第 57 卷 第 4 期 厦门大学学报(自然科学版) Vol. 57 No. 4
2018 年 7 月 Journal of Xiamen University (Natural Science) Jul. 2018
doi:10.6043/j.issn.0438-0479.201708002

网络基础设施中重要网元子图的确定

刘 峥[1,2*],郭舒婷[1,2],周绮凤[3],李 涛[1,2]
(1.南京邮电大学计算机学院,江苏 南京 210023;2.江苏省大数据安全与智能处理重点实验室,江苏 南京 210023;3.厦门大学航天航空学院,福建 厦门 361005)

摘要:网元子图是指大规模网络基础设施中包含承载具体业务网元的拓扑子图,网元子图可用于网络基础设施运维中的故障排查、诊断与修复.首先定义重要网元的概念;其次,为确定重要网元子图,提出一个统一框架来度量网元在结构和业务两方面的影响力,将其作为重要网元的衡量标准,并设计了从重要网元扩展生成重要网元子图的高效算法.基于真实的网络基础设施数据以及合成的业务承载数据进行实验,实验结果验证了该方法可以高效地找到高质量的重要网元子图,并用于网络基础设施的运维,提高运维的效率,节省运维的成本.

关键词:网元子图;邻域影响度;故障网元;网络基础设施运维
中图分类号:TP 274 **文献标志码**:A **文章编号**:0438-0479(2018)04-0558-07

目前,现代大规模复杂网络基础设施的日常运维往往依赖于集中式网络监控平台和人工干预来保证网络服务的可靠性[1-2].重要网元的监视和管理对保障网络服务的可靠性至关重要.网元的重要性往往由网络管理员根据经验来判断,然而在网络基础设施规模日益增大和网络管理人力资源有限的背景下,对于承载重要业务的网元相对较易确定,而对于处于网络基础设施拓扑结构中的重要节点的判断却很难由人工判定,常常出现错误及遗漏.

网络基础设施中面向业务的网元子图是指网络基础设施中承载具体业务的重要网元的拓扑子图,也包括其他非重要网元.通过确定面向业务的网元子图,可以满足大规模网络基础设施中运维需求,仅需投入较少人力对这些拓扑子图中的网元进行重点监控,就可以提高故障分析的效率,节省网络基础设施运维的成本.目前在网络基础设施运维中并没有成熟的网元子图的确定方法,传统的方法是从发生故障的网元出发,将其邻域网元所组成的子图作为网元子图.

在社交网络研究领域,已经有一些学者对节点的影响力进行了研究,并将高影响力的节点作为重要节点.Kempe 等[3]研究了影响最大化问题,集中介绍了描述节点激活行为的模型.Leskovec 等[4]提出一种高性价比的懒惰前向(cost-effective lazy forward, CELF)贪心算法的优化策略,运用独立级联型的次模特性(sub-modularity),大大降低了选择最具影响力节点的工作次数.此外,Lei 等[5]提出了一个能够在市场病毒式营销中学习到传播概率的方法.Farajtabar 等[6]通过对社会活动进行建模,构建了一个凸优化框架确定激励用户刺激社交活动所需的活动水平.但这些社交网络中的影响力评估算法着重关注社交网络中信息的传播模型,而不是故障传播模型,并不适用于评估网络技术设施中网元的影响力.在图数据挖掘的研究领域中,已经有一些学者提出了一些节点之间关系的度量方法.Jeh 等[7]提出了一种通过节点邻域的相似性来度量节点之间的相似度的方法,称为 SimRank,表征从两个节点出发的随机游走的期望相遇距离.Palmer 等[8]定义了一个距离函数来衡量两个属性数据集之间的相似度,用图模型来表示属性值,两个属性值之间的距离定义为重启随机游走的逃逸概率.但这些重要性的度量方法都没有考虑到网络基础设

收稿日期:2017-08-01 录用日期:2018-03-21
基金项目:江苏省自然科学基金(BK20171447);江苏省高等学校自然科学研究项目(17JKB520024);2015 年度教育部-中国移动科研基金项目(5-10);南京邮电大学引进人才科研启动基金(NY215045)
* 通信作者:zliu@njupt.edu.cn
引文格式:刘峥,郭舒婷,周绮凤,等.网络基础设施中重要网元子图的确定[J].厦门大学学报(自然科学版),2018,57(4):558-564.
Citation:LIU Z,GUO S T,ZHOU Q F,et al.Element subgraph discovery in networks infrastructures[J].J Xiamen Univ Nat Sci,2018,57(4):558-564.(in Chinese)

施中网元重要性的特点,没有将重要节点的邻域节点进行深入分析.此外,由于拓扑结构和承载业务两者的异构性,也不能直接利用上述方法度量网元的重要性.另一方面重要网元子图的确定与社交网络研究领域中的社区发现有一定相似.目前,对图上的社区发现算法主要包括:基于谱分析的聚类算法[9],基于图划分的聚类算法[10],基于层次的聚类算法[11]以及基于密度的聚类算法[12].Liu 等[13]搜索网络中的极大团,并依据其连接情况合并极大团来获得网络的社区结构.Pons 等[14]提出利用长度为 l 的随机游走来度量图上两个节点的相似度用于社区发现.Tong 等[15]也提出了利用重启随机游走在图数据上发现中心子图.但社交网络中的社区发现偏重于寻找社交网络中的相似节点的集合,而不是根据节点的影响力来判断节点的影响范围.

为了同时考虑网元连接关系的重要性和承载业务的重要性,本研究将业务转换为图模型上的点,利用邻域影响度来衡量网元之间(图上节点之间)的连接关系,并由此评估节点的重要性.识别重要节点后,根据其重要性衡量标准,将确定网元子图的问题变成一个在图上对节点进行分组的问题.为此,本文中提出了一个 3 步的框架来识别重要网元子图:

1) 评估网元节点的重要性,寻找重要的网元节点;

2) 从每个重要网元节点出发,根据节点之间的相关性,生成网元重要子图;

3) 融合网元重要子图,形成网元重要子图集合.

1　评估网元节点的重要性

为了同时考虑网元连接关系的重要性和承载业务的重要性,首先需要弄清重要网元的特征.从大规模复杂网络运维实际出发,网元的重要性主要体现在以下两方面:1) 重要网元与其他网元的直接连接或间接连接的关系较密切,体现在重要网元与其他网元之间的连通路径较多,一旦重要网元发生故障,可能会影响其他网元之间的通信或服务交互.2) 重要网元所承载的业务的数量较多,级别也较高.一旦重要网元发生故障,会影响到承载同样业务的其他网元服务的稳定性.

根据重要网元的特征,本研究将业务转换为图模型上的节点,再利用邻域影响度捕捉节点的连接关系,进而将连接关系的重要性和承载业务的重要性统一起来.本文中的网络基础设施用图模型 $G=\langle V,E\rangle$ 表示,其中 V 是所有网元节点 v_i 所组成的集合,E 表示网元之间连接关系的集合,即图上两个节点之间的边 e_i 的集合.

对于用图模型 G 表示的网络基础设施,令 A 表示有权重图的邻接矩阵.$A(i,j)$表示边 $e_{ij}=(v_i,v_j)$ 上的权重.基于随机游走的概念,在图 G 上的影响力扩散是指从某一节点 v_0 出发,并按一定概率在图上沿节点和边随机移动.假设目前第 v_s 步的随机游走在节点 $s+1$ 上,则第 $s+1$ 步将以概率 P_{st} 移动到 v_s 的某一相邻节点 v_t 上,即 $v_t\in N(v_s)$,其中 $N(v_s)$表示节点 v_s 在图 G 上的所有相邻节点的集合,其转移概率 $P_{st}=A(s,t)/\sum_{v_k\in N(v_s)}A(s,k)$,所经过的节点路径组成一个马尔科夫链.令 $\boldsymbol{D}$ 表示一个对角矩阵,其中对角线上某个值 $d(s)=\sum_{v_k\in N(v_s)}A(s,k)$,则对应的马尔科夫链的转移概率矩阵 $\boldsymbol{P}$ 的矩阵表示为 $\boldsymbol{P}=\boldsymbol{D}^{-1}\boldsymbol{A}$.从节点 v_j 到节点 v_k 的长度为 l 的概率 Q_{jk} 可以通过转移矩阵 $\boldsymbol{P}$ 经过 l 次乘积后相应位置上的对应元素来表示,故 $\boldsymbol{Q}=(Q_{jk})=\boldsymbol{P}^l$.

当 G 为非二项图时,由于马尔科夫链的无记忆性,从任何节点出发,经过无限步的影响度最后落在同一个节点 v_k 的概率只和最终节点的度的大小有关.可以通过返回概率 c,将影响度倾向于在出发节点的周围的局部的连接关系,同时只考虑长度为 l 的随机游走,即邻域影响度所能达到的节点与出发的节点的最短路径长度不超过 l.值得注意的是,本文中提出的算法可以根据不同需要自由地增大 l 以扩大邻域的范围.

定义 1　邻域影响度:若随机游走的长度为 l,则节点 v_j 到节点 v_k 的邻域影响度(影响力)为

$$\Pi_{jk}=\sum_{\tau:v_j,\cdots,v_k;le(\tau)\leqslant l}P(\tau)c(1-c)^{le(\tau)},$$

其中,$0<c<1$,τ 是从节点 v_j 到节点 v_k 的一条路径,$le(\tau)$表示路径 τ 的长度,$P(\tau)$为节点 v_j 到节点 v_k 的转移概率.

由此可知邻域影响度的矩阵可表示为:

$$\boldsymbol{\Pi}=\sum_{\gamma=1}^{l}c(1-c)^{\gamma}\boldsymbol{P}^{\gamma}. \tag{1}$$

根据邻域影响度(影响力)矩阵,重要网元节点的重要性可定义为

$$S(v_j)=\sum_{v_k\in V}\Pi_{jk}, \tag{2}$$

其中,重要性分值 $S(v_j)$表示节点 v_j 的影响力.是根据重要节点对其他节点的影响力,即邻域影响度长度的总和来决定.

式(2)定义网络基础设施图模型 G 上的重要性,

为了同时考虑承载业务的重要性,本文中提出将业务也转换为图模型上的点,如图 1 所示.假设网络基础设施中的业务有$\{w_1,w_2,w_3,\cdots\}$.每种业务 w_j 在图上都映射相应的节点 w_j.如果某网元承载某业务,则在图上增加连接网元对应节点 v_i 到业务对于节点 w_j 的一条边 $e(v_i,w_j)$.这样生成的包括业务信息的图模型用 G'表示.在图 G'上利用邻域影响度来统一衡量节点的重要性,可同时衡量连接关系的重要性和关于业务的重要性.对于业务的级别,可以通过给边 $e(v_i,w_j)$赋予相应的权重,为简化描述,本文中假设所有业务级别都一样,即相应的边的权重为 1.下文中将不区分 G'和 G,两者都表示包括业务信息的图模型.

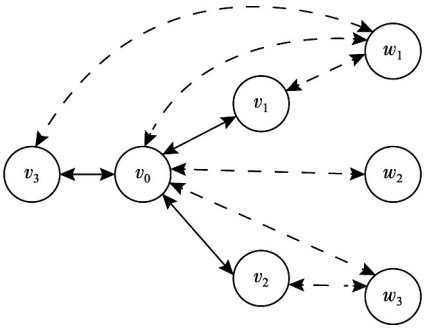


图 1 业务转化为图上的节点

Fig. 1 Service vertices in the graph

网元节点重要性评估算法步骤为:

1) 计算转移概率矩阵 $\boldsymbol{P}$.将业务转化为图上业务节点,利用图上网元节点的邻接关系和网元承载业务情况得到邻接矩阵 $\boldsymbol{A}$,根据 $\boldsymbol{P}=\boldsymbol{D}^{-1}\boldsymbol{A}$ 计算转移概率矩阵 $\boldsymbol{P}$.

2) 计算影响力距离矩阵 $\boldsymbol{\Pi}$.设置合适的随机游走长度 l 和随机游走的重启概率 c,根据式(1) 计算 $\boldsymbol{\Pi}$.

3) 根据式(2) 计算网元节点的重要性分值.

4) 生成重要网元节点列表.重要性分值大于 ξ 的网元节点称为重要节点,其中 ξ 表示重要网元节点重要性的阈值.图 2 所示的是基于本文中实验部分数据集的网络基础设施图 G 上的网元重要性的分布情况.通过对网元节点影响力的曲线拟合,可见网元节点影响力分布曲线与幂律分布函数 $y=1.72x^{-0.06}-1$ 的曲线基本吻合,遵循幂律(power law)分布,因此确定重要节点的重要性的阈值 ξ 并不是绝对的,而是相对的.在下文 3.2 节对 ξ 的取值进行了讨论.

由于评估节点重要性需要对两个矩阵进行乘积,所以整体的时间复杂度是 $O(ln^3)$,其中 n 是图中的网元个数.在实际应用中,可以利用快速稀疏矩阵乘法来代替通常的矩阵相乘算法以加快计算速度.

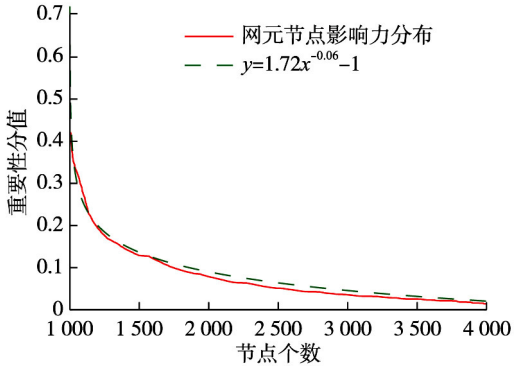


图 2 网元节点影响力分布曲线

Fig. 2 The curve of vertex influence distribution

2 生成重要网元子图

重要网元子图应包括重要网元节点和与重要网元节点相似度较大的节点,即被重要节点影响大的节点,有如下几个特点:1) 重要网元子图应该是一个连通图;2) 重要网元子图应该包括重要节点;3) 重要网元子图应该包括被重要节点影响的节点.

本文中提出了一个类似高维空间中密度聚类思想的扩展策略,从重要网元节点扩展出网元子图.生成重要网元子图的算法步骤如下:

1) 从重要网元节点列表中选择当前重要性分值最大的网元 v_j,初始化子图 g_s 为空,并将该网元插入到空节点图中,从重要网元节点列表中删除 v_j,标记 v_j 为已访问节点,设置子图扩展的停止条件阈值 ε,计算公式为:

$$\varepsilon=\max(\boldsymbol{\Pi}(j,:))\times r_{\mathrm{lb}},$$

其中 r_{lb}的取值为幂律分布曲线的拐点位置.

2) 初始化最大堆 $\boldsymbol{H}$,将所有未被访问过的重要网元 v_j 的相邻节点 v_m 和影响力值 Π_{jm} 插入到最大堆 $\boldsymbol{H}$ 中.

3) 从最大堆 $\boldsymbol{H}$ 的堆顶获取当前影响力最大值节点 v_k 和其影响力值,若该影响力值大于阈值 ε,将所有未被访问过的 v_k 的相邻节点 v_m 和影响力值 Π_{km} 插入到最大堆 $\boldsymbol{H}$ 中,同时将 v_k 加入到图 g_s 中,标记 v_k 为已访问,如果节点 v_k 是重要网元,更新阈值 ε,更新公式为:

$$\varepsilon=\max(\boldsymbol{\Pi}(k,:))\times r_{\mathrm{lb}}.$$

重复步骤 3),若该影响力值小于阈值 ε,则将当前生成的子图 g_s 加入到重要网元子图集合 G_s 中;若重要

网元节点列表不为空,返回步骤 1);否则进行步骤 4).

4) 融合重要网元子图.遍历重要网元子图集合 G_s,若两个重要网元子图存在节点在原图上是直接相连的,将两个网元子图合并成一个网元子图.

生成重要网元子图的算法从重要网元节点出发,通过扩展重要网元来生成重要网元子图.步骤 2)和 3)使用最大堆来维持影响力大的节点,每次从最大堆 $\boldsymbol{H}$ 中取出第一个节点加入到当前的重要网元子图 g_s 中.检查是否结束当前重要网元子图 g_s 的扩展的判断条件是当前网元子图 g_s 中的所有节点对待扩展节点的影响力是否小于阈值 ε.阈值 ε 为最后加入到网元子图的重要节点对图上其他节点的影响力的最大值的 r_{lb} 倍.由于影响力距离遵循幂律分布,r_{lb} 的取值为幂律分布曲线的拐点位置,在本文中根据实验中的具体幂律分布情况,r_{lb} 的值为 0.2.在步骤 4)中,对于生成的重要网元子图集合,如果两个重要网元子图存在节点在原图上是直接相连的,那么这两个网元子图将被合并成一个网元子图.

3　实验结果与分析

本文中实验所使用的运行环境为 OS X 10.10.5,CPU 为 Intel i5 1.6 GHz,内存为 4 GB,算法的实现基于 Python 语言.在所有实验中,使用的随机游走的返回概率为 0.15.

3.1　数据集

本研究在数据集 SNAP1N、SNAP1U、SNAP2N、SNAP2U 上进行实验.这 4 个数据集由数据集 SNAP1 和 SNAP2 生成,数据集 SNAP1 和 SNAP2 均来源于俄勒冈大学路由查看计划,来自 Stanford 大学的 SNAP 项目(https://snap.stanford.edu/data/).两个数据集的基本特征如表 1 所示.

表 1　数据集特征

Tab.1　Dataset characteristics

数据集	节点数	边数
SNAP1	2 107	4 489
SNAP2	6 474	13 233

数据集中每个节点代表一个自治系统(如路由),自治系统之间的通信遵循边界网关协议,基于边界网关协议的日志消息可以构建自治系统之间的通信网络拓扑图.由于 SNAP1 和 SNAP2 中并没有网元所承载的业务信息,本研究中在其上叠加了合成的业务数据.假设有 100 种不同的业务,对于每个节点,允许其承载 10 个不同的业务,承载的业务的数量有均匀分布和正态分布两种情况.随机地从 100 种业务中选择某节点所承载的具体业务.通过对 SNAP1 和 SNAP2 叠加两种不同的业务分布,共生成 4 个不同的数据集供实验中使用,后文用 SNAP1U、SNAP1N、SNAP2U、SNAP2N 表示.其中,SNAP1U 表示在 SNAP1 上叠加均匀分布的业务数量,SNAP1N 表示在 SNAP1 上叠加正态分布的业务数量.SNAP2 的表示与之类似.

3.2　有效性

首先介绍如何衡量重要网元子图的质量.对于确定的某重要网元子图 G_s,可以利用式(3)来衡量重要网元子图的质量:

$$R_{\mathrm{coverage}}(g)=\frac{\sum_{v_j,v_k\in V(G_s),\mathrm{score}(v_j)\geqslant\xi}\Pi_{jk}}{\sum_{v_j\in V(G_s),v_k\in V(G),\mathrm{score}(v_j)\geqslant\xi}\Pi_{jk}},\quad(3)$$

其中 $\boldsymbol{\Pi}$ 是影响力矩阵.$\sum_{v_j,v_k\in V(G_s),\mathrm{score}(v_j)\geqslant\xi}\Pi_{jk}$ 表示重要网元子图内节点之间的影响力,$\sum_{v_j\in V(G_s),v_k\in V(G),\mathrm{score}(v_j)\geqslant\xi}\Pi_{jk}$ 衡量了重要网元子图内的重要节点到子图内其他节点的影响力,R_{coverage} 值越高,说明重要网元子图内捕捉到大部分重要节点的影响力越大.

图 3 显示了在数据集 SNAP1U 和 SNAP1N 上,对于不同的 ξ 找到的前 10 个重要网元子图的 R_{coverage} 的算术平均值.对于每个 ξ,变化邻域影响度的长度为 2～6.当随机游走的长度为 2 时,R_{coverage} 的算术平均值在两个数据集上都只有 0.3 左右,这是因为长度为 2 的随机游走在图上所能访问范围过小,不足以捕捉重要节点的影响力.当随机游走的长度超过 3 时,发现两个数据集上的 R_{coverage} 的算术平均值都超过 70%,例如对于长度为 5,ξ 为 85%时,重要网元子图内的重要节点的影响力超过 85%都在子图内部.

图 4 展示了分别在数据集 SNAP1U 和 SNAP1N 上找到的重要网元子图,其中随机游走的长度为 4,ξ 为 85%,灰色节点表示重要网元子图中的重要节点.可见,找到的重要网元子图里包括相当部分的重要节点,也包括与重要节点相连的非重要节点.寻找网元子图的传统方法是从同一时间窗口内的每个故障网元出发,以每个故障网元为中心,寻找与其路径距离小于 h 的网元节点.将故障网元以及这些故障网元的邻接网元节点所构成的子图作为此故障网元的邻域子图,即为重要网元子图.同一时间窗口内的故障网元邻域子图如有重合,即包含相同节点,则把重合的邻域子图合并.图 5 展示了

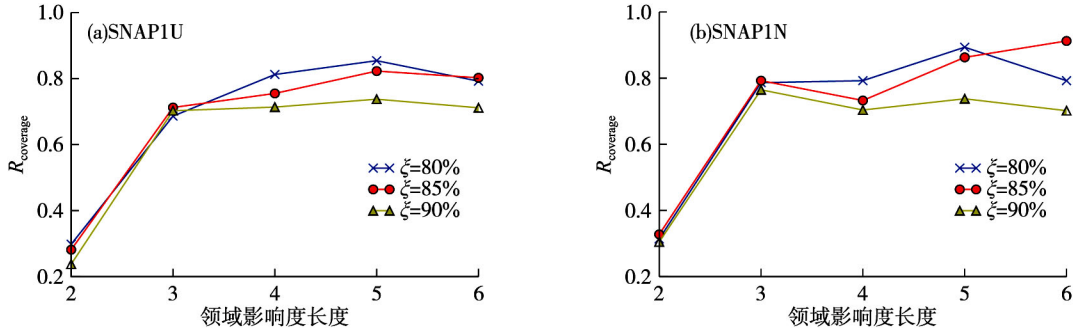
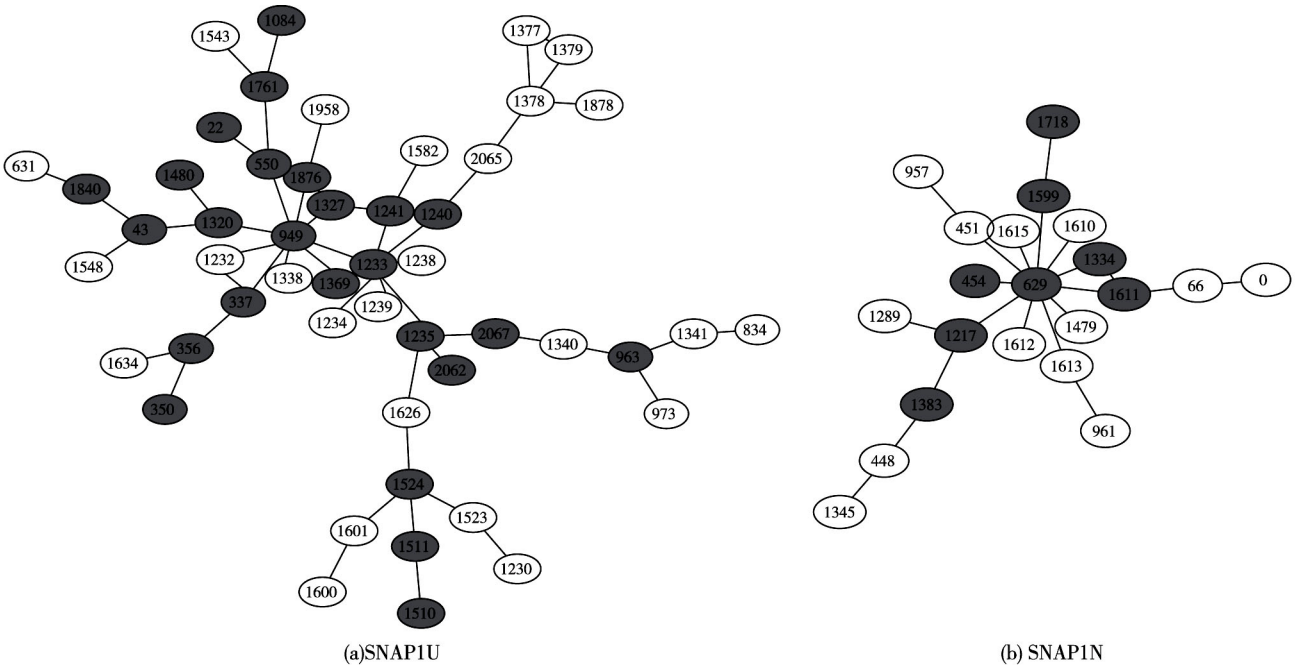


图 3 数据集 SNAP1U 和 SNAP1N 的 $R_{coverage}$

Fig. 3 $R_{coverage}$ of dataset SNAP1U and SNAP1N



图中数值表示网元的编号,下同.

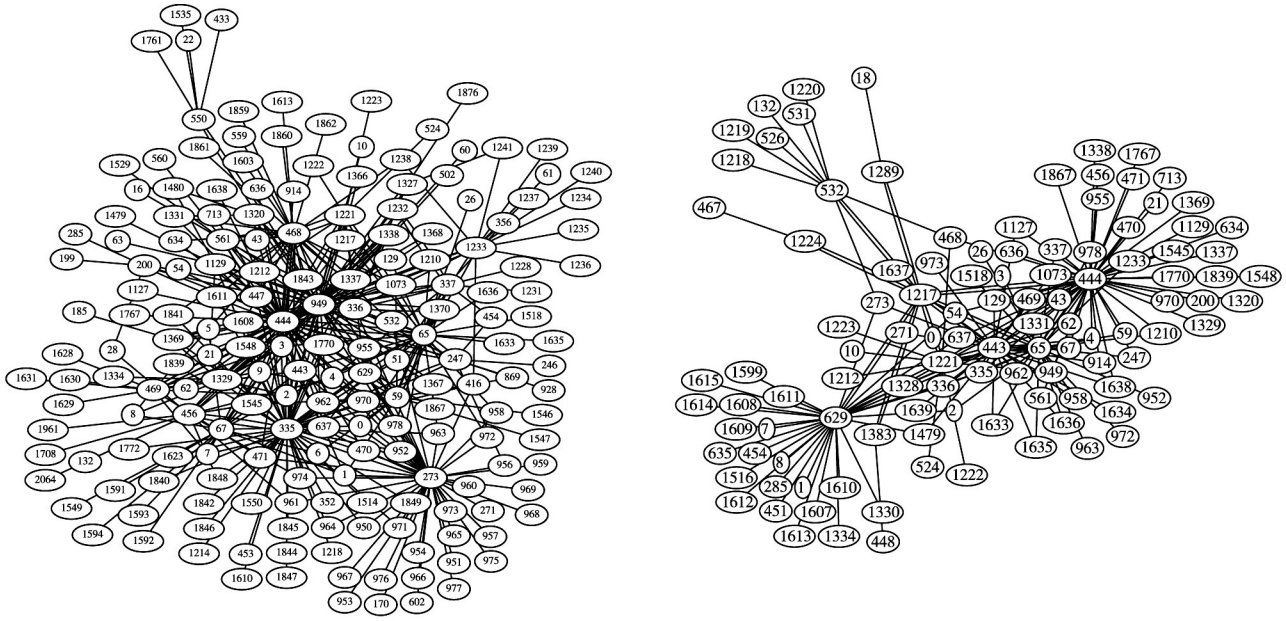
图 4 数据集 SNAP1U 和 SNAP1N 的重要网元子图示例

Fig. 4 An element subgraph of dataset SNAP1U and SNAP1N

在数据集 SNAP1U 和 SNAP1N 上利用传统的方法找到的与图 4 对应的网元子图,即包括发生故障网元的邻域子图,其中邻域子图中的节点到发生故障节点的路径长度为 2.可见,传统方法找到的网元子图包括的节点远远多于本文中所找到的重要网元子图中的节点,会极大地增加运维人员的负担.

3.3 运行时间

本研究在 4 个数据集上都进行了运行时间的实验,其中随机游走的长度为 4 和 6,ξ 为 85%.图 6 显示了在 SNAP1U、SNAP1N、SNAP2U、SNAP2N 的运行时间,包括长度为 4 和 6 的随机游走.运行时间分两部分,分别对应评估网元节点的重要性和生成重要网元子图.其中,SNAP1U 和 SNAP1N 的实验结果对应左边的坐标纵轴,SNAP2U 和 SNAP2N 的实验结果对应右边的坐标纵轴.由于评估网元重要性算法的时间复杂度为 $O(ln^3)$,所以评估网元节点重要性所需的时间随 l 的增加而增加.生成重要网元子图算法是一个启发式的算法,其时间复杂度与重要节点的个数以及重要节点的影响范围有关,但 l 的变化会影响重要节点的个数,所以生成重要网元算法的时间也会随 l 的变化而变化.由于生成重要网元子图中要频繁更新最大堆中所对应的影响力值,其时间复杂度为 $O(\log m)$,其中 m 是最大堆里的元素个数,所以扩展生成重要网



(b)SNAP1U　　(b) SNAP1N

图 5　数据集 SNAP1U 和 SNAP1N 的传统网元子图示例

Fig. 5　An element subgraph of dataset SNAP1U and SNAP1N

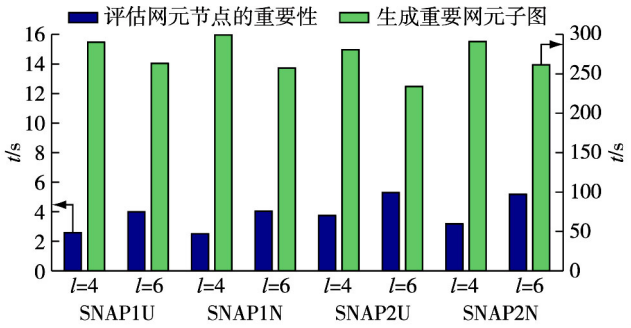


图 6　数据集的运行时间

Fig. 6　The running time on datasets

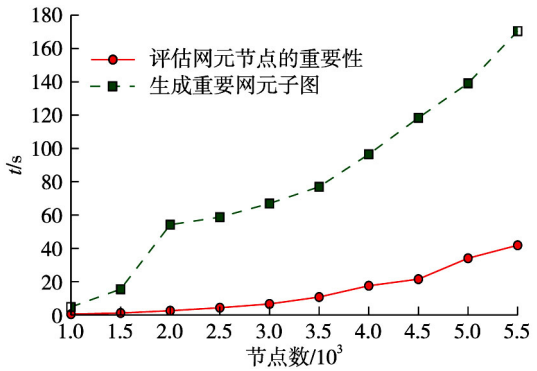


图 7　数据集 SNAP2N 的运行时间

Fig. 7　The running time on dataset SNAP2N

元子图的时间相对较长.

为了衡量算法的时间复杂度,本研究中分别选取不同节点数,在数据集 SNAP2N 上进行实验.图 7 显示了在不同大小的图上算法的运行时间,其中随机游走的长度为 4,ξ 为 85%.可见,随着节点数量的增加,计算网元重要性算法和生成重要网元子图算法的运行时间符合上面的复杂度分析,整体上呈多项式时间增长.当节点数为2 000时,生成重要网元子图步骤的变化也反映了启发式算法的启发策略对整体运行时间的影响.

4　结　论

快速而准确地确定重要网元子图可以用于提高网络基础设施的运维效率,降低运维成本.本研究的主要贡献:1) 针对在大规模网络基础设施中确定面向业务的重要网元子图的问题,提出利用邻域影响度来统一衡量网元在连接和业务两方面的重要性;2) 给出了确定重要网元子图的算法,利用类似密度聚类的思想,通过扩展重要网元生成重要网元子图;3) 在真实的网络基础设施数据集上,通过叠加合成的业务承载数据进行了广泛的实验,实验结果验证了提出的方法可以高效地找到高质量的重要网元子图,并将其用于网络基础设施的运维.未来可能的研究方向包括利用其他影响力模型来构建重要网元子图,以及在动态网络基础设施中如何维护所发现的重要网元子图等.

参考文献:

[1] TANG L, LI T, SHWARTZ L, et al. An integrated framework for optimizing automatic monitoring systems in large IT infrastructures[C]// Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Chicago: ACM, 2013: 1249-1257.

[2] 李涛. 数据挖掘的应用与实践: 大数据时代的案例分析[M]. 厦门: 厦门大学出版社, 2015: 8-9.

[3] KEMPE D, KLEINBERG J, TARDOS É. Maximizing the spread of influence through a social network[C]// Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Washington DC: ACM, 2003: 137-146.

[4] LESKOVEC J, KRAUSE C, GUESTRIN C, et al. Cost-effective outbreak detection in networks [C]// Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2007: 420-429.

[5] LEI S, MANIU S, MO L, et al. Online influence maximization[C]// Proceedings of ACM SIGKDD International Conference on Konwledge Discovery and Data Mining. Sydney: ACM, 2015: 645-654.

[6] FARAJTABAR M, DU N, RODRIGUEZ M G, et al. Shaping social activity by incentivizing users[J]. Advances in Neural Information Processing Systems, 2014, 27: 2474-2482.

[7] JEH G, WIDOM J. SimRank: a measure of structural-context similarity [C]// Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2002: 538-543.

[8] PALMER C. R, FALOUTSOS C. Electricity based external similarity of categorical attributes[C]// Proceedings of Pacific-Asia Conference on Knowledge Discovery and Data Mining. New York: ACM, 2003: 486-500.

[9] NEWMAN M E. Spectral methods for community detection and graph partitioning[J]. Physical Review E Statistical Nonlinear and Soft Matter Physics, 2013, 88(4): 042822.

[10] NEWMAN M E J. Community detection and graph partitioning[J]. EPL, 2013, 103(2): 330-337.

[11] LIN C C, KANG J R, CHEN J Y. An integer programming approach and visual analysis for detecting hierarchical community structures in social networks[J]. Information Sciences, 2015, 299: 296-311.

[12] REN J, WANG J, LI M, et al. Identifying protein complexes based on density and modularity in protein-protein interaction network [J]. BMC Systems Biology, 2013, 7(S4): 1-15.

[13] LIU G, WONG L, CHUA H N. Complex discovery from weighted PPI networks [J]. Bioinformatics, 2009, 25 (15): 1891-1897.

[14] PONS P, LATAPY M. Computing communities in large networks using random walks[J]. Journal of Graph Algorithms and Applications, 2006, 10(2): 191-218.

[15] TONG H, FALOUTSOS C. Center-piece subgraphs: problem definition and fast solutions [C]// Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2006: 404-413.

Element Subgraph Discovery in Networks Infrastructures

LIU Zheng[1,2*], GUO Shuting[1,2], ZHOU Qifeng[3], LI Tao[1,2]

(1. School of Computer Science and Technology, Nanjing University of Posts and Telecommunications, Nanjing 210023, China; 2. Jiangsu Key Laboratory of Big Data Security & Intelligent Processing, Nanjing 210023, China; 3. School of Aerospace Engineering, Xiamen University, Xiamen 361005, China)

Abstract: In many applications, graphs are used to model structural relationships among objects. Large scale network infrastructures can be represented as graphs, where element subgraphs are those subgraphs containing important network elements with many connections and running services. In this paper, we formularize the problem of discovering element subgraphs in network infrastructures. Element subgraphs can help network administrators lower the cost for network infrastructure operation and maintenance. A uniform framework is proposed to model the element importance by using neighborhood influence based on random walk, which considers both structural connections and running services on these network elements. We design an efficient algorithm that skillfully finds the important element subgraphs by expanding the important vertices. Our experiments are based on real data sets with synthetic service information, whose results show that our element subgraphs exhibit high quality.

Key words: element subgraphs; neighborhood influence; faulty elements; network infrastructure management