基于多目标强化学习的社区隐藏框架

陶海成¹, 卜湛¹, 曹杰^{2*}

1. 江苏省电子商务重点实验室 (南京财经大学), 南京 210023

2. 南京理工大学计算机科学与工程学院, 南京 210094

* 通信作者. E-mail: Jie.Cao@nufe.edu.cn

收稿日期: 2020-07-26; 修回日期: 2020-09-09; 接受日期: 2020-10-27; 网络出版日期: 2021-07-13

国家自然科学基金重点项目 (批准号: 91646204)、国家重点研发计划 (批准号: 2019YFB1405000) 和国家自然科学基金项目 (批准号: 71871109, 71701089) 资助

摘要 在复杂网络研究领域, 社区隐藏算法旨在破坏社区发现算法的有效性, 从而隐藏用户之间的关系, 达到保护用户隐私的目的. 现有的社区隐藏算法寄希望于已有用户改变自己的社交关系实现该目标, 这对用户影响较大, 操作空间比较小. 不同于此类算法, 本文所提的算法从增加节点及其相应边的角度出发, 即伪造用户及其关系, 最大程度降低了对用户的影响, 将社区隐藏问题转换为网络增长问题. 首先基于 ratio association 和 ratio cut, 提出两种更适合社区隐藏的指标; 然后基于强化学习框架, 定义动作空间为不同的网络增长模型, 将两个指标在 l 个阶段的一致性策略值作为网络的状态表示, 并将指标值作为奖赏值; 最后采用两种策略来对指标进行优化, 即给每个 Q 函数赋予权重的标量化多目标 Q -learning 算法以及基于 Pareto 最优算法的多目标 Q -learning 算法. 在真实数据集上的大量实验表明, 相比于现有最新的社区隐藏算法, 本文所提算法展现出更好的有效性.

关键词 社交网络分析, 社区发现, 社区隐藏, 多目标优化, 强化学习

1 引言

在复杂网络研究领域, 社区被抽象定义为满足一定拓扑性质的子图结构. 直观上, 社区内节点间的连接比较紧密, 而不同社区节点间的连接比较稀疏^[1]. 社区发现算法已成为复杂网络分析中的重要基础性研究, 并已应用于商务智能、生物医学、行为分析等诸多领域^[2~4]. 尤其在社交网络分析方面, 通过对社交网站上用户的合理分群可以提高目标人群的定位准确性, 既可以友好地为用户推荐相似人群, 又能更快地锁定营销对象, 提高广告投放效率等^[5,6].

然而对社交网络信息过度地挖掘, 会暴露个人隐私, 包括个人的基本信息、所处的圈子, 以及相关的合作伙伴等. 针对此类问题, 国内外机构相继提出各种保护措施. 有算法层面的, 其中最经典的是

引用格式: 陶海成, 卜湛, 曹杰. 基于多目标强化学习的社区隐藏框架. 中国科学: 信息科学, 2021, 51: 1131–1145, doi: 10.1360/SSI-2020-0229
Tao H C, Bu Z, Cao J. A multi-objective reinforcement learning framework for community deception (in Chinese). Sci Sin Inform, 2021, 51: 1131–1145, doi: 10.1360/SSI-2020-0229

基于差分隐私的数学模型^[7], 该模型可以利用有限的网络数据发现特定人群, 比如恐怖组织, 但同时严格保证了正常人群的数据隐私; 也有一些政策层面的, 比如微博在 2017 年更新了个人信息保护政策¹⁾, 明确了个人基本信息的收集以及对 Cookie 的使用等; 欧盟于 2018 年通过了通用数据保护条例 (GDPR)²⁾, 明确个人数据必须匿名化进行存储, 默认使用最高的隐私设置等. 但是, 以上所提的保护措施主要针对的是用户的敏感数据, 包括用户电话信息、家庭地址信息, 以及一些聊天记录, 如何从公开数据层面保护用户隐私仍是一个很大的问题, 特别是社交网络数据, 由于可利用社区发现算法通过用户之间的相连关系推断用户隐私, 研究如何不被社区发现算法所追踪, 从而隐藏个人所在社区的社交关系, 是一个重要问题, 目前为止这方面的研究工作非常少. 另外, 在风险管理领域^[8], 低风险的用户通常与低风险的用户紧密相连, 高风险用户与高风险用户紧密相连. 不法分子 (高风险的用户) 会利用这种特性, 通过与低风险用户构建连接的方式将自己伪装成低风险用户, 从而获取大量的经济资源. 更有甚者, 恐怖组织通过改变网络结构的方式, 使得某成员既不会被执法机构通过社区发现算法所追踪, 成员之间又能互相取得联系^[9]. 因此, 研究社区隐藏算法具有两方面的意义. 一方面, 可以通过破坏社区发现算法的有效性起到保护个人隐私的作用, 另一方面, 由于社区隐藏算法可以作为一种模拟不法分子恶意操控网络结构进行非法活动的模型, 构建能对抗社区隐藏算法的社区发现算法能够使得社区发现算法更具健壮性.

已有的隐藏社区算法大多是在网络中已存在的点之间进行加边或者减边, 但在现实网络中, 比如基于 facebook 的社交网络, 期望用户改变与之相关的结构信息 (特别是指定哪些人应该关注, 哪些人应该取消关注) 可操作性比较小且代价很大. 为此, 本文通过在现有网络中增加点 (即伪造账户) 及其相关的边 (即伪造与该账户相关的社交关系), 来达到社区隐藏的目的; 其次, 该方式 (即加点和加边) 巧妙地将社区隐藏问题转换为网络的增长问题, 继而基于不同的网络增长模型, 可研究其在社区隐藏上的不同表现; 最后, 加点和加边的方式本质上可以描述为一个决策过程, 即根据当前网络的结构, 选择合适的加点和加边策略, 使得最终的指标最大, 而这正好契合强化学习的框架. 为此, 本文针对两种隐藏指标和不同的网络增长模型, 提出了基于多目标强化学习的社区隐藏算法.

2 相关工作

2.1 社区发现算法

已有的社区发现算法众多, 从优化方法的角度大致可分为两大类, 即全局优化和局部扩展方法. 全局优化方法通过预定义目标函数, 如社区结构评价函数, 将社区发现问题转化为基于给定目标函数的全局优化问题. 在此方面, 文献 [1] 中, Newman 提出了基于随机图模型的模块度 (modularity); 基于模块度, 很多学者提出了各种不同的优化算法来发现网络的社区结构. 此外, 基于不同的假设模型, 如 Potts 模型、HW 模型、RN 模型和标签传播模型等, 很多替代性目标函数也被相继提出. 但上述目标函数的全局优化方法都不同程度上存在分辨率限制问题, 即难以发现微观尺度的社区结构^[10]. 局部扩展方法多采用启发式策略挖掘网络中的局部社区, 再组装已挖掘的局部社区获得网络的全局社区结构. 这类方法的共性问题是如何定义局部社区, 除了经典图论中的 k -clique, k -plex 和 k -core 等局部结构可供参考外, 一些学者利用耦合节点集的统计特征来定义局部模块度指标, 并基于局部优化发现紧密耦合社区^[11]. 此外, 还有一些基于启发式算法的模型, 比如基于模拟退火、极值优化, 以及遗传算法的社区发现算法等^[12].

1) <https://www.weibo.com/signup/v5/privacy>.

2) <https://gdpr-info.eu/>.

表 1 与已有社区隐藏算法的区别

Table 1 A comparison of different methods for community deception

Methods	Updates	Techniques	Measures	Categories
Nagaraja ^[13]	Edge additions	Centrality measure	Miss-ratio (false negative rate)	G1
Waniek ^[14]	Edge additions or deletions	Modularity	Concealment	G1
Fionda ^[15]	Edge additions or deletions	Safeness & modularity	Deception score	G1
Chen ^[16]	Edge additions and deletions	Modularity	Modularity & NMI	G2
Liu ^[17]	Edge additions	Normalized residual entropy	Partition similarity & mutual information & query accuracy	G2
Our method	Node and edge additions	Multi-objective reinforcement learning	Negative ratio association & ratio cut	G2

2.2 社区隐藏算法

目前国内外针对社区隐藏的研究以改变网络结构为主,即改变网络中节点的连边,主要可以分为两大类:只改变特定社区内结构的隐藏算法(G1)及可在全局网络中改变结构的社区隐藏算法(G2).这两类算法都是通过优化各自提出的隐藏指标来达到社区隐藏的目的.针对G1类算法,Nagaraja等^[13]将网络分为两个社区,即目标社区以及主社区(除去目标社区里的点),其隐藏指标定义为失效率(miss-ratio),又称假负率(false negative rate).基于此,作者采取给目标社区和主社区加边的方法来达到提高社区发现算法失效率的目的.对于加边的策略,作者基于中心性指标(centrality measure),比如介数中心性、度中心性,以及特征向量中心性,对两个社区中的点按中心性指标降序排列,并顺序从中选取两个点进行加边从而更新网络结构.Waniek等^[14]提出另一种隐藏指标(concealment measure),该指标由两部分组成,第1部分针对前面提到的失效率问题,充分考虑将目标社区中的点均匀分布到各个社区,而不是只针对主社区;指标的第2部分要求分散到各个社区中的点越多越好.针对基于模块度的社区发现算法,作者提出了一种快速的启发式算法,即通过在目标社区内减边、不同社区间加边的方法降低模块度指标,从而达到社区隐藏的目的.Fionda等^[15]提出了更适合社区隐藏指标的3个准则,即在改变网络结构后,目标社区内的点尽量能够保持连通性,目标社区内的点应均匀分布到其他社区中以及分散到大社区中去.基于此,作者提出了一个欺骗分数指标(deception score),并通过优化一个安全函数来达到社区隐藏的目的,同时作者证明只有在目标社区内部减边、不同社区之间加边才能最优化该函数.针对G2类算法,Chen等^[16]提出改变网络结构的同时需要保持每个点的度不变,即对某个点增加一条边的同时也要对其减去相关的一条边,从而提出了3种启发式算法以及基于模块度指标的遗传算法.Liu等^[17]从信息熵的角度定义了原始网络结构的信息熵^[18]和基于社区划分的网络结构的信息熵.两种信息熵之差(即残余差)为社区结构给网络结构带来的信息量,并通过只加边的方式来最小化标准残余差从而达到社区隐藏的目的.

表1展示了本文提出的方法和已有社区隐藏算法的区别.本文所提出的方法主要有3方面的不同:第一,首次提出加点和加边的方式来进行社区隐藏,这种方式对用户的影响较小,更方便网站管理者进行操作;第二,通过强化学习的框架来学习加边策略,这也是强化学习在社区隐藏领域的首次应用;第三,不同于已有工作只是对单一目标进行优化,本文通过强化学习框架同时对两种目标进行优化,也可扩展至多种(> 2)目标.

3 基于加点和加边策略的社区隐藏

3.1 问题描述

现有的社区隐藏方法通常是基于网络结构中的点不变, 通过加边或者减边的方式来改变网络结构. 但在现实生活中, 通过已有点来达到加边或者减边的目的, 可操作性比较小, 尤其是针对社交网络, 寄希望于用户关注自己不喜欢的对象, 或者不再关注已有对象是不现实的, 故本文通过加点和加边的方式降低对用户的影响.

给定一个网络结构 $G = (V, E)$, 特定的社区发现算法 f , 比如 Louvain 算法, 这里用 $f(G)$ 表示基于网络结构 G 发现的社区结构, 社区隐藏的定义如定义 1.

定义1 (社区隐藏) 基于一个或者多个隐藏指标 O , 通过加点和加边的方式得到网络结构 $G' = (V', E')$, 其中 $V' = \{V \cup V^+\}$ 和 $E' = \{E \cup E^+\}$, V^+ 和 E^+ 分别表示增加的点集和边集, 使得

$$\begin{aligned} \max_{G'} \quad & O(f(G), f(G')) \\ \text{s.t.} \quad & |V^+| \leq d, \\ & |E^+| \leq b. \end{aligned} \quad (1)$$

式 (1) 中的隐藏指标 O 用于衡量社区结构在原始网络和改变结构后的网络之间的差异, 通过最大化这种差异来达到社区隐藏的目的, b 和 d 是预先设定的加边和加点的最大量. 社区隐藏算法都是基于优化隐藏指标来达到隐藏社区的目的, 但已有文献中的基于隐藏指标的算法都存在一定的问題. Nagaraja^[13] 提出的基于失效率的指标, 该指标的最大的问题是目标社区中的点有可能都被分配到同一个社区中, 即使失效率很高也达不到隐藏的目的. Waniek 等^[14] 所提出的算法寄希望降低模块度来优化所提出的隐藏指标, 但该算法通过随机加边和减边的方式并不一定能达到降低模块度的目的, 甚至在执行的过程中会出现模块度增加的情形. Fionda 等^[15] 提出的基于欺骗分数的算法, 即在目标社区内部减边、不同社区之间加边, 但前提是目标社区内部的点是连通的, 然而某些社区发现算法, 比如 Louvain 算法能检测出的社区内部有可能是不可连通的, 即由几个连通分量构成的^[19]. Chen 等^[16] 提出的基于模块度的算法, 通过遗传进化模型并不能完全达到模块度降低的目的.

为此, 本文从社区定义出发, 定义了两种可以直接优化的指标. 在一般网络结构中, 社区内部是紧密联系的, 而社区与社区之间的连接比较稀疏 (即社区定义), 假设有 p 个社区为 $\mathcal{P}_i = (V_i, E_i)$, A 表示邻接矩阵, $L(V_i, V_j) = \sum_{i \in V_i, j \in V_j} A_{ij}$, $\bar{V}_i$ 表示除社区 i 外的点, 则指标 ratio association (RA) 可写成式 (2), 表示的社区内部的边的密度, ratio cut (RC) 可写成式 (3), 表示的是社区之间的边的密度, 而社区隐藏的目的就是破坏这种结构, 即社区内部越来越稀疏, 社区之间越来越紧密, 所以, 基于 negative ratio association (NRA) 和 ratio cut, 即 $\text{NRA} = -\text{RA}$ 和 RC , 本文将隐藏指标定义为 $O_{\text{nra}} = \text{NRA}_{\mathcal{P}'} - \text{NRA}_{\mathcal{P}}$ 和 $O_{\text{rc}} = \text{RC}_{\mathcal{P}'} - \text{RC}_{\mathcal{P}}$, $\mathcal{P}'$ 表示网络结构改变后的社区结构.

$$\text{RA} = \sum_{i=1}^p \frac{L(V_i, V_i)}{|V_i|}, \quad (2)$$

$$\text{RC} = \sum_{i=1}^p \frac{L(V_i, \bar{V}_i)}{|V_i|}. \quad (3)$$

基于这两种指标, 社区隐藏算法的实现存在 3 个关键点: (1) 如何加点和加边, 如果使用蛮力法, 即测试每个点及其连边, 然后从中选择最优的方案, 时间复杂度极其高 ($O(|V|^b)$, b 为加入边的个数).

为此, 本文将从网络增长模型的角度进行连边的选择, 不仅可以有效降低模型的复杂度, 同时也可以探寻不种网络增长模型在社区隐藏上的表现. (2) 如何基于不同的网络增长模型, 使得社区隐藏性能更好, 即基于已改变的网络结构, 如何选取适当的生长模型, 使得隐藏指标最大化, 本文将基于强化学习框架给出一个决策模型. (3) 如何同时考虑多种隐藏指标, 使得社区隐藏性能在多种指标上达到最优, 本文将提出两种多目标学习算法.

3.2 基于网络增长模型的社区隐藏

传统的网络增长模型描述了新加入节点与原始网络的连边方式, 与本文所提出的加点和加边方式相符, 故可将社区隐藏问题的连边策略转换为网络增长问题. 本文选取了 4 种不同的网络增长模型, 研究其对社区隐藏的影响, 具体 4 种模型如下:

- **随机网络增长模型 (Rand 模型).** 根据文献 [20] 所述, 该模型由 Erdos 和 Renyi 最早提出, 网络中两个节点之间随机进行连边.

- **基于度的幂律分布的网络增长模型 (BA 模型).** 该网络由 Barabasi [21] 提出, 度数越高的节点被连边的概率越大.

- **基于度和适应度函数的网络增长模型 (BB 模型).** 基于度的幂律分布的模型, 适用于大部分的真实网络, 但在某些网络中, 比如万维网, 最初的时候, 两个网页的度是一样的, 但由于两个网页内容的不同, 越接近时下热点内容的网页的吸引力越大, 新进的网页越有可能与之相连. 基于此, 文献 [21] 中 Barabasi 提出了同时基于度和适应度函数的模型, 其中, 度表示连边的能力, 适应度函数表示该点的吸引力.

- **基于多因素适应度函数的网络增长模型 (GH 模型).** 在另一些网络中, 比如文章引用网络, 一篇文章的引用通常由多种因素决定, 包括作者的偏好、期刊的重要程度, 以及被引用文章的重要性等. 基于此, Ghadge 等 [22] 提出一种基于多种因素的适应度函数, 同时, 基于独立分布以及大数定理, Ghadge 等证明该适应度函数符合对数正态分布.

基于 4 种网络增长模型可以设计出 4 种加点和加边的策略 (见算法 1), 唯一的区别在于连边的方式, 如算法 1 中的第 4 行, 基于随机模型的社区隐藏算法是随机选取点相连; 基于 BA 模型的社区隐藏算法则是度越大的点, 连边的概率越大; 同理, 基于 BB 模型的社区隐藏算法根据度和服从均匀分布的适应度函数选择点连边; 基于 GH 模型的社区隐藏算法是根据服从对数正态分布的适应度函数选择点的连边.

3.3 基于多目标强化学习的社区隐藏

为了达到隐藏社区的目的, 即选取合适的连边策略来最大化定义 1 中的隐藏指标, 3.2 小节提出使用网络增长模型来构建连边策略, 然而这种策略会随着网络的演化而不断变化, 即每次连边策略都会基于改变的网络结构进行调整, 不同于一般的学习任务, 这种动态的决策过程, 需要通过不断地尝试而进行优化. 而这种与环境 (演化的网络) 不断交互的决策行为 (连边策略) 正好契合强化学习的框架.

强化学习 [23] 以马尔科夫 (Markov) 决策过程为数学基础, 由四元组表示, 即 $(\mathcal{A}, \mathcal{S}, \mathbf{r}, \mathcal{P})$, 其中 $\mathcal{S} = \{s_1, s_2, \dots, s_n\}$ 表示状态空间, $\mathcal{A} = \{a_1, a_2, \dots, a_m\}$ 表示动作空间, $r(s, a)$ 表示在状态 s 的情况下, 采取动作 a 所获得的奖赏值, $\mathcal{P}(s'|s, a)$ 表示在状态 s 下, 采取动作 a , 从而转移到状态 s' 的概率, 其目标是学习一个最优策略 π , 该策略是一个状态到动作映射函数, 使得累积奖赏值最大化. 通常

Algorithm 1 Community deception based on the network growth model

Input: The original network $G = \{V, E\}$, the number of edge additions b , the set of node additions V^+ .
Output: The rewired network $G' = \{V', E'\}$.

```

1: Initialize budget = 0,  $G' = G$ ;
2: for budget++ <  $b$  do
3:   while True do
4:     Choose  $V_a$  randomly from  $V^+$ , add links between  $V_a$  and  $V_i$  in  $G$  based on the four network growth models, i.e.,
        $e_{ia}$ ;
5:     if  $e_{ia}$  is not in  $G'$  then
6:        $V' = V' \cup \{V_a\}$ ;
7:        $E' = E' \cup \{e_{ia}\}$ ;
8:       break;
9:     end if
10:  end while
11: end for
12: Return  $G'$ .

```

使用的累积奖赏值如式 (4) 所示, 其中 $\gamma \in [0, 1]$ 为折扣因子, 表示时刻 t 的累积折扣奖赏值.

$$\mathbf{r}_t = \mathbf{r}_{t+1} + \gamma \mathbf{r}_{t+2} + \gamma^2 \mathbf{r}_{t+3} + \gamma^3 \mathbf{r}_{t+4} + \cdots = \sum_{k=0}^{\infty} \gamma^k \mathbf{r}_{t+k+1}. \quad (4)$$

为了得到最优策略, 可使用两种函数, 即状态价值函数 $V(\mathbf{s})$, 如式 (5), 表示在状态 $\mathbf{s}$ 下的期望奖赏值以及状态动作价值函数 $Q(\mathbf{s}, \mathbf{a})$, 如式 (6), 表示在状态 $\mathbf{s}$ 下采取动作 $\mathbf{a}$ 所获得的期望奖赏值.

$$V^\pi(\mathbf{s}) = \mathbb{E}_\pi[\mathbf{r}_t | \mathbf{s}_t = \mathbf{s}], \quad (5)$$

$$Q^\pi(\mathbf{s}, \mathbf{a}) = \mathbb{E}_\pi[\mathbf{r}_t | \mathbf{s}_t = \mathbf{s}, \mathbf{a}_t = \mathbf{a}]. \quad (6)$$

两种函数的贝尔曼 (Bellman) 方程^[24] 分别如式 (7) 和 (8) 所示, 该方程的意义在于可通过迭代计算的方法来计算累积折扣奖赏值. 基于此, 一种被广泛使用的通过更新 $Q(\mathbf{s}, \mathbf{a})$ 来最大化累积奖赏值的算法被提出, 即 Q -learning, 该方法通过构建一张〈状态, 动作〉表 (Q 表) 来存储 $Q(\mathbf{s}, \mathbf{a})$, 表格的更新根据式 (9), 其中 α 为学习速率. 本文的强化学习模型基于 Q -learning 算法.

$$V^\pi(\mathbf{s}) = \sum_{\mathbf{a}} \pi(\mathbf{s}, \mathbf{a}) \sum_{\mathbf{s}'} \mathcal{P}(\mathbf{s}' | \mathbf{s}, \mathbf{a}) [\mathbf{r}(\mathbf{s}' | \mathbf{s}, \mathbf{a}) + \gamma V^\pi(\mathbf{s}')], \quad (7)$$

$$Q^\pi(\mathbf{s}, \mathbf{a}) = \sum_{\mathbf{s}'} \mathcal{P}(\mathbf{s}' | \mathbf{s}, \mathbf{a}) \left[\mathbf{r}(\mathbf{s}' | \mathbf{s}, \mathbf{a}) + \gamma \sum_{\mathbf{a}'} Q^\pi(\mathbf{s}', \mathbf{a}') \right], \quad (8)$$

$$Q(\mathbf{s}, \mathbf{a}) = (1 - \alpha) Q(\mathbf{s}, \mathbf{a}) + \alpha \left(\mathbf{r} + \gamma \max_{\mathbf{a}'} Q(\mathbf{s}', \mathbf{a}') \right). \quad (9)$$

为此, 从强化学习的角度出发, 定义四元组 $(\mathcal{A}, \mathcal{S}, \mathbf{r}, \mathcal{P})$ 如下:

- **动作空间** $\mathbf{a} \in \mathcal{A}$. 选取随机、基于度的幂律分布、基于度和均匀分布以及基于对数正态分布的 4 种加边策略为动作空间.

- **状态空间** $\mathbf{s} \in \mathcal{S} = \{1, -1, 0\}^l$. 由最近 l 个指标值的一致性策略构成. 式 (10) 展示了两个指标值的一致性策略, 如若存在多于两个的指标则可采用投票的方式决定策略的一致性, 在加第 t 条边时,

状态表示为 $S^t = \{s_0^t, s_1^t, \dots, s_{k-1}^t\}$.

$$s_i^t = \begin{cases} 1, & O_{\text{nra}}^{t-i}, O_{\text{rc}}^{t-i} \text{ 同时增大,} \\ 0, & O_{\text{nra}}^{t-i}, O_{\text{rc}}^{t-i} \text{ 一个增大一个减小,} \\ -1, & O_{\text{nra}}^{t-i}, O_{\text{rc}}^{t-i} \text{ 同时减小.} \end{cases} \quad (10)$$

• **奖赏值 r .** 设置为社区隐藏的指标值, 如果有多个指标, 则是由多个指标值构成的向量, 这里使用两种指标, 即 O_{nra} 和 O_{rc} , 则每次加完点和边后, 所获得的奖赏值为 $(O_{\text{nra}}, O_{\text{rc}})$.

• **状态转移概率 $\mathcal{P}$.** 设置该概率为 1, 即为确定性的决策.

为了得到最优策略, 即根据网络结构选择适合的点和边加入使得隐藏指标最大, 但由于存在多个隐藏指标 (本文基于两个指标, 可扩展至多个), 不同的隐藏指标会产生不同的奖赏值, 从而对应不同的策略, 为了同时优化多种指标得到最优的一致性策略^[25], 本文针对社区隐藏问题, 基于强化学习中的 Q -learning 算法, 提供两种基于多种指标的算法模型.

• **标量化多目标 Q -learning 算法 (ScaledRein 算法).** 在 Q -learning 算法中, 动作的选取是在 Q 表上采取 ϵ -贪心算法, 如式 (11) 所示. 但现在由于存在多个目标, 每个目标会对应不同的奖赏值, 从而对应不同的 Q 表, 比如 m 个目标对应 m 个 Q 表, 即 $(Q_1, Q_2, \dots, Q_m)$, 每个动作在 m 个 Q 表中都展现出不同的值, 即 $(Q_1(s, a), Q_2(s, a), \dots, Q_m(s, a))$, 为此定义了一个标量化函数, 如式 (12), 即赋予每个目标对应的 Q 表不同的权重 w , 且权重之和为 1, 从而形成统一的 Q 值, 最后通过 ϵ -贪心算法获取最终的动作选择. 基于标量化多目标 Q -learning 的具体算法如算法 2 和 3.

$$a' = \begin{cases} \arg \max_a Q(s, a), & \text{概率 } \epsilon, \\ \text{随机选择一个动作}, & \text{概率 } 1 - \epsilon, \end{cases} \quad (11)$$

$$Q_w(s, a) = f(Q, w) = \sum_i^m w_i \times Q_i(s, a). \quad (12)$$

Algorithm 2 Action strategy

SelectAction(evalFunction, Q, s, ϵ).

```

1: Initialize an empty list evalList  $\leftarrow \{\}$ ;
2: Get a random number rand, rand  $\in [0, 1]$ ;
3: if rand  $\leq \epsilon$  then
4:   for each action  $a \in \mathcal{A}$  do
5:     Get the index value  $I_a = \text{evalFunction}(Q(s, a))$ ;
6:     Add  $(a, I_a)$  into evalList;
7:   end for
8:   Choose the action  $a^{\text{final}}$  with respect to the largest  $I_a$  in evalList;
9: else
10:  Choose  $a^{\text{final}}$  randomly in  $\mathcal{A}$ ;
11: end if
12: Return  $a^{\text{final}}$ .
```

• **基于 Pareto 最优的多目标 Q -learning 算法 (ParetoRein 算法).** 不同于标量化算法的将多目标问题以权重的形式转换为单目标问题, 基于 Pareto 最优的算法, 寄希望于寻求一种策略能够使得在多种指标上到达 Pareto 最优, 如定义 2 所示.

基于 van Moffaert 等^[26]的算法, 构建两个数据结构来替代 Q 表, 如式 (13) 所示, 其中 $\bar{R}(s, a)$ 表示基于 (s, a) 的平均奖赏值, $\text{ND}(s, a)$ 表示在状态 s 下采取动作 a 到达的下个状态的非支配策略集,

Algorithm 3 Community deception based on scalarized multi-objective Q -learning

Input: The number of edge additions b , the set of node additions V^+ , the number of iterations T , $\epsilon \in [0, 1]$.
Output: The rewired network.

```

1: Initialize  $Q(s, a) \leftarrow 0, \forall (s, a) \in \mathcal{S} \times \mathcal{A}$ ;
2: for episode = 1 to  $T$  do
3:    $s^1 \leftarrow \{-1, 0, 1\}^k$ ;
4:   for  $\tau = 1$  to  $b$  do
5:     Call the Algorithm 2 based on Eq. (12), i.e., SelectAction( $f(Q, w)$ ,  $Q$ ,  $s^\tau$ ,  $\epsilon$ ) and get the action  $a^\tau \in \mathcal{A}$ ;
6:     Update the network based on the action  $a^\tau \in \mathcal{A}$ , i.e., add links between two nodes in  $V^+$  and  $V$ , respectively;
7:     Get the next state  $s^{\tau+1}$  and reward  $r$  based on the action  $a^\tau \in \mathcal{A}$ ;
8:     Call the SelectAction( $Q_w$ ,  $Q$ ,  $s^{\tau+1}$ ,  $\epsilon$ ) in which  $\epsilon = 1$  and then get the action  $a^{\tau+1} \in \mathcal{A}$ ;
9:     for each objective  $o$  do
10:      Update  $Q(s, a)$ :
11:       $Q_o(s^\tau, a^\tau) \leftarrow Q_o(s^\tau, a^\tau) + \ell(r_o + \gamma Q_o(s^{\tau+1}, a^{\tau+1}) - Q_o(s^\tau, a^\tau))$ ;
12:    end for
13:     $s^\tau \leftarrow s^{\tau+1}$ ;
14:  end for
15: end for
16: Return the rewired network.
```

$\oplus$ 表示向量求和, γ 表示折损率. 基于 Pareto 最优的多目标 Q -learning 算法具体如算法 2 和 4. 其中, 策略选择算法是基于 Hypervolume 指标通过 ϵ -贪心算法来选择动作, Hypervolume 指标^[27] 定义为解集中所有点与参考点在空间中所围成的立方体的体积, 体积越大解集越优. 本文中计算 Hypervolume 指标的时间复杂度为 $O(\log n)$. 算法 4 中的 $n(s, a)$ 表示 (s, a) 被选中的次数.

$$Q(s, a) = \bar{R}(s, a) \oplus \gamma \text{ND}(s, a). \quad (13)$$

定义2 (Pareto 最优) 寻求一种策略 (或者称为解) 使得目标函数最大. 策略 π^* 支配策略 (比如 π) 当且仅当基于策略 π^* 的每个目标函数的值不会小于采取策略 π 的目标函数值, 并且至少存在一个较大的目标函数值. 若一个策略集中任何一个策略都不能被该集中其他策略支配, 则称该策略集为不可支配策略集. 所有可行解中的不可支配策略集即为 Pareto 最优策略集.

Algorithm 4 Community deception based on Pareto multi-objective Q -learning

Input: The number of edge additions b , the set of node additions V^+ , the number of iterations T , $\epsilon \in [0, 1]$.
Output: The rewired network.

```

1: Initialize  $Q(s, a) \leftarrow 0, \forall (s, a) \in \mathcal{S} \times \mathcal{A}$ ;
2: for episode = 1 to  $T$  do
3:    $s^1 \leftarrow \{-1, 0, 1\}^k$ ;
4:   for  $\tau = 1$  to  $b$  do
5:     Call the Algorithm 2 based on the Hypervolume, i.e., SelectAction(Hypervolume,  $Q$ ,  $s^\tau$ ,  $\epsilon$ ) and get the action  $a^\tau \in \mathcal{A}$ ;
6:     Update the network based on the action  $a^\tau \in \mathcal{A}$ , i.e., add links between two nodes in  $V^+$  and  $V$ , respectively;
7:     Get the next state  $s^{\tau+1}$  and reward  $r$  based on the action  $a^\tau \in \mathcal{A}$ ;
8:     Update the non-dominated set, i.e.,  $\text{ND}(s^\tau, a^\tau) = \text{ND}(\cup_{a'} Q(s^{\tau+1}, a'))$ ;
9:     Update  $\bar{R}(s^\tau, a^\tau) = \bar{R}(s^\tau, a^\tau) + \frac{r - \bar{R}(s^\tau, a^\tau)}{n(s^\tau, a^\tau)}$ ;
10:    Update  $Q(s^\tau, a^\tau) = \bar{R}(s^\tau, a^\tau) \oplus \gamma \text{ND}(s^\tau, a^\tau)$ ;
11:  end for
12: end for
13: Return the rewired network.
```

4 实验结果展示

本文将展示所提算法在 5 个数据集上, 分别针对 3 个经典社区发现算法的实验结果. 实验数据均

表 2 真实网络
Table 2 Real networks

Name	$ V $	$ E $	$\langle k \rangle$	$\langle C \rangle$
Karate	34	78	4.59	0.588
Dolphins	62	159	5.13	0.303
Football	115	616	10.66	0.403
Email	1133	5451	9.62	0.254
Power	4941	6594	2.67	0.107

是取 50 次的平均值.

4.1 社区发现算法

本实验主要针对 3 个经典的不同类别的社区发现算法:

Louvain 算法 (Louvain). 该算法通过迭代的方式最大化整个网络的模块度从而发现社区结构.

派系过滤算法 (CPM). 该算法通过扩展相似的派系达到发现重叠社区的目的.

标签传播算法 (LPA). 该算法是基于标签传播的局部社区划分方法.

4.2 数据集及评价指标

本文实验将基于 5 个真实的网络, 分别如下:

空手道俱乐部网络 (Karate). 第 1 个数据集是关于美国大学空手道俱乐部 34 个成员的数据. 点表示成员, 边表示成员之间的联系.

海豚网络 (Dolphins). 第 2 个数据集是处在新西兰的海豚形成的网络. 点表示宽吻海豚, 边表示海豚之间存在的频繁联系.

美国大学足球队网络 (Football). 第 3 个数据集是 2000 年美国大学足球队常规赛季的比赛数据, 共有 115 个足球队和 616 场比赛信息.

邮件网络 (Email). 第 4 个数据集是美国高校收集的关于 1133 个师生之间邮件交流的网络. 点表示师生, 边表示两者之间互相邮件联系过.

电力网络 (Power). 第 5 个数据集是美国西部地区的电力网络. 点表示发电机, 边表示它们之间的电源线.

相关数据的详细统计指标和在 3 类社区发现算法上社区个数如表 2, 其中 $\langle k \rangle = 2|E|/|V|$ 表示平均度, $\langle C \rangle$ 表示平均聚类系数.

评价指标. 将改变网络结构前的社区 P 作为标签, 改变网络结构后的社区 P' 作为对比量, 计算 ratio association 的减少比例, 即 $\text{NRAP} = (\text{RA}_{P'} - \text{RA}_P)/\text{RA}_P$ 以及 ratio cut 的增大比例, 即 $\text{RCP} = (\text{RC}_{P'} - \text{RC}_P)/\text{RC}_P$, NRAP 和 RCP 越大, 表明社区隐藏性能越好.

同时可使用社区发现算法中经常使用的标准化互信息 NMI (normalized mutual information) 和校正的兰德指数 ARI (adjusted rand index) 评价社区隐藏的性能, 两者越小, 表明性能越好.

4.3 结果比较

所提算法有两个重要参数, 即加入点的个数 (nodes) 以及边的个数 (budget), 为验证这两个参数对算法的影响, 固定其他参数, 即强化学习框架中的学习率 ℓ 设定为 0.9, 折损率 γ 同样设定为 0.9, 迭代次数 T 设定为 200, 基于标量化强化学习算法中的权重都设置为 0.5; GH 算法中的对数正态分布的

图 1 (网络版彩图) Karate 网络上改变增加边的个数对 3 个社区发现算法的实验结果

Figure 1 (Color online) Community deception performance for three community detection methods with different budgets on Karate. (a) Louvain; (b) CPM; (c) LPA

均值和标准差分别设定为 0 和 1; 社区发现算法中的参数都是默认设置, 其中基于 CPM 的算法的每个 Clique 中点的个数为 3.

首先, 固定加入点的个数为原始网络中点个数的 5%, 通过增加边的个数 (与原始网络中边数成比例, 即 5%, 10%, 15%, 20%, 25%) 测试不同社区隐藏算法对 3 个社区发现算法的影响. 图 1 展示了基于 NRAP 和 RCP 这两个指标, 在 Karate 网络上, 3 个社区发现算法在增加边个数不同情况下的实验结果. 如图 1 所示, 虽然针对 3 个社区发现算法的实验结果数据不一样, 但总体趋势是一样的, 即随着边数的增加, 6 种算法或者策略, 即随机算法 (Rand)、BA 算法 (BA)、BB 算法 (BB)、GH 算法 (GH)、标量化多目标 Q -learning 算法 (ScaledRein), 基于 Pareto 最优的多目标 Q -learning 算法 (ParetoRein), 在 NRAP 和 RCP 指标上都先增长后降低, 尤其是在 Karate 这个网络上, 都在边数增加到原始边数的 15% 的时候获得最佳性能. 其中, ParetoRein 总体上性能最佳, 其次是 ScaledRein, 但最差的并不是基于随机的增长算法, 而是基于对数正态分布的增长模型算法. 同时能看出基于 NRAP 这个指标的数值相比于 RCP 的数值偏低, 究其原因, 加边对原有社区内部结构的影响比社区之间结构的影响小, 尤其是基于模块度和 Clique 的算法.

然后, 固定边的个数, 设置为以上实验取得最佳性能的原始边数的 15%, 点的个数变化情况为原始网络点的 10%, 15%, 20%, 25%, 30%. 如图 2 所示, 当点数增加到一定程度 (Karate 网络的 6 个点), 基于 NRAP 和 RCP 的指标会急剧下降, 尤其是对于 CPM 算法, 究其原因, 点的增加使得网络更稀

图 2 (网络版彩图) Karate 网络上改变增加点的个数对 3 个社区发现算法的实验结果

Figure 2 (Color online) Community deception performance for three community detection methods with different number of nodes on Karate. (a) Louvain; (b) CPM; (c) LPA

疏, 从而导致 CPM 更快失效. 从总体结果看, ParetoRein 算法依然是性能最好的算法. 同时, 与加边的影响不同的是, 加点对原有社区内部结构比社区之间的结构影响大. 通过图 1 所展示的结果进一步研究发现, 固定点的个数和增加加边的个数, 即通过平均度的增大来改变网络结构, 并不一定能起到提高社区隐藏性能的目的; 同理, 从图 2 看出, 平均度的减少也不能完全降低社区隐藏性能.

以上实验结果可知, ParetoRein 算法在所有加点和加边算法中性能最佳, 为了验证该算法的收敛性, 固定加点和加边的个数分别为 15% 和 15%, 计算迭代过程中的 RA 和 RC 指标值. 如图 3 所示, 该算法在 Karate 网络上能很快达到收敛, 在迭代轮次达到 180 左右 RA 和 RC 指标皆趋于平稳, 其中, RC 指标在初始阶段上升速率极快, 表明该算法对破坏社区之间的关系比破坏社区内部的关系效率更高.

最后, 研究社区隐藏算法给改变网络结构前的社区结构带来的变化, 即比较改变网络结构前的社区和改变网络结构后的 NMI 和 ARI 值. 固定加点和加边的个数, 分别为以上实验获得最佳性能时的 15% 和 15%, 在 4 个真实网络上进行测试. 结果如表 3 所示, 粗体表示的是最好指标值. 总体来看, 这几种算法的优劣排序为 ParetoRein $\rightarrow$ ScaledRein $\rightarrow$ BB $\rightarrow$ BA $\rightarrow$ Rand $\rightarrow$ GH. 在 Football 和 Email 网络上所有算法的总体性能最好, 而这两个网络结构比较稠密, 平均度比较大, 故而基于增长模型的社区隐藏算法对密集型网络更适用.

图 3 (网络版彩图) Karate 网络上目标值的变化趋势

Figure 3 (Color online) Learning curves on Karate

表 3 其余 4 个数据集上的实验结果

Table 3 Results on other four datasets

Dolphins	NMI			ARI			Football	NMI			ARI		
	Louvain	CPM	LPA	Louvain	CPM	LPA		Louvain	CPM	LPA	Louvain	CPM	LPA
Rand	0.8065	0.8914	0.8580	0.7892	0.8809	0.8390	Rand	0.6775	0.7084	0.6957	0.6521	0.6981	0.6715
BA	0.7984	0.8740	0.8617	0.7721	0.8515	0.8503	BA	0.6429	0.7137	0.6054	0.6210	0.6719	0.6412
BB	0.7735	0.8452	0.8201	0.7528	0.8493	0.8219	BB	0.6361	0.7395	0.6824	0.6018	0.6623	0.6583
GH	0.8714	0.9062	0.8973	0.8601	0.8631	0.9132	GH	0.6819	0.7047	0.6920	0.6721	0.6914	0.7142
ScaledRein	0.8076	0.8327	0.8109	0.7814	0.7901	0.7592	ScaledRein	0.5823	0.6483	0.6482	0.5572	0.6193	0.6025
ParetoRein	0.7592	0.7626	0.7812	0.7592	0.7382	0.7721	ParetoRein	0.5015	0.6729	0.6286	0.5619	0.6018	0.5592
Email	NMI			ARI			Power	NMI			ARI		
	Louvain	CPM	LPA	Louvain	CPM	LPA		Louvain	CPM	LPA	Louvain	CPM	LPA
Rand	0.7623	0.8308	0.7846	0.7409	0.7720	0.7538	Rand	0.8721	0.9014	0.8915	0.8548	0.8629	0.8491
BA	0.6532	0.7042	0.6859	0.6372	0.6828	0.6913	BA	0.8123	0.8376	0.8326	0.7923	0.8281	0.8162
BB	0.6781	0.6942	0.6801	0.6193	0.6713	0.6745	BB	0.7682	0.8083	0.7824	0.7892	0.8634	0.7629
GH	0.7297	0.7629	0.7423	0.7010	0.7301	0.7842	GH	0.9231	0.9387	0.9201	0.8923	0.8371	0.9068
ScaledRein	0.6208	0.6514	0.6401	0.6090	0.6480	0.6583	ScaledRein	0.7786	0.7912	0.7890	0.7392	0.8015	0.7912
ParetoRein	0.5972	0.6508	0.6386	0.6021	0.6215	0.6192	ParetoRein	0.7084	0.7526	0.8012	0.7021	0.7619	0.7863

现有算法中没有采用加点和加边策略的社区隐藏算法, 见表 1, 故我们从 G1 和 G2 类型的算法中选取只加边策略的两种算法与 ParetoRein 算法进行比较, 即 Nagaraja^[13] 提出基于度的 HD (high degree) 算法和 Liu 等^[17] 提出的基于信息熵的 REM (residual entropy minimization) 算法. ParetoRein 算法、HD 算法和 REM 算法加边个数设置为原始网络边数的 15%, ParetoRein 算法的加边个数设置为原始网络点数的 15%, 图 4 展示了在 NMI 指标值上的结果. 总体来说, ParetoRein 算法性能略优于 REM 算法, HD 算法最差. 从网络结构特性来看, 基于网络增长的 ParetoRein 算法针对平均度和平均聚类系数大的网络, 即本文中的 Football 和 Email 网络, 能起到更好的社区隐藏效果, 更易破坏连接紧密且社区结构比较明显的网络.

同时, 为了更直观地展示本文所提算法的有效性, 基于 Louvain 算法对比了在 Karate 原始网络上的社区结构和本文算法改变原始网络后的社区结构. 如图 5 所示, 图 5(a) 是基于原始网络的社区结构, 图 5(b) 是基于改变后网络的社区结构, 不同颜色表示不同的社区, 图 5(b) 中的三角表示新加入的点 (6 个点及其相关的 12 条边). 从图 5 中可以看出, 改变结构不仅增加了社区数目, 而且破坏了原有

图 4 (网络版彩图) 在 4 个数据集上与其他方法的性能比较

Figure 4 (Color online) Performance comparison of HD, REM, and ParetoRein on four datasets

图 5 (网络版彩图) 社区隐藏算法改变 Karate 网络前后的社区结构对比

Figure 5 (Color online) Communities detected on the original and rewired Karate networks

社区中的关系, 例如点 10 不再和点 34 属于一个社区, 而是与原先不属于一个社区的点 3 构成了一个社区. 由此可知, 本文所提算法确实能起到社区隐藏的作用.

5 结束语

本文从可操作性角度出发, 将加点和加边作为更新网络结构的方式, 从而将社区隐藏问题转换为网络增长问题. 基于 4 种网络增长模型, 即基于随机、基于度的幂律分布、基于度和均匀分布函数, 以及基于对数正态分布的增长模型, 以两个指标的一致性性能向量为状态, 以 4 种增长模型为动作空间, 以指标值作为奖赏值, 提出两种多目标的强化学习框架. 在 5 个数据集以及 4 种指标下, 验证了不同算法的有效性. 其中, 基于 Pareto 优化和标量化的 Q -learning 算法性能最佳, 基于度的幂律分布以及基于度和均匀分布函数的增长模型其次. 在未来的工作中, 除了将探寻不同参数, 比如强化学习框架中的迭代次数、学习率, 以及折损率等参数, 以及不同数据集对社区隐藏算法的影响, 还将进一步研究结合网络拓扑信息和节点特征信息的社区隐藏算法.

参考文献

- 1 Newman M E J. Communities, modules and large-scale structure in networks. *Nat Phys*, 2012, 8: 25–31

- 2 Li C Y, Tang Y, Lin H, et al. Parallel overlapping community detection algorithm in complex networks based on label propagation. *Sci Sin Inform*, 2016, 46: 212–227 [李春英, 汤庸, 林海, 等. 基于标签传播的可并行复杂网络重叠社区发现算法. *中国科学: 信息科学*, 2016, 46: 212–227]
- 3 Qiao S J, Han N, Zhang K F, et al. Algorithm for detecting overlapping communities from complex network big data. *J Softw*, 2017, 28: 631–647 [乔少杰, 韩楠, 张凯峰, 等. 复杂网络大数据中重叠社区检测算法. *软件学报*, 2017, 28: 631–647]
- 4 Li D, Cheng M Q, Xu Y, et al. Optimal community detection method based on average mutual information. *Sci Sin Inform*, 2019, 49: 613–629 [李东, 程鸣权, 徐杨, 等. 基于平均互信息的最优社区发现方法. *中国科学: 信息科学*, 2019, 49: 613–629]
- 5 Barbieri N, Bonchi F, Manco G. Influence-based network-oblivious community detection. In: *Proceedings of IEEE International Conference on Data Mining*, Dallas, 2013. 955–960
- 6 Zhang K, Bhattacharyya S, Ram S. Large-scale network analysis for online social brand advertising. *Mis Quart*, 2016, 40: 849–868
- 7 Kearns M, Roth A, Wu Z S, et al. Private algorithms for the protected in social network search. *Proc Natl Acad Sci USA*, 2016, 113: 913–918
- 8 Akoglu L, Tong H H, Koutra D. Graph based anomaly detection and description: a survey. *Data Min Knowl Disc*, 2015, 29: 626–688
- 9 Johnson N F, Zheng M, Vorobyeva Y, et al. New online ecology of adversarial aggregates: ISIS and beyond. *Science*, 2016, 352: 1459–1463
- 10 Li H J, Bu Z, Li A, et al. Fast and accurate mining the community structure: integrating center locating and membership optimization. *IEEE Trans Knowl Data Eng*, 2016, 28: 2349–2362
- 11 Balasundaram B, Butenko S, Hicks I V. Clique relaxations in social network analysis: the maximum k-plex problem. *Oper Res*, 2011, 59: 133–142
- 12 Pizzuti C. Evolutionary computation for community detection in networks: a review. *IEEE Trans Evol Comput*, 2017, 22: 464–483
- 13 Nagaraja S. The impact of unlinkability on adversarial community detection: effects and countermeasures. In: *Proceedings of International Symposium on Privacy Enhancing Technologies*, Berlin, 2010. 253–272
- 14 Wanek M, Michalak T P, Wooldridge M J, et al. Hiding individuals and communities in a social network. *Nat Hum Behav*, 2018, 2: 139–147
- 15 Fionda V, Pirro G. Community deception or: how to stop fearing community detection algorithms. *IEEE Trans Knowl Data Eng*, 2017, 30: 660–673
- 16 Chen J Y, Chen L H, Chen Y X, et al. GA-based Q-attack on community detection. *IEEE Trans Comput Soc Syst*, 2019, 6: 491–503
- 17 Liu Y, Liu J, Zhang Z, et al. REM: from structural entropy to community structure deception. In: *Proceedings of Advances in Neural Information Processing Systems*, Vancouver, 2019. 12918–12928
- 18 Li A S, Pan Y C. Structural information and dynamical complexity of networks. *IEEE Trans Inform Theory*, 2016, 62: 3290–3339
- 19 Traag V A, Waltman L, van Eck N J. From Louvain to Leiden: guaranteeing well-connected communities. *Sci Rep*, 2019, 9: 1–12
- 20 Frieze A, Karonski M. *Introduction to Random Graphs*. Cambridge: Cambridge University Press, 2016
- 21 Barabasi A L. *Network Science*. Cambridge: Cambridge University Press, 2016
- 22 Ghadge S, Killingback T, Sundaram B, et al. A statistical construction of power-law networks. *Int J Parallel Emerg Distrib Syst*, 2010, 25: 223–235
- 23 Arulkumaran K, Deisenroth M P, Brundage M, et al. Deep reinforcement learning: a brief survey. *IEEE Signal Process Mag*, 2017, 34: 26–38
- 24 Kiumarsi B, Vamvoudakis K G, Modares H, et al. Optimal and autonomous control using reinforcement learning: a survey. *IEEE Trans Neural Netw Learn Syst*, 2018, 29: 2042–2062
- 25 Barrett L, Narayanan S. Learning all optimal policies with multiple criteria. In: *Proceedings of the 25th International Conference on Machine Learning*, Helsinki, 2008. 41–47
- 26 van Moffaert K, Nowé A. Multi-objective reinforcement learning using sets of pareto dominating policies. *J Mach*

Learn Res, 2014, 15: 3483–3512

- 27 Shang K, Ishibuchi H, He L J, et al. A survey on the hypervolume indicator in evolutionary multi-objective optimization. IEEE Trans Evol Comput, 2021, 25: 1–20

A multi-objective reinforcement learning framework for community deception

Haicheng TAO¹, Zhan BU¹ & Jie CAO^{2*}

1. Jiangsu Provincial Key Laboratory of E-Business, Nanjing University of Finance and Economics, Nanjing 210023, China;

2. School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China

* Corresponding author. E-mail: Jie.Cao@nufe.edu.cn

Abstract Community deception algorithms aim to disable the community detection algorithms and thus hide relationships among users in social networks. Previous studies on community deception focus on modifying existing graph structures, which is infeasible in most real-world applications. In contrast, it is more practical to inject nodes and edges into existing graphs. Firstly, We model community deception as network growth and introduce two measures based on ratio association and ratio cut. Then, based on the reinforcement learning framework, we define the action space as the different network growth models, state as the performance vector of the agent's latest l strategies and reward as the values of two measures. Finally, we adopt two strategies to optimize the two objectives, i.e., scalarized multi-objective Q -learning and pareto-optimized multi-objective Q -learning. Extensive experiments on real-world datasets demonstrate that the proposed approach can achieve better performance than latest community deception methods.

Keywords social network analysis, community detection, community deception, multi-objective optimization, reinforcement learning

Haicheng TAO was born in 1987. He received his Ph.D. degree in computer science from Nanjing University of Science and Technology, Nanjing, in 2020. Currently, he is a lecturer at Nanjing University of Finance and Economics. His research interests include data mining, machine learning, reinforcement learning, and optimization.

Zhan BU was born in 1987. He received his Ph.D. degree in computer science from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2014. Currently, he is a professor with Jiangsu Provincial Key Laboratory of E-Business, Nanjing University of Finance and Economics, Nanjing. His research interests include complex network, data mining, and game theory.

Jie CAO was born in 1969. He received his Ph.D. degree in manufacturing automation from Southeast University, Nanjing, China, in 2002. Currently, he is a visiting professor at the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing. Dr. Cao has been selected in the Program for New Century Excellent Talents in University and awarded with Young and Mid-Aged

Expert with Outstanding Contribution in Jiangsu province. His research interests include cloud computing, business intelligence, and data mining.