中国科学 : 物理学 力学 天文学 2024 年 第 54 卷 第 3 期 : 230511
SCIENTIA SINICA Physica, Mechanica & Astronomica physcn.scichina.com
《中国科学》杂志社 SCIENCE CHINA PRESS
论 文

基于时序网络节点嵌入的影响力最大化算法

詹秀秀[1,2], 谢晓雯[1], 张恺悦[1], 刘闯[1], 张子柯[2,3*]

1. 杭州师范大学复杂科学研究中心, 杭州 311121;
2. 浙江大学传媒与国际文化学院, 杭州 310058;
3. 浙江大学数字沟通研究中心, 杭州 310058
*联系人, E-mail: zkz@zju.edu.cn

收稿日期: 2023-04-08; 接受日期: 2023-06-06; 网络出版日期: 2024-02-06
国家自然科学基金(编号: 72371224)、浙江省自然科学基金(编号: LQ22F030008)、中央高校基本科研业务项目专项资金和杭州师范大学科研启动经费(编号: 2021QDL030)资助项目

摘要 随着互联网的高速发展, 影响力最大化算法在病毒营销、谣言遏制等众多领域得到广泛应用, 它旨在搜索一个指定规模的种子节点集, 以实现影响力传播范围最大化. 尽管已有许多影响力最大化算法被提出, 复杂系统的时序特征却鲜有被纳入考量. 为了解决时序网络中的影响力最大化问题, 本文提出了一种基于时序网络节点嵌入的影响力最大化算法(Micro- and Macro-based Clustering, MMC). 该算法能够通过节点嵌入有效地学习时序网络中的微观和宏观维度上的动力学特征. 基于嵌入向量, 对时序网络进行簇聚类, 以降低影响力最大化算法的复杂度, 减小种子节点集间的影响力范围重叠, 从而进一步提升算法的效果. 为了验证MMC的有效性, 本文在真实时序网络中进行了影响力传播实验与相似性比较. 结果显示, 相较于其他5种对比算法, MMC筛选出的种子节点在保证较好的传播效果的同时, 能够跳出既有算法的框架. 参数敏感性的分析表明, 节点聚类过程有效地提升了算法效果.

关键词 影响力最大化, 时序网络, 节点嵌入, 簇聚类, 传播模型

PACS: 89.75.-k, 89.75.Da, 89.75.Hc, 89.75.Fb, 89.75.Kd

1 引言

影响力最大化旨在从给定的社交网络中找到固定个数的种子用户, 经由特定的网络传播方式, 使得最终活跃节点的数目达到最大[1]. 该问题最早由Domingos等人[2]提出, 用于为病毒营销策略提供依据和建议, 它关注如何选择一群有影响力的人作为初始用户, 使得公司的产品在用户群体中得到最大范围的推广. 近年来, 随着互联网和社交媒体的高速发展, 影响力最大化算法在病毒营销和政府舆论控制等经济与管理领域的作用日益重要, 引起了广泛关注. 对影响力最大化问题的研究主要基于静态网络, 即网络中的节点和边的拓扑结构不随时间的发展而变化[3]. 基于静态网络的影响力最大化算法主要分为三种: 贪婪算法、启发式算法和社区划分算法. Kempe等人[1]将贪婪算法应用于解决影响力最大化问题, 该算

引用格式: 詹秀秀, 谢晓雯, 张恺悦, 等. 基于时序网络节点嵌入的影响力最大化算法. 中国科学: 物理学 力学 天文学, 2024, 54: 230511
Zhan X-X, Xie X W, Zhang K Y, et al. A node-embedding-based influence maximization algorithm in temporal network (in Chinese). Sci Sin-Phys Mech Astron, 2024, 54: 230511, doi: 10.1360/SSPMA-2023-0134

© 2024 《中国科学》杂志社

法始于一个空的种子节点集, 迭代地向种子节点集中添加具有最大边际收益的节点. 贪婪算法易于理解, 且具有$1-1/(e-\epsilon)$的近似保证, 但时间复杂度较高, 难以被应用于规模较大的网络. 针对贪婪算法的低效性, Leskovec等人[4]利用传播影响力函数的子模性提出了一种优化的贪婪算法——CELF算法. 除该算法外, Chen等人[5]提出了MixedGreedy算法, 该算法在每次的仿真实验中计算所有节点集的影响力, 然后利用CELF算法以优化贪婪算法的执行效率. 上述两个算法均大幅提升了贪婪算法的执行效率, 但在大型网络中执行上述算法仍需要消耗较长时间. 为了进一步提升影响力最大化算法的执行效率, 又有一系列的启发式方法被提出, Chen等人[5]提出了度消减算法, 该算法的原理是对种子节点的邻居进行度消减, 以降低种子节点之间影响力的重叠效应. 此外, Morone等人[6]提出了一种通过考虑远距离节点的度来选择种子节点的方法Collective Index (CI). 除这些启发式算法外, 还有一些学者提出基于网络节点中心性的高效率启发式算法[7–10]. 然而, 启发式方法对网络结构依赖程度较高, 在结构不同的网络上, 其算法表现相差较大; 并且在大部分网络中, 基于中心性的影响力最大化算法的有效性都大幅低于贪婪算法. 为了提升算法的效率, 同时降低种子节点集之间的影响力重叠范围, Wang等人[11]提出了基于社团检测的CGA算法来解决影响力最大化问题. 该算法分为两个阶段: 首先, 依据节点之间的紧密程度将网络划分为不同的社团; 随后, 在社团内贪婪地筛选种子节点并组成种子节点集. Chen等人[12]提出了一种基于簇聚类的影响力最大化算法CIM, 该算法在将网络划分为若干个簇后, 在簇中筛选候选种子节点集, 最后在候选节点中确定种子节点集. Huang等人[13]使用分治策略(Divide-and-Conquer Strategy)来进行社团划分, 并寻找种子节点集, 以实现病毒营销的影响力最大化. 然而, 在现实生活中, 许多复杂系统无法被简单地抽象为静态网络, 例如, 在社交网络中, 节点间只在某些特定时间存在连边[14]; 在疾病传播网络中, 疾病的传播存在先后顺序[15]. 尽管影响力最大化算法已被广泛应用于推荐系统[16,17]、信息扩散[18,19]等场景, 但是, 这些算法大多基于静态网络提出, 对时序信息的忽略或将影响算法的现实应用效果. 不少研究也指出, 网络结构随时间发生的微小改变, 可能对影响力最大化算法的效果带来极大的影响[20–22]. 为了进一步刻画复杂网络的时序特征, Holme等人[23,24]提出了时序网络的概念. 一个时序网络(图1(a))可以表示为$G=(V,E,\Gamma)$, 其中V和E分别为节点集和边集, $\Gamma=\{1,\ldots,T\}$为时间序列. 每条时序边$o=(u_i,u_j,t)\in E$代表节点u_i和u_j在时间步t产生连边. 而静态网络$G_s=(V,E_s)$则可以通过将时序网络压缩为单层网络来获得, 即$E_s=\{(u_i,u_j)|(u_i,u_j,t)\in E\}$. 静态网络虽然简单明了, 却也不可避免地造成了信息的丢失. 不同于静态网络, 时序网络记录了每条连边出现的时间, 从而能够反映网络的时序结构. 当网络结构随时间演化时, 从常规拓扑结构出发的信息处理方法很难捕捉到时序信息. 而在机器学习方法中, 基于节点嵌入的特征表示学习是一种学习图形结构的有效方法[25,26]. 通过节点嵌入技术, 得以将复杂的网络结构映射到低维嵌入向量中, 这些向量可用于许多下游任务的研究, 例如链路预测[27]、社区检测[28]等. 嵌入向量不仅保留了原始网络中的拓扑信息, 而且可以捕捉时序关系的演化, 反映节点之间的复杂关系, 为时序网络的应用提供了有力的支持[29]. 本文提出了一种基于时序网络节点嵌入的影响力最大化算法(Micro- and Macro-Based Clustering, MMC), 具体流程如图1所示. 该算法主要分为节点嵌入(图1(b))、节点簇聚类(图1(c))和筛选种子节点集(图1(d))三个阶段. 基于蒙特卡罗仿真的传播结果和算法相似性分析表明, 本文提出的方法在各个网络中的表现都较为稳定, 相较于对比算法, MMC筛选出的种子节点集不仅具有更高的传播影响力, 并且与对比算法间的相似性较低. 参数敏感性的分析表明, 节点聚类过程能有效提升传播范围.

2 算法

本文提出了一种基于时序网络节点嵌入的影响力最大化算法, MMC, 用于解决时序网络中的影响力最大化问题. MMC算法具体包括三个部分: 第一部分, 该算法对时序网络进行节点嵌入, 将网络中节点用向量表示, 以捕捉网络中微观和宏观上的动力学特征; 第二部分, 该算法通过节点嵌入向量计算节点间的相似性, 根据相似性将节点聚类为不同的簇; 第三部分, 该算法依据节点中心性在不同簇中选择种子节点, 构成最终的种子节点集.

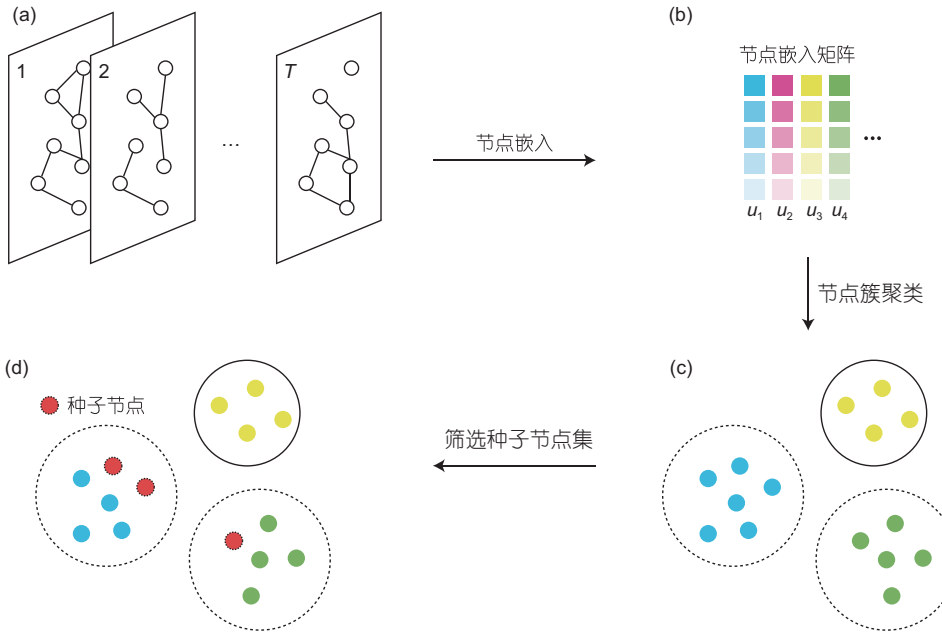

图 1 (网络版彩图)本文提出的基于时序网络节点嵌入的影响力最大化算法流程. (a) 时序网络示意图; (b) 将时序网络中的节点映射为低维嵌入向量; (c) 通过聚类算法将相似度高的节点划分至同一簇内; (d) 在簇内选择高影响力节点以最终组成种子节点集

Figure 1 (Color online) The flowchart of the proposed influence maximization algorithm based on temporal network node embedding. (a) Schematic diagram of a temporal network; (b) mapping of nodes in the temporal network to low-dimensional embedding vectors; (c) segmentation of nodes with high similarity using a clustering algorithm; (d) selection of high-influence nodes within vital cluster to form the final set of seed nodes.

2.1 节点嵌入

给定时序网络$G=(V,E,\Gamma)$, 其时序边表示为$o=(u_i,u_j,t)\in E$, 在时序网络上进行节点嵌入的目标是要学习一个映射函数$f: V\to R^d$, 其中d为嵌入维度, 且$d\ll|V|$, 以表征时序网络的拓扑特征和节点之间的相似性. 本文采用了Lu等人[30]提出的一种基于时序网络上微观和宏观动力学的节点嵌入方法, MMDNE. 微观动力学描述网络结构的形成过程, 定义为$I=\{(u_i,u_j,t)_m\}_{m=1}^{|E|}$, 代表节点$u_i$和$u_j$在$t$时刻建立连边这一时序事件; 宏观动力学描述网络规模的演化过程, 定义为$\mathcal{A}=e(t)_{t=1}^{T}$, 其中$e(t)$表示时序网络在$t$时刻的累计边数. 经过函数$f$映射得到的节点嵌入矩阵表示为$U=[\boldsymbol{u}_i]^{\mathrm{T}}$.

2.1.1 基于网络微观动力学的节点嵌入过程

时序网络的微观动力学指网络结构的形成过程, 具体表现为时序边的建立与消失. 为了刻画时序网络上的微观动力学, MMDNE基于节点之间的相似性和节点间影响力随时间衰退的特性, 将时序边$o=(u_i,u_j,t)$出现的强度定义为$\tilde{\lambda}_{i,j}(t)$, 具体公式如下:

$$\begin{aligned}\tilde{\lambda}_{i,j}(t)=&g(\boldsymbol{u}_i,\boldsymbol{u}_j)+\beta_{ij}\sum_{p\in H^i(t)}\alpha_{pi}(t)g(\boldsymbol{u}_p,\boldsymbol{u}_j)\kappa(t-t_p)\\&+(1-\beta_{ij})\sum_{q\in H^j(t)}\alpha_{qj}(t)g(\boldsymbol{u}_q,\boldsymbol{u}_i)\kappa(t-t_p),\end{aligned}\tag{1}$$

其中, $g\left(\boldsymbol{u}_i,\boldsymbol{u}_j\right)=-\|\boldsymbol{u}_i-\boldsymbol{u}_j\|_2^2$, 是度量两个节点间相似性的函数; $H^i(t)$和$H^j(t)$分别表示节点u_i和u_j在t时刻之前的历史邻居集; $\kappa(t-t_p)=\exp\left(-\delta_i\left(t-t_p\right)\right)$, 是一个随时间衰减的函数, 与$t$越相近的时间$t_p$被赋予的权重将越大; α和β为两个由层次时序注意力机制定义的注意力系数. 基于以上定义, 进一步将节点u_i和u_j在t时刻建立一条连边的概率表示为

$$p\left(u_i,u_j|H^i(t),H^j(t)\right)=\frac{\lambda_{i,j(t)}}{\sum_{i'}\lambda_{i',j}(t)+\sum_{j'}\lambda_{i,j'}(t)},\tag{2}$$

其中, $i'\in H^j(t)$, $j'\in H^i(t)$. 节点嵌入在微观层面的损失函数定义为

$$L_{\mathrm{mi}}=-\sum_{t\in\Gamma}\sum_{(u_i,u_j,t)\in E}\log p\left(u_i,u_j|H^i(t),H^j(t)\right).\tag{3}$$

2.1.2 基于网络宏观动力学的节点嵌入过程

宏观动力学指网络规模的演化, 可以被网络的累计连边数所描述. 为了刻画时序网络上网络规模的演化, 该方法定义了时序网络在t时刻新产生的连边数$\Delta e'(t)$:

$$\Delta e'(t) = n(t)r(t)\zeta\left(n(t)-1\right)^{\gamma}, \tag{4}$$

其中, $n(t)$表示t时刻的累计节点数; ζ为线性稀疏系数, γ为幂律稀疏指数; $r(t)$表示任意节点对之间在t时刻产生连边的概率. 考虑到连边概率应同时受到时序信息和网络结构两个因素的影响, MMDNE基于时间参数和节点的嵌入向量相似性来定义连接率$r(t)$:

$$r(t) = \frac{|E|^{-1}\sum_{(u_i,u_j,t)\in E}\sigma\left(-\|\boldsymbol{u}_i-\boldsymbol{u}_j\|_2^2\right)}{t^{\theta}}, \tag{5}$$

其中, t^{θ}是一个时间衰减函数, $\sigma(\cdot)$为sigmoid函数. 至此, 可以得到节点嵌入在宏观层面的损失函数:

$$L_{\mathrm{ma}} = \sum_{t\in\Gamma}\left(\Delta e(t)-\Delta e'(t)\right)^2. \tag{6}$$

2.1.3 基于微观和宏观动力学的节点嵌入过程

由于微观与宏观动力学相互驱动时序网络的演化, 进而交替影响节点嵌入学习, 所以该方法将微观动力学和宏观动力学相结合, 构造成一个统一的模型, 最终嵌入的损失函数表示为

$$L = L_{\mathrm{mi}} + \epsilon L_{\mathrm{ma}}, \tag{7}$$

其中, ϵ为权重系数, 且$\epsilon\in[0,1]$.

2.2 节点聚类

在完成时序节点嵌入的基础上, 本文基于节点的嵌入向量, 使用k-means聚类算法将节点划分到不同簇, 以将相似的节点聚类, 从而达到降低算法复杂度的目的. k-means算法以节点嵌入矩阵$U=[\boldsymbol{u}_i]^{\mathrm{T}}$和需要划分的簇个数$k$为输入, 根据节点之间的距离将$N$个节点划分到$k$个簇中. 该算法先随机选择$k$个节点作为质心, 即簇的中心, 再将所有节点分配给距离它最近的质心. 随后, 该算法将更新每个簇的质心, 并将每个节点重新分配给最近的质心, 该过程将被不断重复, 直至新的质心和原质心之间的距离小于某一个设置的阈值.

2.3 种子节点选择

为了降低影响力最大化算法的复杂度, 相较于简单地在每个簇中选取种子节点, 本文将在筛选种子节点之前识别时序网络中的重要簇, 以缩小候选种子节点集的范围, 再在重要簇中识别重要节点作为种子节点. 对于给定的时序网络$G=(V,E,\Gamma)$和种子节点集的规模K, 通过节点聚类得到一组簇$\boldsymbol{C}=\{c_1,c_2,\ldots,c_p\}$, 其中$c_j$为第$j$个簇, n_j为簇c_j中的节点数, 将重要簇$\boldsymbol{C}_s$定义如下:

$$\boldsymbol{C}_s = \left\{c_j\in\boldsymbol{C}\middle|n_j\geq\frac{\sum_{i=1}^{p}n_i}{K}\right\}. \tag{8}$$

识别出重要簇后, 基于配额的思路, 确定给每个重要簇分配的种子节点数. 假设重要簇$\boldsymbol{C}_s=\{c'_1,c'_2,\cdots,c'_q\}$, 并且每个簇$c'_i$有$n'_i$个节点, 则$c'_i$中分配的种子节点数为

$$Q(c'_i) = \left[K\times\frac{n'_i}{\sum_{i=1}^{q}n'_j}\right], \tag{9}$$

其中, $[\cdot]$为取整函数. 在确定了$\boldsymbol{C}_s$中每个重要簇的种子节点个数后, 通过收集每个重要簇中前$Q(c'_i)$个动态度[31]大的节点得到最终的种子节点集. 节点u的动态度$D_T(u)$定义为

$$D_T(u) = \sum_{1<t\leq T}\left[\frac{|N(u,t-1)\setminus N(u,t)|}{|N(u,t-1)\cup N(u,t)|}\times|N(u,t)|\right], \tag{10}$$

其中, $N(u,t)$是节点u在t时刻的邻居节点集.

3 网络传播模型

本文采用易感-感染(SI)传播模型来进行传播模拟, 以检验影响力最大化算法识别超级传播者的效果, 其中S指易感者, 即与感染者接触后可能受到感染的人, 在信息传播模型中也可以理解为没有接收到信息的人, 与接收到信息的人接触后可能会得知信息; I指感染者, 即感染疾病的人, 在信息传播模型中可以理解为已经接收到信息的人. 在传播的开始时刻, 将种子节点集中的节点设定为初始感染节点, 即$I(t_1)=\{u_1,u_2,\ldots,u_K\}$. 在$t_i$ $(i=1,\cdots,T)$时刻, 时序网络中所有处于I状态的节点将独立地以μ的概率感染该网络中处于S状态的邻居, 若感染成功, 则该邻居将在t_{i+1}时刻变为I状态. 重复以上过程后, 将截至T时刻

时序网络中的I状态节点数作为种子节点集的最终传播范围, 传播范围越大, 说明算法效果越好. 为了减小实验随机性的影响, 本文对每个实验均进行了2000次独立的蒙特卡罗仿真来估计传播结果.

4 结果与分析

4.1 数据描述

为了验证本文提出方法在真实数据集中的有效性, 本文使用了以下6个真实数据集, 生成不同的时序网络, 这些网络的基本拓扑性质如表1所示, 其中静态边为将时序网络压缩成静态网络后的边. 具体描述如下:

• Email Eu (EEU1, EEU2): EEU1与EEU2两个数据集收集了欧洲一家大型研究机构的电子邮件数据, 该数据集将人员之间的邮件来往进行匿名化处理[32]. 网络中的节点表示研究机构中发送电子邮件或接收电子邮件的员工. 若在t时刻员工u_i向员工u_j发送电子邮件, 则建立连边(u_i, u_j, t).

• Hypertext 2009 (HT2009): 该数据集收集了2009年ACM Hypertext会议与会者之间的面对面接触数据[33]. 在该数据集中节点表示参加会议的人员, 若在t时刻人员u_i和人员u_j出现接触, 则建立连边(u_i, u_j, t).

• Primary School (PS): 该数据集通过基于射频识别设备的接近感应器, 收集法国某小学在校儿童和教师之间的面对面接触数据[34,35]. 在该数据集中节点表示小学中的儿童或教师, 在t时刻人员u_i和人员u_j出现接触, 则建立连边(u_i, u_j, t).

• Haggle: 该数据集通过无线设备检测人与人之间的接触. 在该数据集中节点表示检测到的参与接触的人员[36]. 在t时刻人员u_i和人员u_j出现接触, 则建立连边(u_i, u_j, t).

• High School 2013 (HS2013): 该数据集通过传感器和日记收集了法国一所高中人员的面对面接触数据[37]. 在该数据集中节点表示该高中的学生或老师, 在t时刻人员u_i和人员u_j出现接触, 则建立连边(u_i, u_j, t).

表 1 数据集拓扑信息

Table 1 Topological information of datasets

网络名称	节点数	时序边数	时间步	静态边数
EEU1	142	48141	26496	1375
EEU2	309	61046	35097	3031
HT2009	113	20818	5246	2196
PS	242	125773	3100	8317
Haggle	274	28244	15662	2124
HS2013	327	188508	7375	5818

4.2 对比方法

本文使用Entropy-Based method, K-Shell, Dynamic Degree, Dynamic Degree Discount和Dynamic Collective Index五种影响力最大化算法进行对比实验. 对比方法的具体描述如下. Entropy-Based method (EB): Michalski等人[38]基于熵定义了一种时序网络上的影响力最大化算法, 即Entropy-Based method. 该算法通过衡量节点覆盖的邻居集合的大小来筛选重要性前K的节点作为种子节点, 节点u_i的重要性由下式得到:

$$EB(u_i) = -\sqrt{|e_{i\cdot}|}\sum_{t=1}^{T-1} p(NB_{u_i,t,t+1})\ln(p(NB_{u_i,t,t+1})), \quad (11)$$

其中, $|e_{i\cdot}|$为节点u_i的总出边(在无向网络中为u_i的所有边)条数, $NB_{u_i,t,t+1}$为节点u_i在相邻的时间窗口$[t, t+1]$ $(t = 1, \ldots, T-1)$中的邻居节点集合. K-Shell (KS): K-Shell分解, 也称为K-核分解, 最早由Carmi等人[39]提出, 它通过剪枝的方式来对网络进行拆解, 以区分节点在网络中所处位置的中心程度. 核值为k的节点的度$\geq k$. K-Shell分解被广泛应用, 在解决影响力最大化和节点中心性问题上有较好的表现[40,41]. 本文将时序网络压缩为静态网络, 计算节点的核值, 随后筛选核值前K的节点作为种子节点集. Dynamic Degree (DD): Habita等人[42]利用节点前后时间之间邻居节点集的对称差定义了时序网络上的动态度, 在该算法中, 动态度前K的节点将被放入种子节点中, 动态度的定义见式(10). Dynamic Degree Discount (DDD): DDD算法是Chen等人[5]提出的静态网络中的度消减算法在时序网络中的拓展. 在该算法中, Murata等人[31]针对时序网络的特点将度消减算法中的静态度修改为动态度, 在每次迭代中, 选择动态度最大的节点放入种子节点集, 并对该节点的邻居的度进行更新, 直至种子节点集的大小达到K. Dynamic CI (DCI): DCI算法是Morone等人[6]提出的静态网络上的CI方法在动态网络中的扩展. 具体而言, 该方法基于节点u_i的度和到节点u_i的距离为l的节点度之和来识别种子节点集, 而

在时序网络中, 节点的度被定义为时序度, 从节点u_1到节点u_k之间长度为$k-1$的路径可以表示为(u_1,u_2,t_1), (u_2,u_3,t_2), …, (u_{k-1},u_k,t_{t-1}), 其中$t_1<t_2<\cdots<t_{k-1}$, 并且$\forall i,\ j\ (i\neq j)$, $u_i\neq u_j$. 从路径的开始到结束的时间跨度$t_{k-1}-t_1$为路径长度, 节点u_1和节点u_k之间的距离即最短路径长度. 在将Morone等人提出的方法中的度和距离扩展到动态网络中后, DCI算法定义了指数$DCI_l(u_i)$:

$$DCI_l(u_i)=D_T(u_i)\sum_{u_i\in DBall(u_i,l)}D_T(u_i), \tag{12}$$

其中, $DBall(u_i,l)$表示到节点u_i距离为l的节点集. l是一个参数, 取值范围为$1\le l\le T$, 本文取$l=1$. 在DCI算法中, 需要计算每个节点的$DCI_l(u_i)$指数, 并选择前K的节点作为种子节点.

4.3 基于传播规模的算法性能比较

本文采用第4.1节中提及的6个数据集开展SI传播实验, 对于较为稠密的网络(HT2009和PS), 使用较小的传播概率$\mu=0.01$, 在其余网络中进行传播时使用的传播概率为$\mu=0.1$, 以确保能较好地区分不同算法的效果. 在对时序网络进行k-means聚类时, 本文将簇个数的取值定为5. 图2展示了本文提出的MMC算法与其余5种对比算法的传播范围比较. 从结果中可以看出, 本文提出的MMC算法在大多数数据集上的传播范围都超过了对比算法. 具体而言, 有以下观察: 在PS和Haggle中(图2(c)和(d)), MMC的传播范围都超过了对比算法, 尤其是在种子节点集大小的取值较大时, 说明了本文提出的算法的有效性. 同时, 还观察到, 随着种子节点数的增大, MMC的算法效果在EEU1和EEU2两个网络中(图2(b)和(e))都实现了比其他算法更大的效果提升. 而在其他的网络中, 如HT2009和HS2013中(图2(a)和(f)), MMC的效果则与其他对比方法接近. 对于动态度(DD)和动态度消减(DDD), 一方面, 二者之间的算法效果在大多数网络上都较为接近, 说明度消减对算法提升的效果较小, 也即在所选数据集中, 大度节点之间的影响力重叠范围比较小; 另一方面, 考虑到本文提出的算法是在每个关键簇中优先选择动态度大的节点放入种子节点

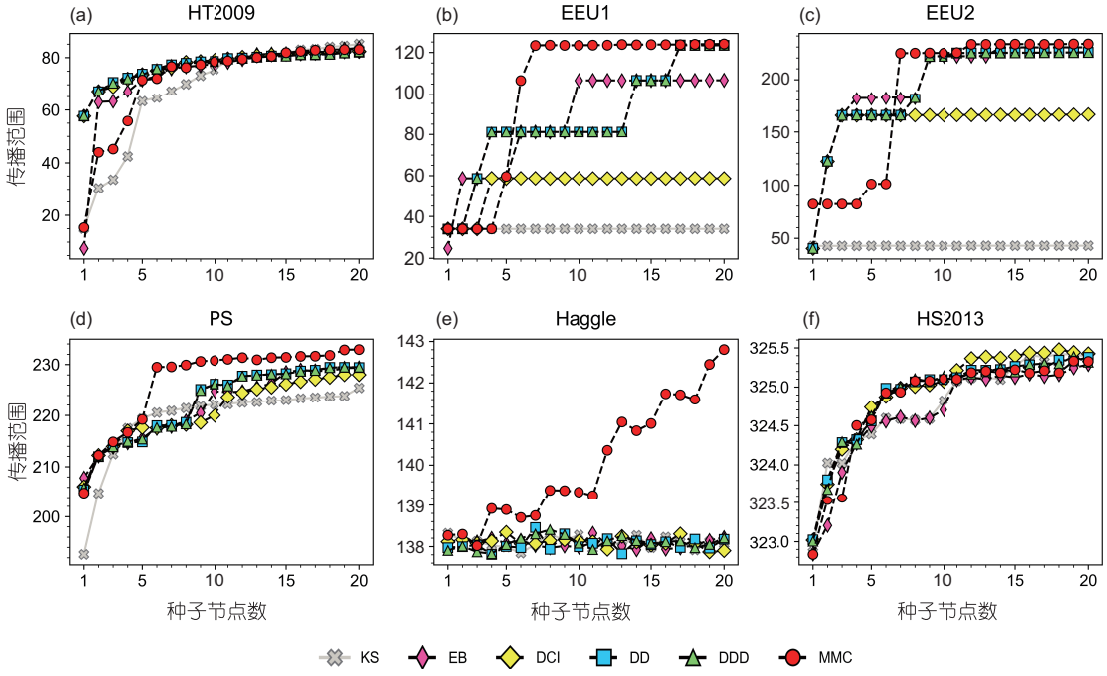

图 2 (网络版彩图)影响力最大化算法在不同时序网络上的传播范围. (a) HT2009; (b) EEU1; (c) EEU2; (d) PS; (e) Haggle; (f) HS2013

Figure 2 (Color online) The influence spread of influence maximization algorithms in different temporal networks. (a) HT2009; (b) EEU1; (c) EEU2; (d) PS; (e) Haggle; (f) HS2013.

集, 而MMC相较于DD和DDD存在较为明显的效果优化, 这从一个侧面说明了嵌入和聚类过程的有效性. 对于DCI算法, 我们发现, DCI仅在HS2013这个数据集上有较好的表现: 在种子节点集的规模超过10时, DCI的算法效果稍好于其他算法. 而在其他网络中, DCI的算法效果则较为一般. EB算法在HT2009, PS, Haggle, EEU2和HS2013这5个网络上的表现都与其他对比算法接近, 且在EEU1, PS, Haggle和HS2013中的传播范围小于所提出的MMC算法. KS算法在大多数网络上的表现都同对比算法相似, 而在EEU1和EEU2两个网络上的传播范围明显小于其他算法, 说明这两个网络的时序信息较为丰富, 将时序网络压缩至静态网络会带来较大的信息流失, 从而导致算法效果大打折扣. 这一观察也表明, 在解决影响力最大化问题的过程中, 考虑时序信息是很有必要的. 综上所述可以发现, 在大多数网络中(如EEU1, EEU2, PS, Haggle), 本文提出的方法都有较好的表现, 尤其在种子节点集的大小取得较大时, MMC的效果会得到显著提升, 这说明相较于其他对比方法, MMC在解决时序网络影响力最大化问题上有较大优势.

4.4 种子节点相似性比较

在验证算法效果的基础上, 为更深入地比较MMC与现有算法之间的区别, 使用Jaccard相似性系数来计算不同影响力最大化算法筛选出的种子节点集之间的相似程度. Jaccard相似性系数是一种用于比较有限样本集S_A和S_B间相似性的测量指标. 定义为S_A, S_B的交集大小除以S_A, S_B的并集大小, 用数学公式表示为

$$J(S_A, S_B) = \frac{|S_A \cap S_B|}{|S_A \cup S_B|}, \tag{13}$$

其中, S_A和S_B分别为影响力最大化算法A和B的种子节点集, $|S_A|$表示集合S_A的大小. 本文计算了种子节点集的大小K取1–20时, 所有方法之间的Jaccard相似性, 并对不同的K值对应的相似性系数取平均值, 得到最终的比较结果. 如图3所示, 在大多数时序网络中, MMC同其他算法之间的相似性较低, 均低



图 3 (网络版彩图)影响力最大化算法之间的Jaccard相似性系数. (a) HT2009; (b) EEU1; (c) EEU2; (d) PS; (e) Haggle; (f) HS2013

Figure 3 (Color online) The Jaccard similarity between influence maximization algorithms. (a) HT2009; (b) EEU1; (c) EEU2; (d) PS; (e) Haggle; (f) HS2013.

于0.5; 仅在HS2013上, 观察到MMC与DD, DDD之间的Jaccard相似性为0.69. 该分析表明, MMC筛选出的种子节点与其他影响力最大化算法之间存在一定程度的差异. 结合4.3节中的分析结论, 可以认为, MMC筛选出的种子节点不仅有较好的传播能力, 同时又跳出了既有方法的框架, 能够筛选出与其他方法不同的种子节点. 对于KS, 观察到该算法与其他算法之间的相似性普遍很低(Jaccard相似性系数不高于0.3), 而在图2中, KS筛选出的种子节点集的传播范围较其他方法较小, 验证了在实际应用中考虑时序信息的必要性和重要性. EB的相似性系数则在不同的网络上呈现不同的模式: 在所有网络上, EB与MMC, KS之间的相似性都较低, 它仅在HS2013数据集中与MMC的相似性为0.48, 在其他数据集中与MMC的相关性则为0.25–0.4; 其与KS的相关性则不高于0.24 (Haggle); 而在EEU1, EEU2, PS和Haggle四个网络中, EB与DCI, DD, DDD之间的相似性超过了0.5. 对于DCI算法, 其与KS, MMC之间的相似性普遍较低, 例如, 在数据集HT2009中, DCI与KS的相似性为0.073, 与MMC的相似性为0.31; 而与DD, DDD之间的相似性普遍较高, 均高于0.5, 它与DD, DDD之间的相似性在HS2013上相似性最小, 为0.62. DD和DDD也有相似的模式, 并且二者之间有很强的相似性, 在所有的网络上, 二者之间的相似性系数都达到1, 这与4.3节中的实验结果一致. 以上分析在宏观上对比了影响力最大化算法之间的相似性与差异性, 为了更细致地探究MMC在筛选高影响力种子节点方面与其他算法间的差异, 本文选取K为20时, 所有影响力最大化算法在Haggle中筛选出的种子节点集作为分析对象. 具体而言, 首先筛选出不在对比方法的种子节点集中, 但存在于MMC筛选出的种子节点集中的节点集合, 记为$\bar{S}$, 然后计算这些节点在对比方法中的节点排名. 如表2所示, MMC筛选出了8个异于对比方法的节点, 而这些节点在对比方法中的排名均较为落后. 例如, 节点176在KS, EB, DCI, DD和DDD中的排名分别为103, 130, 75, 76和76, 意味着这些节点均被对比方法识别为较不重要的节点. 而结合图2(e)中的结果可以发现, 这些节点未被其他算法识别出, 是令MMC的算法效果超过其他对比算法的关键节点. 通过以上相似性分析发现, MMC和KS与其他影响力最大化算法之间的Jaccard相似性较低, 但与KS相比, MMC筛选出的种子节点具有更高的传播影响力; 而其他对比方法之间均在一些数据集中具有较高的相似性.

表 2 $\bar{S}$中的节点在对比方法中的排名. 数据集为Haggle, 种子节点集的大小为20

Table 2 Rankings of nodes in $\bar{S}$ in baselines. The dataset is Haggle, and the size of seed node set is 20

$\bar{S}$	KS	EB	DCI	DD	DDD
176	103	130	75	76	76
179	121	162	78	77	77
66	193	236	92	92	92
67	131	151	93	93	93
68	132	174	94	94	94
73	61	66	56	54	54
98	73	82	118	118	118
228	129	136	79	82	82

4.5 参数敏感性

为了进一步验证算法的普适性, 测试了MMC对于聚类个数的参数敏感性. 图4显示当簇的个数取不同值(1–10)时, MMC在不同网络上的传播范围. 首先, 发现相比于聚类个数取1, 其他的取值对应的传播范围都有所提升, 这说明了节点聚类对提高传播范围的有益效果. 其次, 当改变聚类个数(k取值大于1时), MMC在大多数网络中的传播结果都较为相近, 例如, 在时序网络HT2009, EEU1, PS, Haggle和HS2013中, 传播范围的最小值和最大值之间的差值在1–5个节点; 而在时序网络EEU2中, 改变聚类个数可以为算法带来15个感染节点的提升. 总体而言, 聚类个数的具体取值存在相对较优的值, 调整聚类个数可以在一定程度上提升算法的效果, 在具体应用中, 可根据需求来确定最终的聚类个数.

5 讨论和结论

本文基于时序网络节点嵌入和簇聚类方法, 提出了一种新的影响力最大化算法MMC. 首先对时序网络的微观和宏观动力学特征进行嵌入, 得到了时序网络的结构和规模特征. 接着, 采用簇聚类算法将时序网络分成多个簇, 以降低算法的计算复杂度, 降低种子节点集之间的重叠度. 最后, 优先选择动态度较大的节点放

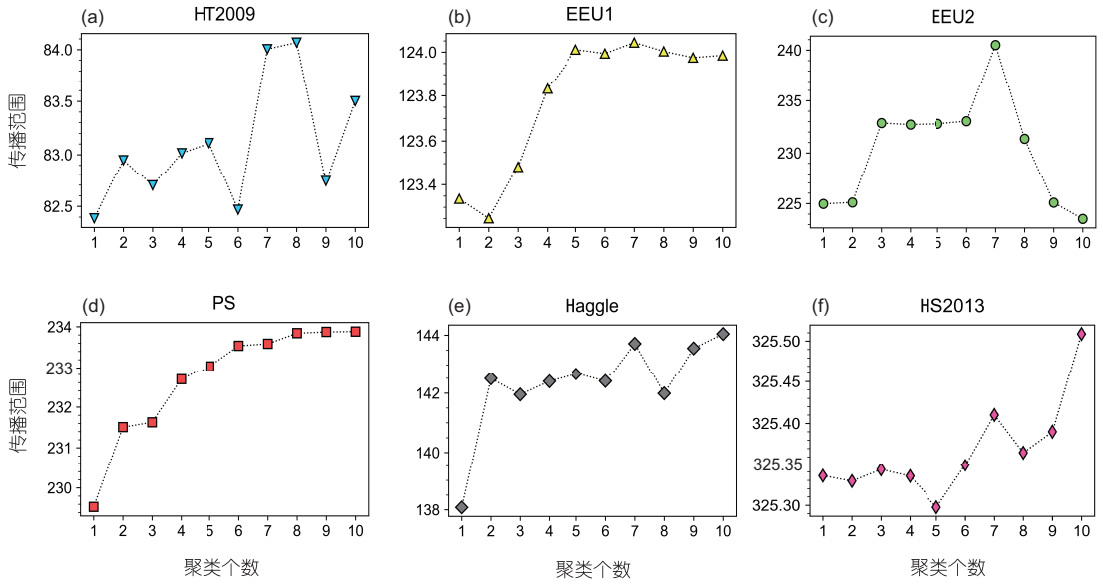

图 4 (网络版彩图)不同聚类个数下, MMC在不同时序网络上的传播范围. 种子节点集的规模统一为20. (a) HT2009; (b) EEU1; (c) EEU2; (d) PS; (e) Haggle; (f) HS2013

Figure 4 (Color online) The influence spread of MMC in different temporal networks by using different clustering numbers. The size of seed set is set to 20. (a) HT2009; (b) EEU1; (c) EEU2; (d) PS; (e) Haggle; (f) HS2013.

入种子节点集, 从而实现影响力最大化. 将MMC与其他5种影响力最大化算法进行比较实验. 在6个数据集上的实验结果表明, MMC算法在大多数时序网络中都能取得比其他对比算法更大的传播范围. 通过影响力最大化算法的相似性分析发现, 所提出的算法与其他对比算法之间存在差异性: MMC在实现较高的传播影响力的同时, 能够跳出现有方法的框架, 筛选出不同的高影响力种子节点. 除此之外, 本文的分析还表明, 适当调整聚类个数能够提升MMC算法的效果, 从一定程度上解决了实际应用过程中的参数选择问题. 综上所述, 我们所提出的基于时序网络节点嵌入和簇聚类的影响力最大化算法可以帮助我们更好地理解和优化时序网络上的传播过程以及寻找种子节点集, 从而帮助更好地解决现实中的经济与管理问题. 未来, 我们将继续从以下几个方面拓展和完善本文的工作: 首先, 探索更多适用于时序网络的复杂网络分析算法, 与静态网络相比, 时序网络的拓扑结构具有动态特性, 而现有的网络分析方法仍以基于静态网络为主, 将考虑时序维度的影响, 将传统的复杂网络分析方法推广到时序网络中. 其次, 采用深度学习模型或者高阶结构[43]来对时序网络进行更加准确和可靠的节点嵌入, 以解决影响力最大化问题, 深度学习模型具有强大的表达能力和泛化能力, 在解决时序网络问题时具有很大的潜力; 同时, 还将研究高阶结构在时序网络影响力最大化算法中的应用, 例如基于张量分解的节点嵌入算法等. 第三, 在解决实际问题时, 还需要结合更多业务场景需求来进一步完善算法, 影响力最大化问题是一个具有广泛应用场景的问题, 不同的业务场景可能会有不同的约束条件和优化目标. 因此, 将根据实际需求, 进一步优化算法, 例如考虑影响力最大化的预算、考虑不同节点的权重等. 总之, 希望通过努力, 能够在时序网络的影响力最大化问题上有更多的突破.

参考文献

1 Kempe D, Kleinberg J, Tardos é. Maximizing the spread of infuence through a social network. In: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Washington: ACM Press, 2003. 137–146

2 Domingos P, Richardson M. Mining the network value of customers. In: Proceedings of the 7th ACM SIGKDD International Conference on

Knowledge Discovery and Data Mining. San Francisco: ACM Press, 2001. 57–66
3 Chen W. Research on influence diffusion in social network (in Chinese). Big Data Res, 2015, 1: 75–91 [陈卫. 社交网络影响力传播研究. 大数据, 2015, 1: 75–91]
4 Leskovec J, Krause A, Guestrin C, et al. Cost-effective outbreak detection in networks. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Jose: ACM Press, 2007. 420–429
5 Chen W, Wang Y, Yang S. Efficient influence maximization in social networks. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Paris: ACM Press, 2009. 199–208
6 Morone F, Min B, Bo L, et al. Collective influence algorithm to find influencers via optimal percolation in massively large social media. Sci Rep, 2016, 6: 1
7 Zhang X L, Tang J X, Cao L C. Influence maximization algorithm based on reverse PageRank (in Chinese). Int J Comput Appl Technol, 2020, 40: 96–102 [张宪立, 唐建新, 曹来成. 基于反向PageRank的影响力最大化算法. 计算机应用, 2020, 40: 96–102]
8 Gao C, Jiang S H, Wang Z, et al. A novel method to identify influential stations based on dynamic passenger flows (in Chinese). Sci Sin-Inf, 2021, 51: 1490–1506 [高超, 蒋世洪, 王震, 等. 基于动态客流的城市轨道交通关键站点识别. 中国科学: 信息科学, 2021, 51: 1490–1506]
9 Fang H, Jiang J L, Li S Q, et al. The influence maximization algorithm based on local propagation centrality (in Chinese). Comput Appl Softw, 2023, 40: 259–264 [方辉, 姜久雷, 李盛庆, 等. 基于局部传播中心性的影响力最大化算法. 计算机应用与软件, 2023, 40: 259–264]
10 Chen J Y, Zhang D J, Lin X, et al. False message propagation suppression based on influence maximization (in Chinese). Comput Sci, 2020, 47(S1): 17–23, 33 [陈晋音, 张敦杰, 林翔, 等. 基于影响力最大化策略的抑制虚假消息传播的方法. 计算机科学, 2020, 47(S1): 17–23, 33]
11 Wang Y, Cong G, Song G, et al. Community-based greedy algorithm for mining top-k influential nodes in mobile social networks. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Washington: ACM Press, 2010. 1039–1048
12 Chen Y C, Zhu W Y, Peng W C, et al. CIM. ACM Trans Intell Syst Technol, 2014, 5: 1–31
13 Huang H, Shen H, Meng Z, et al. Community-based influence maximization for viral marketing. Appl Intell, 2019, 49: 2137–2150
14 Cencetti G, Battiston F, Lepri B, et al. Temporal properties of higher-order interactions in social networks. Sci Rep, 2021, 11: 7028
15 Cheng C H, Kuo Y H, Zhou Z. Outbreak minimization vs influence maximization: An optimization framework. BMC Medical Inform Decis Mak, 2020, 20: 1–13
16 Coró F, D'angelo G, Velaj Y. Link recommendation for social influence maximization. ACM Trans Knowl Discov Data, 2021, 15: 1–23
17 Azaouzi M, Mnasri W, Ben Romdhane L. New trends in influence maximization models. Comput Sci Rev, 2021, 40: 100393
18 Li Y, Fan J, Wang Y, et al. Influence maximization on social graphs: A survey. IEEE Trans Knowl Data Eng, 2018, 30: 1852–1872
19 Liu X Y, He D B, Liu C, et al. Social network information propagation dynamic analysis based on time-varying damping motion (in Chinese). Sci Sin-Inf, 2021, 51: 1867–1884 [刘小洋, 何道兵, 刘超, 等. 基于时变阻尼运动的社交网络信息传播动力学分析. 中国科学: 信息科学, 2021, 51: 1867–1884]
20 Wang H, Zhan X X. Information diffusion backbone. In: Temporal Network Theory. Cham: Springer, 2019. 199–217
21 Chen S, Ren Z M, Liu C, et al. Identification methods of vital nodes on temporal networks (in Chinese). J Univ Electron Sci Technol China, 2020, 49: 291–314 [陈诗, 任卓明, 刘闯, 等. 时序网络中关键节点的识别方法研究进展. 电子科技大学学报, 2020, 49: 291–314]
22 Liu X, Liao X, Li S, et al. On the shoulders of giants: Incremental influence maximization in evolving social networks. Complexity, 2017, 2017: 1–14
23 Holme P, Saramäki J. Temporal networks. Phys Rep, 2012, 519: 97–125
24 Holme P. Modern temporal network theory: A colloquium. Eur Phys J B, 2015, 88: 234
25 Cui P, Wang X, Pei J, et al. A survey on network embedding. IEEE Trans Knowl Data Eng, 2018, 31: 833–852
26 Hamilton W, Ying Z, Leskovec J. Inductive representation learning on large graphs. In: Advances in Neural Information Processing Systems 30. Red Hook: Curran Associates, Inc., 2017. 1025
27 Mohan A, Pramod K V. Link prediction in dynamic networks using time-aware network embedding and time series forecasting. J Ambient Intell Hum Comput, 2021, 12: 1981–1993
28 Su X, Xue S, Liu F, et al. A comprehensive survey on community detection with deep learning. IEEE Trans Neural Netw Learn Syst, doi: 10.1109/TNNLS.2021.3137396
29 Tang D, Du W, Shekhtman L, et al. Predictability of real temporal networks. Natl Sci Rev, 2020, 7: 929–937
30 Lu Y, Wang X, Shi C, et al. Temporal network embedding with micro- and macro-dynamics. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management. Beijing: ACM Press, 2019. 469–478
31 Murata T, Koga H. Extended methods for influence maximization in dynamic networks. Comput Soc Netw, 2018, 5: 1–2
32 Paranjape A, Benson A R, Leskovec J. Motifs in temporal networks. In: Proceedings of the 10th ACM International Conference on Web Search

and Data Mining. Cambridge: ACM Press, 2017. 601–610
33 Isella L, Stehlé J, Barrat A, et al. What's in a crowd? Analysis of face-to-face behavioral networks. J Theor Biol, 2011, 271: 166–180
34 Gemmetto V, Barrat A, Cattuto C. Mitigation of infectious disease at school: Targeted class closure vs school closure. BMC Infect Dis, 2014, 14: 1
35 Stehle J, Voirin N, Barrat A, et al. High-resolution measurements of face-to-face contact patterns in a primary school. PloS One, 2011, 6: e23176
36 Hui P, Chaintreau A, Scott J, et al. Pocket switched networks and human mobility in conference environments. In: Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-tolerant Networking. Philadelphia: ACM Press, 2005. 244–251
37 Mastrandrea R, Fournet J, Barrat A. Contact patterns in a high school: A comparison between data collected using wearable sensors, contact diaries and friendship surveys. PLoS ONE, 2015, 10: e0136497
38 Michalski R, Jankowski J, Pazura P. Entropy-based measure for influence maximization in temporal networks. In: Computational Science–ICCS 2020: 20th International Conference. Amsterdam: Springer, 2020. 277–290
39 Carmi S, Havlin S, Kirkpatrick S, et al. A model of internet topology using k-shell decomposition. Proc Natl Acad Sci USA, 2007, 104: 11150–11154
40 Samir A M, Rady S, Gharib T F. LKG: A fast scalable community-based approach for influence maximization problem in social networks. Physica A, 2021, 582: 126258
41 Wu H, Yue K, Fu X, et al. Parallel seed selection for influence maximization based on k-shell decomposition. In: Collaborate Computing: Networking, Applications and Worksharing: 12th International Conference, CollaborateCom 2016. Beijing: Springer, 2017. 27–36
42 Habita Y Y, Berger-Wolf T Y, Saia J. Finding spread blockers in dynamic networks. In: Advances in Social Network Mining and Analysis: 2nd International Workshop, SNAKDD 2008. Las Vegas: Springer, 2010. 55–76
43 Zhang K, Gao J Y, Hu W J, et al. Tightly related sets and collective degree distribution on hypernetworks (in Chinese). Sci Sin-Phys Mech Astron, 2023, 53: 270511 [张科, 高靖宇, 胡文军, 等. 超网络中的紧密相关集和组度分布特性研究. 中国科学: 物理学 力学 天文学, 2023, 53: 270511]

A node-embedding-based influence maximization algorithm in temporal network

ZHAN Xiu-Xiu[1,2], XIE XiaoWen[1], ZHANG KaiYue[1], LIU Chuang[1] & ZHANG Zi-Ke[2,3*]

[1]*Research Center for Complexity Sciences, Hangzhou Normal University, Hangzhou 311121, China;*
[2]*College of Media and International Culture, Zhejiang University, Hangzhou 310058, China;*
[3]*Research Center for Digital Communications, Zhejiang University, Hangzhou 310058, China*
**Corresponding author (email: zkz@zju.edu.cn)*

The rapid growth of social networks has resulted in the increased application of influence maximization algorithms in various fields, such as viral marketing and rumor containment. It searches for a set of seed nodes of a given size that can maximize the range of influence propagation. Several influence maximization algorithms have been proposed; however, the temporal characteristics of complex systems are rarely considered. To address this issue, we propose an influence maximization algorithm based on temporal network node embedding called micro- and macro-based clustering (MMC). The proposed algorithm can effectively learn the micro- and macro-dynamic characteristics of temporal networks by node embedding. We cluster the temporal network based on the embedding vectors to decrease the complexity of the influence maximization algorithm and the overlapping of influence spread among seed node sets, thereby further improving the performance of the algorithm. Furthermore, we conducted diffusion and similarity comparison experiments on real-world temporal networks to validate the effectiveness of MMCs. The results revealed that compared to five other comparative algorithms, the seed nodes selected by MMC not only guarantee greater influence spread but also go beyond the framework of existing algorithms. The analysis of parameter sensitivity revealed that effective clustering of nodes improves algorithm performance.

influence maximization, temporal network, node embedding, clustering, spreading model

PACS: 89.75.-k, 89.75.Da, 89.75.Hc, 89.75.Fb, 89.75.Kd

doi: 10.1360/SSPMA-2023-0134