

基于 K-shell 的关键节点识别算法

王飞飞, 孙泽军, 赵 岩

(平顶山学院 信息工程学院, 河南 平顶山 467036)

摘 要:在复杂网络中,对各节点的影响力进行识别并找出其中的关键节点,对于研究病毒的传播与控制、电网故障控制与预警等具有重要作用.在对现有关键节点算法分析研究的基础上,提出了一种基于 K-shell 的关键节点识别算法(KNIK),该算法综合考虑节点的全局与局部信息,同时引入节点与邻居节点之间的关联度,计算节点在网络中的最终影响力.为了对算法性能进行验证,以 SIR 模型为参照进行仿真实验,实验结果表明,KNIK 能够有效地检测节点的影响力,识别网络中的关键节点.

关键词:复杂网络;K-shell 算法;影响力;关键节点

中图分类号:TP391

文献标识码:A

文章编号:1673-1670(2023)05-0048-07

0 引言

现实世界中的许多关系都可以采用复杂网络的形式表示,比如互联网、社交网络、物流网络、电力网络等^[1].复杂网络中的关键节点是指与其他节点相比,对网络的结构和功能影响更大的节点^[2-4].例如:在电力网络中,通过关键节点的识别可以对重要节点进行预测与控制^[5-6];在病毒传播网络中,可以有效地判断出传播过程中的关键点,进行隔离与阻断^[7];在社交网络中,可以加快良性信息的传播,阻断恶意信息的蔓延^[8].因此,学者对于复杂网络关键节点识别的研究逐渐深入,许多关键节点识别算法被提出,比较典型的有 CELF 算法^[9]、度折扣(Degree Discount)^[10]、K-shell 分解法^[11]、度中心性(Degree Centrality, DC)^[12]、接近中心性(Closeness Centrality, CC)^[13]、介数中心性(Betweenness Centrality, BC)^[14]、网页排名算法(PageRank, PR)^[15]等.但是这些方法存在的共性问题为简单的算法结果不准确,影响力识别准确的算法过于复杂.因此,笔者提出了一种基于 K-shell 的关键节点识别算法(KNIK),该算法在计算节点影响力时,综合考虑节点自身影响力与其邻居节点能够为其提供的影响力,节点自身影响力基于节点自身的 K_s 值(K_s 值根据 K-shell 算法获得)与度进

行计算,邻居节点提供的影响力基于邻居节点的 K_s 值与度计算,同时引入邻居节点与该节点之间的关联性,该方法可以有效地对节点重要性进行识别.

1 相关工作

1.1 K-shell 算法

K-shell 算法对复杂网络中节点进行的是粗粒度划分,通过该算法计算每一个节点的 K_s 值,将节点分为若干层.具体思想为:首先,确定网络中度数为 1 的节点,删掉其连接的边,在剩余的节点中继续查找是否存在度为 1 的节点,如有则删掉其连边,直至网络中不存在度为 1 的节点,则该部分节点位于网络的第 1 层,其 K_s 值均为 1;其次,确定网络中度数为 2 的节点,删掉其连边,在剩余的节点中继续查找是否存在度小于等于 2 的节点,如有则删掉其连边,直至网络中没有度小于等于 2 的节点,该部分节点位于网络的第 2 层,其 K_s 值为 2;最后,以此类推,循环执行上述过程直至确定每个节点的 K_s 值.

以图 1 为例进行说明,图 1 为无向无权网络图,包括 14 个节点,17 条边.

通过 K-shell 算法计算图 1 中各节点的 K_s 值,其中, K_s 值为 3 的节点有 1、6、10、12, K_s 值为 2 的节点有 2、4, K_s 值为 1 的节点有 3、5、7、8、9、11、13、14.



收稿日期:2023-02-28

基金项目:河南省科技攻关项目(232102210041,232102210147,232102210099);河南省高等学校重点科研项目(23A520051)

作者简介:王飞飞(1982—),女,河南省叶县人,理学硕士,平顶山学院信息工程学院讲师,主要从事复杂网络研究.

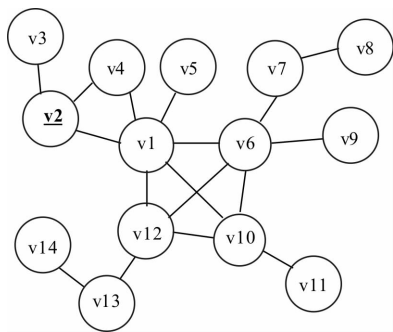


图1 网络示例

通过K-shell算法将图1中的节点分为3层, K_s 值相同的节点位于同一层,其重要性相同. 该算法过于粗粒化,是对网络中节点位置的一种粗略划分,节点之间的区分度不大,有一定的局限性.

1.2 其他经典算法与近几年有代表性的算法

关键节点识别的经典算法中,度中心性判断关键节点最为简单直观,但是该算法仅考虑节点的局部信息,导致判断结果不够精确;接近中心性通过所有节点对之间的相对距离确定其影响力,虽然识别比较准确,但复杂度较高;介数中心性根据经过节点的最短路径数来确定节点的重要性,由于要对所有节点的路径数进行计算,复杂度较高;PageRank算法基于网页的链接结构对网页进行排序,但是其每个节点的跳转概率相同,且其中的参数需要实验获得,不具有普适性. 与以上方法不同,CELF与度折扣算法基于贪婪算法进行改进,与传统贪婪算法相比,计算效率有所提高,但是准确度略微降低.

近几年来,在对已有算法研究的基础上,出现了许多算法,如ELKSS^[16]、GIN^[17]、RLGI^[18]、MLI^[19]等. 在ELKSS算法中,基于KLSS提出了一种扩展局部K-shell和中心性的观点,但是该算法只考虑了节点的 K_s 值,具有一定的局限性. 在GIN算法中,节点的重要性不但与自身有关,还与其连接节点的重要性相关,该算法较为简单,但是仅以自身与邻居节点的度、与邻居节点的距离作为影响因素,节点影响力识别的准确度不高. RLGI算法在形成前 n 个影响节点集的同时考虑其他节点的影响,该算法考虑了节点在网络中的位置,避免了重要节点聚集在某一范围内,但是该算法在计算时会降低强影响力节点邻近节点的影响力. MLI算法考虑到网络的拓扑信息,引入网络嵌入和机器学习,在识别动态传播下的关键节点方面效果较好,但是

复杂度较高.

无论哪种算法,都以准确地识别节点影响力为目的,笔者在考虑运行结果准确性与算法复杂性的前提下,结合节点的局部与全局信息,提出了一种新的算法KNIK,该算法在节点 K_s 值的基础上引入了节点的度,同时引入节点与邻居节点之间的相似关系、邻居节点的 K_s 值与度,邻居节点贡献的加入可以使节点的检测更加有效;与其他算法相比,该算法简单,时间复杂度低,适用于较大规模的网络.

2 KNIK 算法描述

2.1 算法设计

一个节点的影响力除了其自身,还会受到其邻居节点的影响,如果其邻居节点所处的位置越居于中心,影响力越强,则该节点的影响力也会越高,因此KNIK算法主要综合考虑节点自身的影响力与邻居节点对它的贡献. 首先,计算节点的度与 K_s 值,根据节点自身的度与所在的层数计算节点自身初始影响力;然后,通过邻居节点的度、邻居节点所在的层与邻居节点与该节点的关联度来计算邻居节点为其提供的初始影响力;最后,综合节点自身初始影响力与邻居节点初始影响力,计算节点在网络中的影响力.

给定网络 $G=(V,E)$ 为无向无权网络, V 为网络中的节点集合, E 为网络中的边的集合.

1) 计算节点的度与 K_s 值. 在网络 $G=(V,E)$ 中,节点 v_i 的度为 $D(v_i)$,求其中所有节点度的最大值 k_{\max} 与最小值 k_{\min} :

$$k_{\max} = \max(D(v_i)), \quad (1)$$

$$k_{\min} = \min(D(v_i)). \quad (2)$$

通过K-shell算法计算所有节点的 K_s 值,节点 v_i 的 K_s 值为 $K_s(v_i)$,求其中所有节点 K_s 值的最大值 $K_{s,\max}$ 与最小值 $K_{s,\min}$:

$$K_{s,\max} = \max(K_s(v_i)), \quad (3)$$

$$K_{s,\min} = \min(K_s(v_i)). \quad (4)$$

2) 计算节点自身与邻居节点的初始影响力. 节点 v_i 的自身初始影响力为 $S(v_i)$,根据节点 v_i 的度、网络中节点度的最大值、节点 v_i 的 K_s 值与网络中节点 K_s 值的最大值计算,由于节点的 K_s 值仅能粗粒化地反映节点的影响力,但是区分度不够,因此引入节点的度,两个因素共同作用可以一定程度

上区分同一层节点的影响力,同时由于节点的度与 K_s 值不在同一计数层面,对其进行处理. 计算公式如下:

$$S(v_i) = D(v_i)/k_{\max} + K_s(v_i)/K_{s,\max}. \quad (5)$$

节点 v_i 的邻居节点 v_j 的初始影响力为 $N(v_j)$, 在计算邻居节点初始影响力时,由于不同位置的邻居对节点 v_i 的影响不同,因此根据 $K_s(v_i)$ 与 $K_s(v_j)$ 的比较结果,将邻居节点分为 3 类, v_j 与 v_i 在同一层, v_j 的层数小于 v_i , v_j 的层数大于 v_i , 同时引入节点 v_i 与邻居节点 v_j 的关联度:

$$\text{simj}(v_i, v_j) = \left| \frac{\Omega(v_i) \cap \Omega(v_j)}{\Omega(v_i) \cup \Omega(v_j)} \right|. \quad (6)$$

式中, v_i 的邻节点集合为 $\Omega(v_i)$, v_j 的邻节点集合为 $\Omega(v_j)$.

当 $K_s(v_j)$ 小于 $K_s(v_i)$ 时, 弱化 v_j 对于节点 v_i 的影响力:

$$N(v_j) = \text{simj}(v_i, v_j) (D(v_j) / (k_{\max} + k_{\min})) + K_s(v_j) / (K_{s,\max} + K_{s,\min}). \quad (7)$$

当 $K_s(v_j)$ 与 $K_s(v_i)$ 相等时,

$$N(v_j) = \text{simj}(v_i, v_j) (D(v_j) / K_{s,\max}) + K_s(v_j) / K_{s,\max}. \quad (8)$$

当 $K_s(v_j)$ 大于 $K_s(v_i)$ 时, 增强 v_j 对于节点 v_i 的影响力:

$$N(v_j) = \text{simj}(v_i, v_j) (D(v_j) / (K_{s,\max} - k_{\min})) + K_s(v_j) / (K_{s,\max} - K_{s,\min}). \quad (9)$$

3) 计算节点 v_i 在网络中的影响力.

$$K(v_i) = S(v_i) + \sum_{v_j \in \Omega(v_i)} N(v_j) / n_s. \quad (10)$$

式中, n_s 为网络中的节点总数.

该算法的详细过程如下所示:

输入:

$$G = (V, E)$$

输出:

所有节点的影响力值排序

计算所有节点的度值, 度的最大值与最小值

计算所有节点的 K_s 值, 最大 K_s 值与最小 K_s 值

for 集合 V 中的每一个节点 v_i do

 采用式(5) 计算节点 v_i 的初始影响力 $S(v_i)$

 for 节点 v_i 的邻居节点集中的节点 v_j do

 采用式(6) 计算节点 v_i 与 v_j 的关联度

 if $K_s(v_j) < K_s(v_i)$ then

 采用式(7) 计算 $N(v_j)$

 endif

 if $K_s(v_j) = K_s(v_i)$ then

 采用式(8) 计算 $N(v_j)$

 endif

 if $K_s(v_j) > K_s(v_i)$ then

 采用式(9) 计算 $N(v_j)$

 endif

 endfor

 采用式(10) 计算节点 v_i 在网络中的影响力

endfor

返回 Rank(V);

2.2 实例分析

为了对 KNİK 的算法过程进行描述, 以图 1 中的节点 v_2 为例: $D(v_2) = 3, K_s(v_2) = 2$; v_2 有 3 个邻居节点, 分别为 $v_1, v_3, v_4, D(v_1) = 6, K_s(v_1) = 3, D(v_3) = 1, K_s(v_3) = 1, D(v_4) = 2, K_s(v_4) = 2$; 在该网络中, $k_{\max} = 6, k_{\min} = 1, K_{s,\max} = 3, K_{s,\min} = 1$.

根据式(5) 计算节点 v_2 自身的初始影响力 $S(v_2) = 1.167$; 根据式(7)、(8)、(9) 计算 v_2 的邻居节点为其提供的初始影响力: $N(v_3) = 0.321, N(v_4) = 0.917, N(v_1) = 1.95$; 根据式(10) 计算节点 v_2 在网络中的最终影响力, $K(v_2) = 1.394$. 根据该方法可以计算出网络中所有节点的影响力.

将 KNİK 算法与各比较算法、SIR 模型在 Python 中编程实现运行, 得到网络中每个节点的影响力值之后进行排序, 表 1 为排序后的结果. SIR 模型中节点的感染概率设置为 0.03.

表 1 各算法在图 1 网络中的排序结果

BC	CC	K-shell	PR	GIN	ELKSS	RLGI	KNİK	SIR
1	1	1	1	1	1	1	<u>1</u>	<u>1</u>
6	6	6	6	6	6	6	<u>6</u>	<u>6</u>
12	12	10	10	12	12	10	<u>12</u>	<u>12</u>
2	10	12	12	10	10	12	<u>10</u>	<u>10</u>
10	2	2	2	2	2	2	<u>2</u>	<u>2</u>
7	4	4	7	4	4	13	<u>4</u>	<u>4</u>
13	7	5	13	7	5	8	<u>7</u>	<u>7</u>
4	5	3	4	5	7	14	<u>13</u>	<u>13</u>
5	13	7	8	13	13	4	<u>5</u>	<u>5</u>
3	9	9	14	9	9	7	<u>9</u>	<u>9</u>
9	11	8	3	11	11	3	<u>11</u>	<u>8</u>
8	3	11	9	3	3	11	<u>3</u>	<u>14</u>
11	8	13	11	8	8	9	<u>8</u>	<u>11</u>
14	14	14	5	14	14	5	<u>14</u>	<u>3</u>

由表 1 可知, KNİK 算法的前 10 个节点与 SIR



模型一致,节点影响力识别较为准确。

3 实验评估

在本实验中,选用6个有代表性的真实网络,将KNIK与BC、CC、K-shell、PR、GIN、ELKSS、RLGI等7种算法进行比较分析,所有实验在台式计算机上运行,操作系统为Win10,CPU为i7-9700,内存为16GB。

3.1 数据集说明

6个网络为空手道网络、宽吻海豚网络、电子邮件网络、欧洲公路网络、美国航空网络与电网网络。

空手道网络(Karate network):该网络是美国一所大学的一个空手道俱乐部成员组成的网络。

宽吻海豚网络(Dolphins network):该网络是根据新西兰62只宽吻海豚的交往构造的社会网络。

电子邮件网络(Email network):西班牙一所大学的电子邮件通信网络,节点代表用户,节点之间的边表示用户之间存在电子邮件往来。

欧洲公路网络(Euroroad network):一个主要位于欧洲的道路网,节点表示城市,节点之间的边表示它们通过E-road连接。

美国航空网络(USAir network):该网络由美国联邦航空管理局国家飞行数据中心首选航线数据库构建。

电网网络(Powergrid network):该网络包含美国西部各州电网信息,节点是发电机、变电站或变压器,节点之间的边是用于供电的线路。

以上6个真实网络的相关特性统计信息如表2所示。

表2 6个真实网络的特性统计

DataSets	V	E	<k>	k_{max}	<C>
Karate	34	78	4.588	17	0.588
Dolphins	62	159	5.129	12	0.303
Email	1 133	5 451	9.622	71	0.254
Euroroad	1 174	1 417	2.414	5	0.02
USAir	1 226	2 410	3.931	0.073	34
Powergrid	4 941	6 594	2.669	19	0.107

在表2中,|V|为节点数,|E|为边数,<k>为平均度, k_{max} 为最大度,<C>为平均聚类系数。

3.2 评价指标

3.2.1 SIR传播模型

SIR传播模型^[20]是一个经典的关键节点识别的评价标准。过去十几年中,虽然已经提出了多种评价方法,但SIR模型仍然是最为广泛使用的节点影响力评估指标之一。因此,笔者采用SIR传播模型来对各种算法在不同网络上的运行结果排序进行比较分析。

在SIR传播模型中,根据节点状态不同将其分为3类:易感染节点S(Susceptible)、感染节点I(Infected)与恢复节点R(Recovered)。网络初始化时,随机选择若干节点作为感染者I,其他节点则为易感节点S,每一个时间步内,I以一定的感染概率 α 感染与其相连的S类节点使其成为I类节点,同时已感染的I类节点以一定的治愈概率 β 进行恢复。此过程执行多次直至网络中不存在感染节点I,然后按照最终的感染节点数量获得所有节点的感染影响力排序结果。

3.2.2 Kendall τ 系数

Kendall τ 系数^[21]用来衡量2个元素个数相同的排序列表之间的相似性。假设某算法的结果序列用 $X=(x_1, x_2, \dots, x_n)$ 表示,SIR模型的仿真结果序列用 $Y=(y_1, y_2, \dots, y_n)$ 表示。如果 X 、 Y 中的序列对 (x_i, y_i) ($i=1, 2, \dots, n$)同时满足 $x_i > x_j$ 且 $y_i > y_j$,或者 $x_i < x_j$ 且 $y_i < y_j$,则认为这两个序列是一致的;如果 $x_i > x_j$ 且 $y_i < y_j$,或者 $x_i < x_j$ 且 $y_i > y_j$,则认为这两个序列是不一致的;否则,认为既不是一致,也不是不一致。计算公式如下:

$$\tau(X, Y) = \frac{n_c - n_d}{0.5n(n-1)}. \quad (11)$$

式中, n 表示序列中的元素个数, n_c 与 n_d 表示两个序列中一致与不一致的数量。 τ 值越大,则相似性越大,算法生成的排序结果越准确。

3.3 实验性能分析

3.3.1 不同数据集上前10个重要节点

为了对算法的性能进行分析,笔者将感染概率设置为0.03,恢复概率设置为1,通过实验获得SIR在不同网络中的执行结果。将各种算法在6个网络上运行获得节点影响力值,并将结果按降序排列。由于空间有限,此处只显示Dolphins与USAir

网络的前 10 个节点.

表 3 是 Dolphins 网络前 10 个节点排序,KNIK 算法与 SIR 模型获取的节点序列中有 9 个节点相同,其中前 4 个节点完全一致.

表 3 Dolphins 网络前 10 个节点排序

BC	CC	K-shell	PR	GIN	ELKSS	RLGI	KNIK	SIR
37	37	1	15	38	15	52	15	15
2	41	11	18	15	38	15	46	46
41	38	15	52	46	46	58	38	38
38	21	16	58	41	34	46	34	34
8	15	41	38	34	41	18	52	21
18	2	43	46	21	51	38	30	18
21	29	48	34	37	30	30	21	30
55	8	2	30	51	52	34	58	14
52	34	18	14	19	21	2	18	58
58	9	20	2	30	17	14	41	52

表 4 是 USAir 网络的前 10 个节点排序,KNIK

算法与 SIR 模型的前 8 个节点相同.

表 4 USAir 网络前 10 个节点排序

BC	CC	K-shell	PR	GIN	ELKSS	RLGI	KNIK	SIR
68	68	1	312	68	52	312	52	68
52	52	2	68	52	68	523	68	52
213	148	3	52	148	47	187	113	312
312	135	4	212	135	113	68	89	113
135	711	6	113	44	44	89	312	44
136	424	10	187	116	34	604	187	187
212	136	14	523	124	116	868	47	89
660	617	16	89	110	46	218	44	47
523	20	18	604	46	102	52	109	34
221	116	20	218	113	89	113	135	46

3.3.2 不同感染概率下的肯德尔值

在该实验中,感染概率的取值范围为 $[0.01, 0.1]$,SIR 模型的模拟迭代次数设置为 1 000.

图 2 为 6 个网络中 8 种算法获取的节点序列与不同感染概率下 SIR 模型获取的节点序列根据式(11)计算得到的 Kendall τ 值.

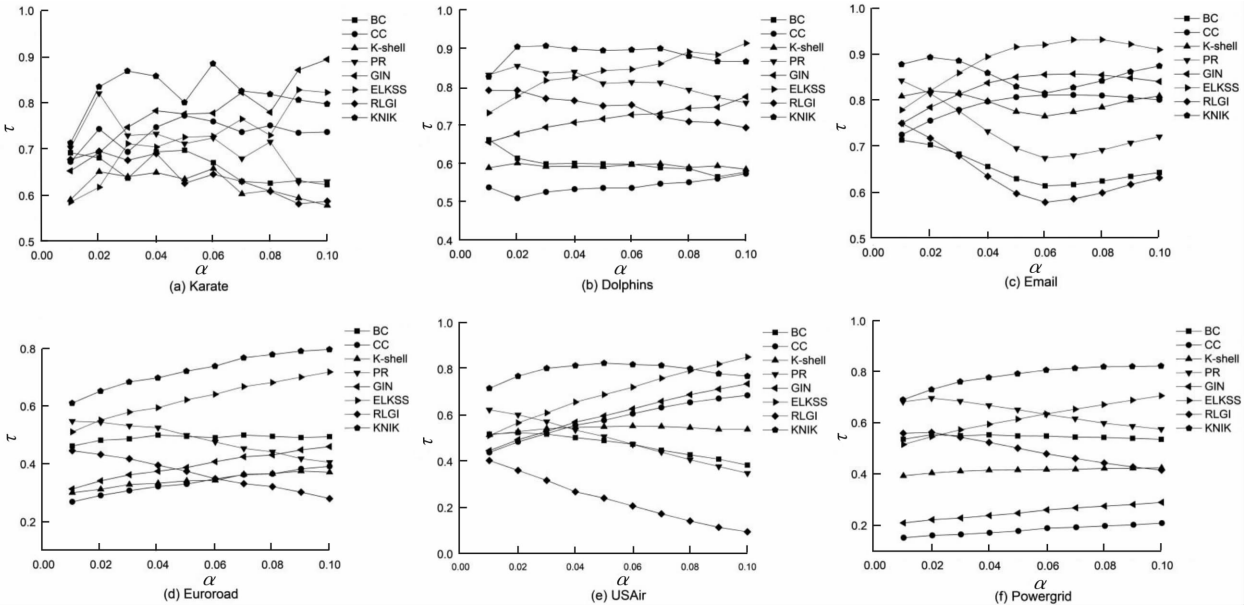


图 2 Kendall τ 值

由图 2 可知:KNIK 算法在 6 个网络中整体效果较好,其中 Euroroad 与 Powergrid 网络在所有感染概率下 Kendall τ 值均大于其他算法,效果最好;在 Karate 与 Email 网络中,在少数感染概率下,Kendall τ 值低于算法 GIN 与 ELKSS;在 Dolphins 与 USAir 网络中,在少数感染概率下,Kendall τ 值仅低于算法 ELKSS,但是在其他感染概率下,KNIK 的效果仍为最好.因为 KNIK 算法不仅考虑了节点在网络中的位置,同时引入了节点的度与邻居节点

能够为其提供的的影响力,提高了节点影响力值计算的准确性.

3.3.3 8 种算法在 SIR 模型上的传播能力值比较

笔者将 SIR 模型中的节点感染概率设置为 0.03,恢复概率设置为 1.将 8 种算法在 6 个网络上运行取得的结果序列与 SIR 获得的结果序列进行比较,获取各节点在 SIR 模型中的感染能力,节点的重要性与其感染能力成正比.如果算法效果较好,结果序列与 SIR 模型结果序列较为一致,则曲

线自左至右呈平滑下降趋势,否则呈不规则的折线.

在图 3 中,根据网络节点多少采用不同的显示形式,Karate、Dolphins 两个网络节点数较少,数据以线性形式显示,其余网络节点数较多,数据采用取常用对数的形式,该方式可以将影响力较大的节点进行侧重显示.由图 3 可知,在 Karate、Dolphins

与 Euroroad 网络中,KNIK 算法的曲线基本呈下降趋势,表明该算法效果优于其他算法;在 USAir 与 Powergrid 网络中,虽然 KNIK 算法曲线存在跳跃,但是整体效果仍优于其他算法;在 Email 网络中,KNIK 算法与 ELKSS 效果相似,但是优于其他算法.因此,在所有网络中,KNIK 算法整体效果较好.

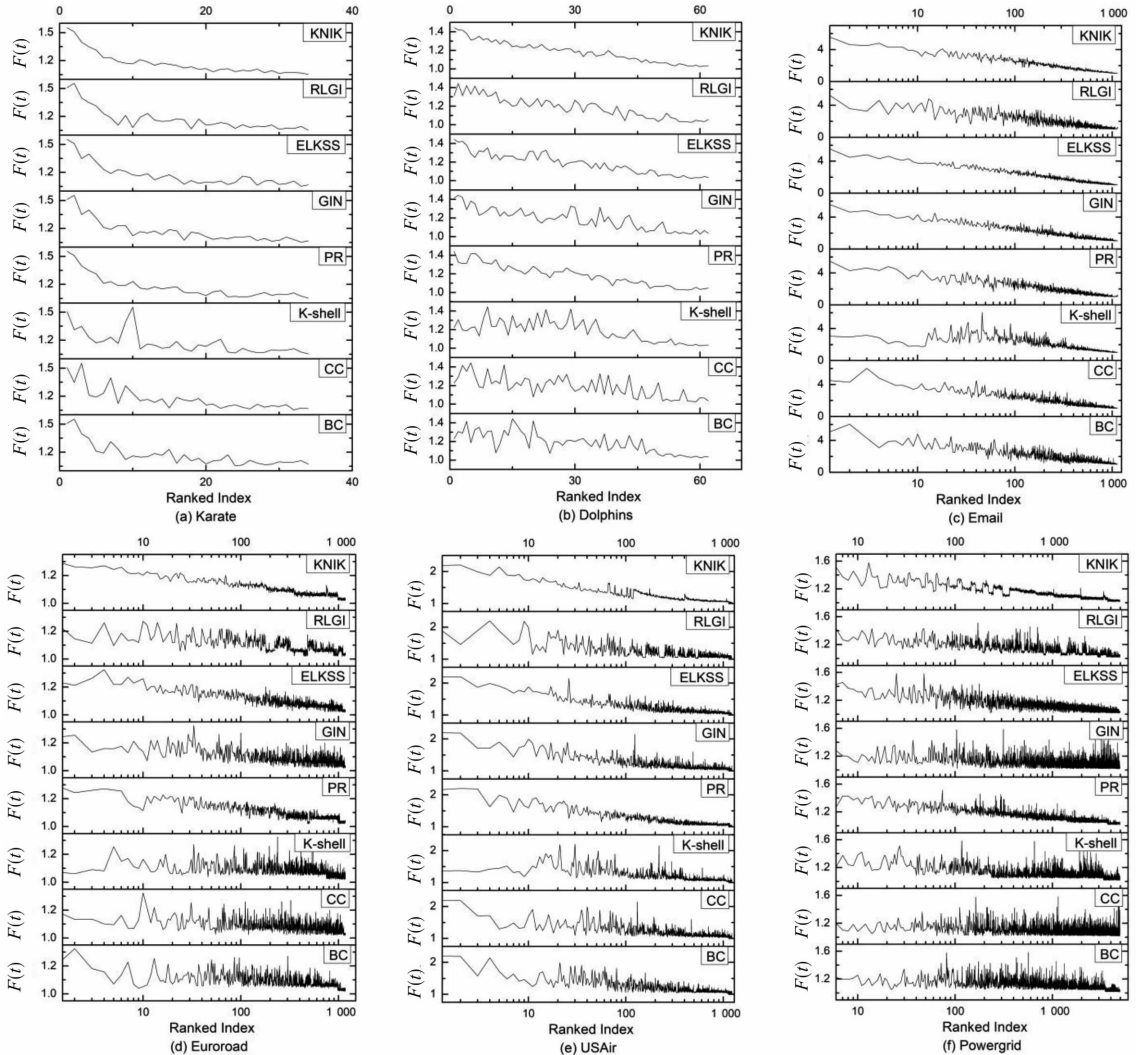


图 3 8 种算法在 6 个网络中的节点感染能力

4 结论

笔者在对现有关键节点识别算法进行研究的基础上,提出了一种基于 K-shell 的关键节点识别算法,该算法结合节点的局部与全局信息,根据节点度与 K_s 值、邻居节点的度与 K_s 值,同时引入节点与邻居节点之间的关联度计算节点的影响力,该算法解决了 K-shell 算法区分节点粗粒化的问题,提高了节点影响力识别的准确性.为了对算法效果进行验证,以 SIR 模型为参照,以 Kendall τ 值与节点在 SIR 模

型中的感染能力为评价标准,与其余 7 种算法在 6 个网络上进行实验.实验结果表明,该算法能够较为准确地识别网络中的关键节点,整体效果较好.

参考文献:

[1] 朱敬成,刘辉,王伦文,等. 基于网络拓扑重合度的关键节点识别方法[J]. 计算机应用研究,2021(12):3581 - 3585.



- [2] 苟玲, 肖玉芝, 李政辉. 两层复杂网络的特性及其抗毁性分析[J]. 计算机应用与软件, 2021(11): 114 - 120, 209.
- [3] 吴嫣媛, 刘向军. 考虑级联失效影响的复杂网络关键节点识别[J]. 计算机工程与设计, 2021(4): 920 - 926.
- [4] 王亭, 张永, 周明妮, 等. 融合网络拓扑结构特征与客流量的城市轨道交通关键节点识别研究[J]. 交通运输系统工程与信息, 2022(6): 201 - 211.
- [5] 邹艳丽, 谭林毅, 刘欣妍, 等. 基于子网划分的电网关键节点识别[J]. 计算物理, 2022(3): 361 - 370.
- [6] 陈世明, 王聪. 考虑电力潮流特性的信息物理系统关键节点辨识[J]. 计算机应用与软件, 2021(12): 50 - 55.
- [7] YANG J, YAO C, MA W, et al. A study of the spreading scheme for viral marketing based on a complex network model[J]. Physica A: Statistical Mechanics and Its Applications, 2010, 389: 859 - 870.
- [8] 刘彩虹, 刘强, 祁瑞华. 一种邻近 H-index 的社会网络关键节点识别技术[J]. 中国科技论文, 2019(3): 296 - 302.
- [9] LESKOVEC J, KRAUSE A, GUESTRIN C, et al. Cost-effective outbreak detection in networks[C]. San Jose: the 13th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2007: 420 - 429.
- [10] CHEN W, WANG Y, YANG S. Efficient influence maximization in social networks[C]. Paris: the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2009: 199 - 208.
- [11] KITSACK M, GALLOS L K, HAVLIN S, et al. Identification of influential spreaders in complex networks[J]. Nature Physics, 2010, 6: 888 - 893.
- [12] BONACICH P. Factoring and weighting approaches to status scores and clique identification[J]. Journal of Mathematical Sociology, 1972, 2: 113 - 120.
- [13] FREEMAN L C. Centrality in social networks conceptual clarification[J]. Social Networks, 1979, 1: 215 - 239.
- [14] FREEMAN L C. A set of measures of centrality based on betweenness[J]. Sociometry, 1977, 40: 35 - 41.
- [15] BRIN S, PAGE L. The anatomy of a large-scale hyper-textual web search engine[J]. Computer Networks and IS-DN Systems, 1998, 30: 107 - 117.
- [16] YANG F, ZHANG R, YANG Z, et al. Identifying the most influential spreaders in complex networks by an extended local K-shell sum[J]. International Journal of Modern Physics C, 2017, 28: 1750014.
- [17] ZHAO J, WANG Y, DENG Y. Identifying influential nodes in complex networks from global perspective[J]. Chaos, Solitons & Fractals, 2020, 133: 109637.
- [18] GUPTA M, MISHRA R. Spreading the information in complex networks: identifying a set of top-N influential nodes using network structure[J]. Decision Support Systems, 2021, 149: 113608.
- [19] YU E Y, FU Y, CHEN X, et al. Identifying critical nodes in temporal networks by network embedding[J]. Scientific Reports, 2020, 10: 12494.
- [20] TULJAPURKAR S. Infectious diseases of humans: dynamics and control[J]. Science, 1991, 254: 591 - 593.
- [21] KENDALL M G. A new measure of rank correlation[J]. Biometrika, 1938, 30: 81 - 93.

(责任编辑: 王彦江)

Key Node Identification Algorithm Based on K-shell

WANG Feifei, SUN Zejun, ZHAO Yan

(School of Information Engineering, Pingdingshan University, Pingdingshan, Henan 467036, China)

Abstract: In complex network, identifying the influence of each node and finding out the key nodes are of great importance to the study of the spread and control of viruses and the control and early warning of power grid fault. On the basis of the analysis and research of existing key node algorithms, a key node identification algorithm KNIK based on K-shell is proposed. This algorithm comprehensively considers the global and local information of nodes, and introduces the correlation degree between nodes and neighbors to calculate the final influence of nodes in the network. In order to verify the performance of the algorithm, the SIR model is used as a reference to carry out simulation experiments. The experiments show that KNIK can effectively detect the influence of nodes and identify key points in the network.

Key words: complex network; K-shell algorithm; influence; key points

