performance regarding different α

图 8 不同参数α对识别准确性的影响

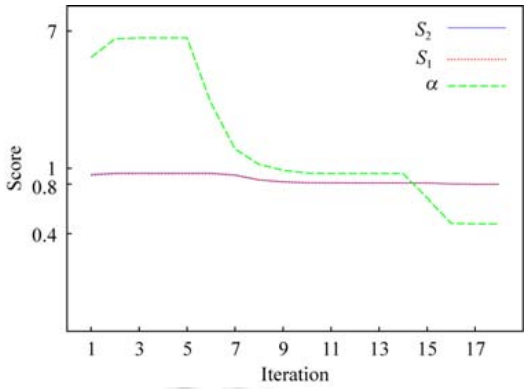

Fig.9 Estimation of the parameters

图 9 实验参数估计

6 结束语

本文主要针对基于用户结构信息的跨社交网络用户识别问题进行研究.首先,借鉴传统最大公共子图问题,提出了求解自适应参数的方法,使得最大公共子图问题可适用于求解不同类型的网络对齐问题;其次,针对最大公共子图计算复杂度过高的问题,本文提出了基于最大公共子图的迭代式网络对齐算法MCS_INA,相比于传统算法,MCS_INA 在每次迭代过程中,仅针对部分候选匹配用户进行匹配,且本文所提出的候选匹配算法有效结合了网络对齐模型,具有严格的理论支持;最后,本文在真实数据集和合成数据集上进行了实验,实验结果表明本文所提出算法具有较高的识别准确率与较低的时间代价.在未来的工作中,将着重针对初始匹配用户过于集中的问题,同时结合用户属性信息、用户行为信息以处理跨网络用户识别问题.

References:

[1] Ding L, Zhou L, Finin T, *et al*. How the semantic Web is being used: An analysis of FOAF documents. In: Proc. of the 38th Annual Hawaii Int'l Conf. on System Sciences. IEEE, 2005. 113c.

[2] Mika P. Flink: Semantic Web technology for the extraction and analysis of social networks. Web Semantics: Science, Services and Agents on the World Wide Web, 2005,3(2-3):211–223.

[3] Zhou X, Liang X, Du X, *et al*. Structure based user identification across social networks. IEEE Trans. on Knowledge and Data Engineering, 2018,30(6):1178–1191.

[4] Raad E, Chbeir R, Dipanda A. User profile matching in social networks. In: Proc. of the 13th Int'l Conf. on Network-Based Information Systems (NBiS). IEEE, 2010. 297–304.

[5] Malhotra A, Totti L, Meira Jr W, *et al*. Studying user footprints in different online social networks. In: Proc. of the 2012 Int'l Conf. on Advances in Social Networks Analysis and Mining (ASONAM 2012). IEEE Computer Society, 2012. 1065–1070.

[6] Kong X, Zhang J, Yu PS. Inferring anchor links across multiple heterogeneous social networks. In: Proc. of the 22nd ACM Int'l Conf. on Information & Knowledge Management. ACM, 2013. 179–188.

[7] Liu S, Wang S, Zhu F, *et al*. Hydra: Large-scale social identity linkage via heterogeneous behavior modeling. In: Proc. of the 2014 ACM SIGMOD Int'l Conf. on Management of Data. ACM, 2014. 51–62.

[8] Yartseva L, Grossglauser M. On the performance of percolation graph matching. In: Proc. of the 1st ACM Conf. on Online Social Networks. ACM, 2013. 119–130.

[9] Korula N, Lattanzi S. An efficient reconciliation algorithm for social networks. Proc. of the VLDB Endowment, 2014,7(5): 377–388.

[10] Feizi S, Quon G, Recamonde-Mendoza M, *et al*. Spectral alignment of graphs. arXiv Preprint arXiv: 1602.04181, 2016.

[11] Islam MS, Liu C, Li J. Efficient answering of why-not questions in similar graph matching. IEEE Trans. on Knowledge and Data Engineering, 2015,27(10):2672–2686.

[12] Zhu Y, Qin L, Yu J X, *et al*. High efficiency and quality: Large graphs matching. The Int'l Journal on Very Large Data Bases, 2013,22(3):345–368.

[13] Zheng R, Li J, Chen H, *et al*. A framework for authorship identification of online messages: Writing-style features and classification techniques. Journal of the American Society for Information Science and Technology, 2006,57(3):378–393.

[14] Bokhari SH. On the mapping problem. IEEE Trans. on Computers, 1981,(3):207–214.

[15] Narayanan A, Shmatikov V. De-anonymizing social networks. In: Proc. of the 30th IEEE Symp. on Security and Privacy. IEEE, 2009. 173–187.

[16] Zhang Z, Gu Q, Yue T, *et al*. Identifying the same person across two similar social networks in a unified way: Globally and locally. Information Sciences, 2017,394:53–67.

[17] Zhou X, Liang X, Zhang H, *et al*. Cross-platform identification of anonymous identical users in multiple social media networks. IEEE Trans. on Knowledge and Data Engineering, 2016,28(2):411–424.

[18] Man T, Shen H, Liu S, *et al*. Predict anchor links across social networks via an embedding approach. In: Proc. of the IJCAI. 2016,16:1823–1829.

[19] Tan S, Guan Z, Cai D, *et al*. Mapping users across networks by manifold alignment on hypergraph. In: Proc. of the AAAI. 2014,14:159–165.

[20] Liu L, Cheung W K, Li X, *et al*. Aligning users across social networks using network embedding. In: Proc. of the IJCAI. 2016. 1774–1780.

[21] Fabiana C, Garetto M, Leonardi E. De-anonymizing scale-free social networks by percolation graph matching. In: Proc. of the 2015 IEEE Conf. on Computer Communications (INFOCOM). IEEE, 2015. 1571–1579.

[22] Kazemi E, Hassani SH, Grossglauser M. Growing a graph matching from a handful of seeds. Proc. of the VLDB Endowment, 2015, 8(10):1010–1021.

[23] Erd P, Rényi A. On random graphs I. Publicationes Mathematicae Debrecen, 1959,6:290–297.

[24] Karrer B, Newman MEJ. Stochastic blockmodels and community structure in networks. Physical Review E, 2011,83(1 Pt 2): 016107.

[25] Viswanath B, Mislove A, Cha M, *et al*. On the evolution of user interaction in Facebook. In: Proc. of the 2nd ACM Workshop on Online Social Networks. ACM, 2009. 37–42.

[26] Barabási AL, Albert R. Emergence of scaling in random networks. Science, 1999,286(5439):509–512.

冯朔(1989－),男,辽宁沈阳人,学士,CCF 学生会员,主要研究领域为社交网络用户识别,网络对齐.

寇月(1980－),女,博士,副教授,CCF 专业会员,主要研究领域为实体搜索,数据挖掘.

申德荣(1964－),女,博士,教授,博士生导师,CCF 高级会员,主要研究领域为分布式数据管理,数据集成.

于戈(1962－),男,博士,教授,博士生导师,CCF 会士,主要研究领域为数据库,大数据管理.

聂铁铮(1980－),男,博士,副教授,CCF 专业会员,主要研究领域为数据质量,数据集成.