



基于节点中心性排序和交替乘子法的网络重构

王建波^{1,2*}, 刘江华¹, 杜占玮^{2*}, 李平¹, 许小可^{3,4*}

1. 西南石油大学计算机与软件学院, 成都 610500

2. 香港大学公共卫生学院, 香港 999077

3. 北京师范大学计算传播学研究中心, 珠海 519087

4. 北京师范大学新闻传播学院, 北京 100875

* 通信作者. E-mail: jwang@swpu.edu.cn, zwdu@hku.hk, xuxiaoke@bnu.edu.cn

收稿日期: 2024-09-10; 修回日期: 2024-10-30; 接受日期: 2025-01-08; 网络出版日期: 2025-03-18

国家自然科学基金 (批准号: 62173065)、北京市自然科学基金 (批准号: 4242040)、深港澳科技计划项目 (C 类项目) (批准号: SGDX20230821091559022) 和智慧警务与国家安全风险治理重点实验室 2024 年度开放课题重点项目 (批准号: ZHKFZD2401) 资助

摘要 如何从观测时间序列来重构复杂网络结构是网络科学领域的热点和难点问题. 近年来统计推断和压缩感知两类方法广泛用于网络重构, 然而代表性重构方法的精度、速度和抗噪声性能均有大幅提升空间. 本文在经典压缩感知算法的基础上, 提出了一种基于节点中心性排序和交替乘子法的网络重构算法. 首先对节点状态特征矩阵按照节点中心性的判断方法重新排序重构节点的顺序, 然后利用压缩感知对排序后的节点时间状态特征矩阵进行处理, 最后引入交替乘子法优化矩阵实现网络拓扑结构的重构. 理论分析证明, 所提算法的复杂度低于代表性的压缩感知重构方法. 实验结果表明, 本文将所提算法相比于现有代表性重构方法, 提升了重构精度, 减少了重构时间, 具有更高的抗噪声鲁棒性. 本研究提升了网络重构算法应用于大规模网络的准确性、适用性和可靠性, 并可在未来扩展到不同类型约束的加权网络重构中.

关键词 网络重构, 压缩感知, 时间序列, 节点中心性, 交替乘子法

1 引言

1.1 研究背景

现实世界中随处可见复杂网络, 无论是细胞网络、生物网络还是社交网络、互联网等各式各样的网络, 使得研究者们可以从网络结构特征^[1]、功能性质及其动力学^[2~5]等不同角度进行深入探讨. 然而在许多实际情况下, 已知网络节点状态的观测数据, 而拓扑结构则是未知或不确定的^[6,7], 但准确的复杂网络结构信息是全面理解网络动力学和群体行为的重要条件, 这就需要对网络结构进行重构. 复杂网络的重构通常基于网络节点状态的观测数据, 需要在给定节点的复杂动力系统中对网络节点进行观测. 由于实验成本和测量技术的限制, 直接获取网络拓扑结构的信息非常困难. 一个巨大的挑战是,

引用格式: 王建波, 刘江华, 杜占玮, 等. 基于节点中心性排序和交替乘子法的网络重构. 中国科学: 信息科学, 2025, 55: 949–966, doi: 10.1360/SSI-2024-0276

Wang J B, Liu J H, Du Z W, et al. Network reconstruction based on node centrality sorting and alternating direction method of multipliers. Sci Sin Inform, 2025, 55: 949–966, doi: 10.1360/SSI-2024-0276

在网络结构未知的情况下,只有有限的观测时间序列可用^[8].因此,开发一种能够从有限测量数据中提取网络结构信息的通用方法非常重要.从观测数据中重构复杂网络的结构,即反问题,已成为当前网络科学和工程的一个热点问题^[9~12].

1.2 相关工作

为解决上述问题,研究者们提出了许多创新的方法来重构复杂网络的拓扑结构.这些方法通过不同的技术手段和理论基础,旨在从有限的观测数据中准确重构网络的结构.下面主要介绍两类典型的网络重构方法.

(1) 基于统计推断的方法:利用概率统计的方法计算测量矩阵中的数值,推断网络中节点的连接概率,从而重构网络的拓扑结构.这类方法通过统计推断来估计节点之间的潜在连接,实现网络拓扑结构的还原.例如,Sachs等^[13]使用贝叶斯网络分析生物数据集,发现其可以揭示直接和间接的分子相互作用,提出了一种贝叶斯网络计算方法来预测复杂的网络拓扑.Liu等^[6]在未知动力学网络上,提出了一种识别不确定广义时滞复杂网络的拓扑结构方法,并将其推广到在线演化.Xu等^[14]提出了一种非线性控制方案,用于识别具有非延迟和时变延迟耦合的加权广义复杂网络模型的拓扑结构.Wu等^[15]提出了一种基于格兰杰(Granger)因果检验的复杂网络拓扑重构方法,并针对不同的情况,提出了不同的格兰杰因果关系检验识别方法.Liu等^[16]通过重构耦合的泰勒(Taylor)展开系数,开发了一种新方法重构连续时变和状态相关网络的结构.Ma等^[17]通过双层网络上传播动力学模型,基于平均场极大似然估计实现多路复用网络结构的重构.Wu等^[18]使用布尔(Boole)概率理论结合递归神经网络方法,讨论了基于先验布尔函数和采样数据的概率布尔网络的最优重建问题.Wang等^[19]开发了一个结合统计推理和期望最大化的通用框架,基于两种离散状态动力学的二元时间序列数据去重构网络模型.

(2) 基于压缩感知的方法:压缩感知已经在许多网络重构领域得到了广泛应用.它不仅有效减少了所需数据量,显著提升了重构效率和精度,还在通信网络、社交网络、生物网络等多个领域展示了其优越性,推动了各类网络分析和设计的发展.例如,Chen等^[20]证明了压缩感知在重构无线传感器网络中的有效性,提出了一种采样率指示器反馈方案,最大限度地降低能耗以保持可接受的重建性能.Shen等^[8]使用了一种基于压缩感知理论(compressive sensing theory, CST)的框架成功重构发生随机传播动力学的网络,并且对网络中的隐藏资源进行了相应的重构.Xie等^[21]提出了一种基于半传感器乘积的信号压缩和重构新模型,从相干性、受限等距特征方面进行研究,对传统CST进行推广.Zhang等^[22]为了克服投影矩阵依赖于网络动力学的时间序列数据这个困难,通过调整乘数的交替方向方法来找到候选邻接矩阵.Dai和Liu^[23]通过分析和提取未加权和无向网络的潜在特征,使用一种结构化压缩感知方法从而全局重构复杂网络的拓扑结构.

除此之外,网络重构的方法还有很多,例如,在非线性动力学系统中,Zhou等^[24]对具有特定拓扑结构的复杂网络中存在的几何特征、同步以及控制等问题进行了相关的研究,利用Lyapunov理论,提出了一种自适应反馈控制方法来识别一个通用的加权动态网络模型的拓扑;对于非线性集群动力学模型的推理,Li等^[25]提出了一种基于时效网络上发生的扩散过程的首次时间的逆建模方法,并利用人工网络 and 实际网络验证了算法的有效性;Casadiego等^[26]通过生成的非线性集群动力学以及引入显式依赖矩阵来揭示网络(两点)和超网络(三点)交互的能力;在多路复用延时网络中,Zhao等^[27]在延迟网络的拓扑识别中,提出一种有效恢复各层拓扑以及节点系统参数的算法,同时运用一个双层网络来验证结果的有效性.

在统计推断的网络重构中不依赖于网络结构的稀疏性假设,适用于各种类型的复杂网络,能够建模复杂网络中的各种关系和结构特征.当数据量充足时,统计推断方法能够充分利用数据的统计特性,通常能够更精确地重构网络结构.使用其他网络重构的方法有着一定的局限性,其条件通常更为严格,运用范围相对来说较小,并且由于本文使用的数据是高维稀疏数据,压缩感知对稀疏数据网络重构相

对而言更具有优势, 其原因是, 它能够在较少量观测数据下有效重构网络, 利用了信号的稀疏性或可压缩性, 通过少量的采样点, 仍能重建复杂网络的全貌, 对于处理大规模网络、数据缺失或采样成本高的场景更具有优势.

1.3 动机与贡献

压缩感知^[28]最初用于数值计算、信号处理和计算机视觉领域, 是一种应用于网络重构领域的新方法^[29]. 根据压缩感知, 稀疏信号恢复的任务是将线性问题转化为可以通过某种贪婪或凸优化方法解决的优化问题, 在多个动力学模型上的适用性已被证明, 可用于揭示时间序列^[30]中耦合振荡网络或进化博弈的节点相互作用模式, 揭示流行病传播网络^[8], 但主要用于规模较小的网络, 当网络规模超出一定阈值后很难满足限制等距性质 (restricted isometry property, RIP) 条件^[31]. 然而对于最优化理论中的交替乘子法 (alternating direction method of multipliers, ADMM) 而言, 其更擅长对大规模优化问题进行处理, 所以在此基础上引入了交替乘子法对网络节点时间矩阵进行最优化操作, 来提高网络矩阵的计算操作流程. 根据上述讨论, 本文结合压缩感知开发了一个基于节点中心性的从时间序列重构复杂传播网络的通用理论框架, 将两者结合提出了一种基于节点中心性排序和交替乘子法的网络重构 (node centrality sorting and ADMM, NCSA) 算法, 运用交替乘子来优化矩阵, 最终达到快速高效重构复杂网络的目的. 网络重构其本质是揭示网络节点之间的连边关系. 而这种连边关系主要由 0, 1 二值来表示, 其表达方式使得网络重构的过程更容易被解释和理解. 因为其结构相对简单, 所以在对观测数据进行统计和计算的过程也更为方便. 本文所使用的 SIS 模型不仅更契合研究的需求, 还能够将该方法运用到二值动力学系统的网络重构中, 例如 Ising 动力学、演化博弈动力学等, 所以文中假设只有二进制时间序列可以测量, 它表示疫情暴发后任何时间网络中节点的状态, 无论是感染还是易感.

本文的主要贡献总结如下.

(1) 提出了基于节点中心性排序和交替乘子法的网络重构算法, 通过算法分析得出所提算法的复杂度低于代表性的压缩感知重构方法.

(2) 利用节点中心性来改变压缩感知重构节点的顺序, 按照优先级的降序排列依次对节点的邻居进行重构操作, 然后引入交替乘子法, 将其运用于问题分解和局部重构问题, 即将原来存在的一个大问题拆分为多个更小的子问题来处理操作, 使得较大规模网络的重构成为可能.

(3) 在多个真实和人工网络的重构实验中, 证明了所提算法的重构精度以及鲁棒性优于代表性的压缩感知重构方法, 同时减小了重构计算时间.

本文的其余部分组织如下. 第 2 节对现有的压缩感知网络重构方法以及问题进行了回顾. 第 3 节介绍了节点中心性排序压缩感知和交替乘子法的网络重构方法的设计与分析. 第 4 节对多个人工和真实网络进行了一系列实验来验证所提出方法的有效性和鲁棒性. 最后, 第 5 节对本文的工作进行了总结和讨论.

2 问题描述

为了使用压缩感知算法重构 $Y = \Phi X$ 形式的向量 X , 矩阵 Φ 需要满足限制等距性质 (RIP). 复杂网络的精确重构结果表明, 由于传播动力学的内在随机性, RIP 的条件得到了满足. 文中采用二进制时间序列作为压缩感知时传入的数据类型, 向量 X 的稀疏性以及二值性的特征使得在对网络拓扑进行重构过程中只需要较少的数据量. 通过有限观测数据进行复杂网络拓扑结构识别操作可以转化为从具有比方程更多未知数的线性欠定方程组重构向量 $x \in \mathbb{R}^n$. 可以用 $\Phi x = y$ 来表示, 其中 $\Phi \in m \times n$, $y \in \mathbb{R}^m$, $m < n$, m 是测量的数量. 在本文中, 假设矩阵 Φ 具有全行秩, x 是稀疏的. 在这种情况下, 去

求解最小化 x 的非零分量的数量, 以获得 $\Phi x = y$ 的稀疏解, 即通过压缩感知解决以下优化问题^[32]:

$$\begin{aligned} \min_x & \|X\|_0 \\ \text{s.t. } & \Phi x = y. \end{aligned} \quad (1)$$

寻找方程 (1) 的稀疏解通常是一个 NP 难题^[33]. 最近, 计算的难处理性导致了问题 (1) 的替代方案的发展, 已被证明等效于以下内容:

$$\begin{aligned} \min_x & \|X\|_1 \\ \text{s.t. } & Ax = y, \end{aligned} \quad (2)$$

其中 $\|x\|_1 = |x_1| + |x_2| + \cdots + |x_n|$ 是稀疏向量 X 的 $1NF(l^1)$ 范数, l^1 范数是凸函数. 在很多情况下, 求解 $\Phi x = y$ 会产生数值不稳定性. 为了增加数值稳定性, 将使用经典正则化:

$$\min_x \frac{1}{2} \|\Phi x - y\|_2^2 + \alpha \|x\|_1, \quad (3)$$

其中 α 是为避免与最优解有较大偏差而采用的正则化参数^[34,35].

值得注意的是, 使用压缩感知的稀疏信号重构的一个关键区别在于, 网络结构重构任务中的投影矩阵是由网络动力学定义的, 该网络动力学往往不能满足 RIP 条件:

$$(1 - \psi_s) \|X\|_2^2 \leq \|\Phi X\|_2^2 \leq (1 + \psi_s) \|X\|_2^2.$$

当重构网络规模较大时, 存在着中心节点, 中心节点与其他边缘节点的连接边数之间的差距可能存在跨越几个数量级的差距, 其邻接向量可能不稀疏, 因此需要更多的观测数据来精确地重构这些向量, 在实际情况下几乎不可能收集到如此庞大的数据. 这一问题是在压缩感知在直接应用于网络重构时不可避免的缺陷.

在本文中, 节点中心性定义为通过统计节点状态时间序列中 01, 11 对的数量, 其数量越多表示该节点在网络中被感染的概率越高, 从而可以间接说明该节点在网络中的邻居数也越多, 具有更高的中心性. 不将 10, 00 对作为判断条件的原因是节点重构是其自发的, 不受邻居节点的影响. 因此本文提出节点中心性压缩感知, 其目的是通过优先重构这些高中心性的节点来加速网络的重构, 确保资源被有效利用, 使得网络可以在较短的时间内重构其最大的价值. 然而, 对于压缩感知后的邻接矩阵来说, 重构效率并不能达到预期的效果. 因此, 引入交替乘子法来对矩阵进一步优化操作以此来提升重构精度. 相比之下, 交替乘子法擅长处理大规模优化问题, 通过分解策略, 将任务分解, 从而大大减少网络重构所需的时间. 同时在分布式架构中, 如果某个节点出现故障, 其他节点可以接管其任务, 这提高了整个系统的稳定性和容错性. 本文首先对节点时间片进行“中心”节点的处理操作. 使得网络中心性更高的节点能够更快地重构, 在此基础上进行矩阵处理操作, 最后在处理的矩阵基础上使用交替乘子法算法添加相关的约束条件进一步优化处理, 使得其准确率得到更大的提升.

3 算法设计与分析

本文提出的基于节点中心性排序和交替乘子法的网络重构 (NCSA) 算法主要包括以下步骤.

(1) 基于 SIS 流行病传播模型产生的节点时间状态矩阵设置相应的阈值以及汉明 (Hamming) 距离的条件要求来进行目标状态时间片的选取, 记为测量矩阵 X . 对测量矩阵按照本文所提的节点中心性判断条件准则对矩阵进行重新排序操作, 得到输出后的测量矩阵 $\tilde{X}$.

(2) 使用压缩感知对测量矩阵 $\tilde{X}$ 进行操作, 在进行 L_1 正则化最小值优化时, 引入交替乘子法算法对 $\tilde{X}$ 进行问题分解和局部重构优化操作, 同时设置算法迭代阈值以及停止准则的对偶残差, 以此实现时间复杂度的降低使得算法运行时间减少, 最终通过交替乘子法算法优化输出得到相应的网络邻接矩阵.

3.1 流行病传播模型

在流行病传播的过程中, 网络中的节点状态伴随时间的推移而不断变化, 其变化特征受动力学模型以及网络中节点的相互影响, 即网络中节点之间状态转换存在着一定的概率, 本文将概率经过等式转化操作后引入到重构参数中. 假设疾病从少数感染节点开始传播, 可以通过节点的时间序列数据来重构网络拓扑结构. 将网络中任意节点 i 的状态记为 S_i , 其中,

$$S_i = \begin{cases} 0, \text{易感}, \\ 1, \text{感染}. \end{cases} \quad (4)$$

对于 SIS 动力学, 任意节点 i 在时刻 t 被其邻居感染的概率 $P_i^\lambda(t)$ 为

$$P_i^\lambda(t) = 1 - (1 - \lambda_i)^{\sum_{j=1, j \neq i}^N a_{ij} S_j(t)}, \quad (5)$$

其中 λ_i 代表网络中节点的感染率, a_{ij} 表示网络邻接矩阵中的元素 (如果 $a_{ij} = 1$ 则表示网络中节点 i 和 j 之间存在连接, 否则 $a_{ij} = 0$), $S_j(t)$ 是节点 j 在 t 时刻的状态. 与此同时, 节点 i 的恢复率 $P_i^\delta(t) = \delta_i$, 其中 δ_i 代表节点 i 的恢复率^[8]. 为了对上式进行化简操作使之满足压缩感知的基本等式, 对式 (5) 两边取对数可改写为

$$\ln[1 - P_i^\lambda(t)] = \ln(1 - \lambda_i) \sum_{j=1, j \neq i}^N a_{ij} S_j(t). \quad (6)$$

如果测量的时间段是可用的, 上式可以转化为矩阵形式 $Y_{m \times 1} = A_{m \times (N-1)} \times X_{(N-1) \times 1}$, 其中 Y 表示不同时间段的 $\ln[1 - P_i^\lambda(t)]$, 对于一般网络, 包含 i 的链接节点邻接矩阵 X 是稀疏的.

对于 SIS 动力学, 假设可以在有限的时间序列 $t = t_1, t_2, \dots, t_m$ 中进行测量. 令其 $\bar{\Gamma}_i(t) = \ln[1 - P_i^\lambda(t)]$, $\Omega_{ij} = \ln(1 - \lambda_i) a_{ij}$ 可得出以下矩阵形式 $Y_{m \times 1} = A_{m \times (N-1)} \times X_{(N-1) \times 1}$:

$$\begin{bmatrix} \bar{\Gamma}_i(t_1) \\ \bar{\Gamma}_i(t_2) \\ \vdots \\ \bar{\Gamma}_i(t_m) \end{bmatrix} = \begin{bmatrix} S_1(t_1) & \dots & S_{i-1}(t_1) & S_{i+1}(t_1) & \dots & S_N(t_1) \\ S_1(t_2) & & S_{i-1}(t_2) & S_{i+1}(t_2) & \dots & S_N(t_2) \\ \vdots & & \ddots & \vdots & & \\ S_1(t_m) & \dots & S_{i-1}(t_m) & S_{i+1}(t_m) & \dots & S_N(t_m) \end{bmatrix} \begin{bmatrix} \Omega_{i1} \\ \vdots \\ \Omega_{i,i-1} \\ \Omega_{i,i+1} \\ \vdots \\ \Omega_{iN} \end{bmatrix}, \quad (7)$$

其中, 向量 $X_{(N-1) \times 1}$ 可以表示为网络中节点 i 和其他节点之间的可能连接, $X_{(N-1) \times 1}$ 在一般的复杂网络中通常是稀疏的. 所以如果向量 $Y_{m \times 1}$ 和矩阵 $A_{m \times (N-1)}$ 可以从观测的节点时间序列中得到, 那么 $X_{(N-1) \times 1}$ 就可以通过 CST 来求解.

3.2 节点中心性状态矩阵选取

为了开发一种从节点状态估计节点之间连边概率的方法, 本文统计了每个节点的时间状态序列, 对每个时间状态序列进行筛选处理操作, 根据节点的中心性来优先对中心特征较高的节点进行排序重构操作, 其核心思想在于, 在网络中节点中心性较高的节点所处位置更具优势. 这类节点能够迅速地将信息传播至网络的其他部分, 能够以较快的速度覆盖广泛的节点范围. 与此同时, 中心性越高的节点对网络的整体影响也就越大. 通过优先处理这些节点, 能够更为有效地利用网络信息, 减少后续处理过程中的信息冗余、误差值堆积以及不确定性, 进而使得对剩余节点的重构变得更加快捷准确. 本文设

置了两个阈值 Λ, Π , 阈值与由不同 t 处的 $S_j(t)$ ($j \neq i$) 组成的字符串之间的归一化汉明距离有关^[8], 以识别 $\hat{t}_\alpha$ 处的基字符串和受基约束的一组字符串, 根据选取的基字符串中 01, 11 对的数量重新对重构节点进行排序, 按照顺序的降次来进行压缩感知重构. 如图 1 所示, 给定一组长度为 t_n 的时间序列, 重构网络中的任意节点 i 的相邻向量 X , 首先根据阈值 Π 找到满足条件的所有基串. 具体操作如下: 选择 $S_i(t)$ 连续两个时间片段中首次满足 00 或 01 的时间片段, 将 $S_i(t)$ 时间段对应的字符串 $S_j(t)$ 设置为第一个基本字符串 $S_j(\hat{t}_1)$, 然后依次找到满足 $H[S_j(\hat{t}_1), S_j(\hat{t}_2)] > \Pi$ 的第二个基串 $S_j(\hat{t}_2)$. 同理可以得到第三个基串 $S_j(\hat{t}_3)$ 满足 $H[S_j(\hat{t}_1), S_j(\hat{t}_3)] > \Pi$, $H[S_j(\hat{t}_2), S_j(\hat{t}_3)] > \Pi$. 重复这个过程, 直到将二进制时间序列矩阵全部枚举, 这就产生了一组基串 $S_j(\hat{t}_1), S_j(\hat{t}_2), S_j(\hat{t}_3), \dots, S_j(\hat{t}_m)$. 对于每个基串组合 $S_j(\hat{t}_\alpha)$, 给出了一组时间序列 t_k ($k = 1, 2, \dots, n$), 枚举时间序列选取满足 $H[S_j(\hat{t}_\alpha), S_j(t_v)] < \Lambda$ 的所有序列 $S_j(t_k)$ ^[8]. 然后, 基于 $\hat{t}_\alpha$ 和 t_k 计算两个平均值:

$$\begin{aligned} \langle \bar{\Gamma}_i(\hat{t}_\alpha + 1) \rangle &= \frac{S_i(\hat{t}_\alpha + 1) + \sum_{k=1}^l S_i(t_k + 1)}{l + 1}, \\ \langle \bar{\Gamma}_j(t_\alpha) \rangle &= \frac{S_j(\hat{t}_\alpha) + \sum_{k=1}^l S_j(t_k)}{l + 1}, j = 1, 2, \dots, i - 1, i + 1, \dots, N. \end{aligned} \quad (8)$$

两个平均量分别用于构建向量 Y 和矩阵 A . 对上述选取的所有基本字符串重复此过程, 从而达到满足重构形式 $Y = AX$.

3.3 节点平均状态转换

通过上述分析, 使用大数定律来处理节点时间状态转移矩阵, 概率 $P_i^\lambda(\hat{t}_\alpha)$ 的相关取值可以通过选取的测量矩阵的时间状态 $S_i(t + 1)$ 上的平均值来估计计算. 通过阈值 Π 判断并选取满足条件的基本字符串. 最终得到满足压缩感知矩阵形式的重构方程:

$$\begin{bmatrix} \langle \bar{\Gamma}_i(\hat{t}_1 + 1) \rangle \\ \langle \bar{\Gamma}_i(\hat{t}_2 + 1) \rangle \\ \vdots \\ \langle \bar{\Gamma}_i(\hat{t}_m + 1) \rangle \end{bmatrix} = \begin{bmatrix} \langle \bar{\Gamma}_1(\hat{t}_1) \rangle & \cdots & \langle \bar{\Gamma}_{i-1}(\hat{t}_1) \rangle & \langle \bar{\Gamma}_{i+1}(\hat{t}_1) \rangle & \cdots & \langle \bar{\Gamma}_N(\hat{t}_1) \rangle \\ \langle \bar{\Gamma}_1(\hat{t}_2) \rangle & & \langle \bar{\Gamma}_{i-1}(\hat{t}_2) \rangle & \langle \bar{\Gamma}_{i+1}(\hat{t}_2) \rangle & \cdots & \langle \bar{\Gamma}_N(\hat{t}_2) \rangle \\ \vdots & & \ddots & \vdots & & \\ \langle \bar{\Gamma}_1(\hat{t}_m) \rangle & \cdots & \langle \bar{\Gamma}_{i-1}(\hat{t}_m) \rangle & \langle \bar{\Gamma}_{i+1}(\hat{t}_m) \rangle & \cdots & \langle \bar{\Gamma}_N(\hat{t}_m) \rangle \end{bmatrix} \begin{bmatrix} a_{i1}/k_i \\ \vdots \\ a_{i,i-1}/k_i \\ a_{i,i+1}/k_i \\ \vdots \\ a_{iN}/k_i \end{bmatrix}, \quad (9)$$

其中 $\hat{t}_1, \hat{t}_2, \dots, \hat{t}_m$ 表示选取基本字符串相关的时间片, $\langle \cdot \rangle$ 表示满足条件阈值的平均值. 然后, 通过得到的节点时间状态平均序列值来表示向量 $Y_{m \times 1}$ 和矩阵 $A_{m \times (N-1)}$ 的值. 从少量的观测数据矩阵中实现了节点 i 的所有邻居的重构操作. 以此类推, 使用相同的处理方式对剩余的时间序列进行处理来重构所有节点的邻居节点向量, 最后对得到的邻居集合进行匹配来实现网络拓扑的重构. 在 SIS 动力学模型中有

$$\langle \bar{\Gamma}_i(\hat{t}_\alpha + 1) \rangle \cong \langle P_i^\lambda(\hat{t}_\alpha) \rangle = \frac{\lambda_i \sum a_{ij} \langle S_j(\hat{t}_\alpha) \rangle}{k_i}, \quad (10)$$

其中 k_i 表示网络中节点 i 的度数, $\langle (S_j(\hat{t}_\alpha)) \rangle$ 表示节点 j 在时刻 t_α 时满足阈值条件时邻居节点的平均状态矩阵. 然后, 使用阈值 Π 选择满足条件的基串建立相关方程, 以矩阵形式 $Y_{m \times 1} = A_{m \times (N-1)} \times X_{(N-1) \times 1}$ 表示, 其中 A 与方程 (7) 中的形式类似, 但 Y 和 X 的表达由下式给出:

$$\begin{aligned} Y &= [\langle \bar{\Gamma}_1(\hat{t}_1 + 1) \rangle, \langle \bar{\Gamma}_2(\hat{t}_2 + 1) \rangle, \dots, \langle \bar{\Gamma}_i(\hat{t}_m + 1) \rangle]^T, \\ X &= \left[\frac{\lambda_i}{k_i} a_{i1}, \dots, \frac{\lambda_i}{k_i} a_{i,i-1}, \frac{\lambda_i}{k_i} a_{i,i+1}, \dots, \frac{\lambda_i}{k_i} a_{iN} \right]^T. \end{aligned} \quad (11)$$

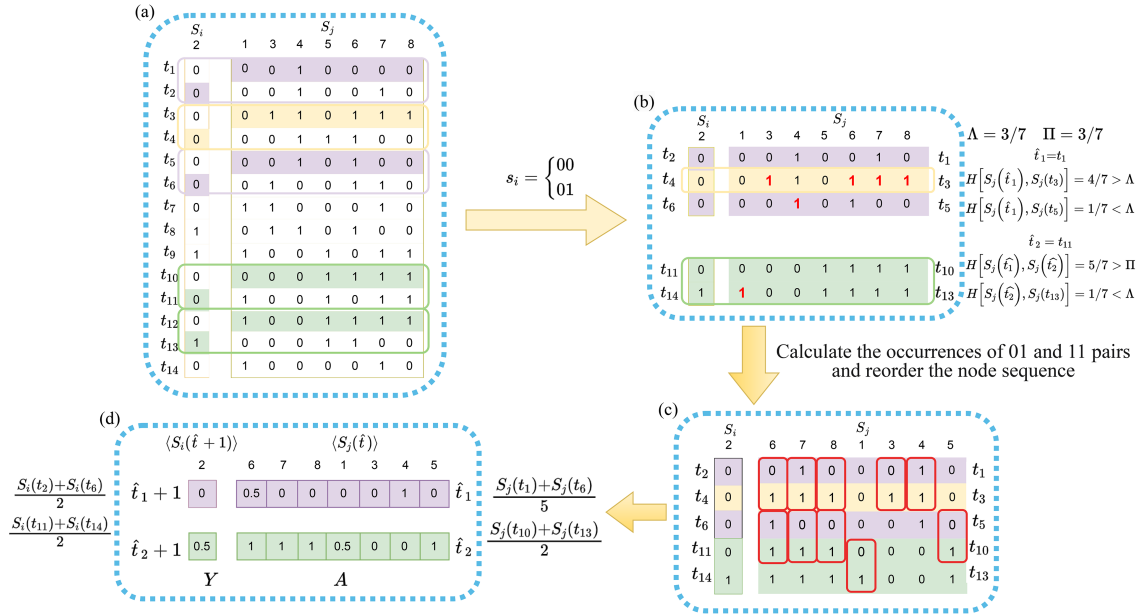


图 1 (网络版彩图) 考虑节点中心性从二进制时间序列建立 Y 和 A 的示意图。(a) 样本网络中包含 8 个节点以及每个节点 $t_1 \sim t_{14}$ 时刻数据的 14 个快照, 图中以节点 2 的时间序列 S_i 为例来重构其邻域节点, S_j 为邻域节点在不同时刻的字符串。其中只有 S_i ($i = 2$) 时间序列中的 00 和 01 对, 以及相应的 S_j 包含有关网络的有用信息, 用不同的颜色标记。(b) 由于 $S_i(t+1)$ 时刻的状态由 $S_j(t)$ 决定, 本文对 a 中选取的时间序列, 根据阈值参数 $\Lambda = 3/7, \Pi = 3/7$, 计算每对字符串 $S_j(t)$ 之间的归一化汉明距离, 选取满足条件 H (归一化汉明距离) $> \Pi$ 作为基本字符串, H (归一化汉明距离) $< \Lambda$ 作为相同字符串组, 计算基本字符串与相同字符串组之间其余字符串的汉明距离 $S_j(\hat{t}_{\alpha})$, 并以红色标记每个字符串差异。(c) 对选取满足条件的字符串组, 用红色标注出来, 即统计每个节点时间序列中两个连续的时间段 t_i, t_{i+1} 满足 01 或 11 时的数量, 然后按照其数量降序对节点顺序重新排序。(d) 计算其平均值 $\langle S_i(\hat{t}_{\alpha} + 1) \rangle, \langle S_j(\hat{t}_{\alpha}) \rangle$ 来分别表示矩阵 Y, A 。

Figure 1 (Color online) Schematic diagram of constructing Y and A from binary time series considering node centrality. (a) The sample network consists of 8 nodes, with 14 snapshots of data for each node at times $t_1 \sim t_{14}$. The diagram uses the time series S_i of node 2 to reconstruct its neighborhood nodes, with S_j representing the string of neighborhood nodes at different time points. Only the pairs 00 and 01 in S_i ($i = 2$) and the corresponding S_j containing relevant network information are highlighted in different colors. (b) Since the state of $S_i(t+1)$ is determined by $S_j(t)$, the normalized Hamming distance between each pair of strings $S_j(t)$ from the selected time series is calculated using the threshold parameters $\Lambda = 3/7$ and $\Pi = 3/7$. Pairs with a normalized Hamming distance H (Normalized Hamming Distance) $> \Pi$ are considered fundamental strings, while those with H (Normalized Hamming Distance) $< \Lambda$ are grouped as identical strings. The Hamming distance $S_j(\hat{t}_{\alpha})$ between the fundamental strings and the remaining strings in the identical string group is computed, with differences marked in red. (c) For the selected string groups, red highlights are used. Specifically, the number of consecutive time periods t_i, t_{i+1} in each node's time series, where the strings are "01" or "11", is counted. The nodes are then reordered in descending order of these counts. (d) The average values $\langle S_i(\hat{t}_{\alpha} + 1) \rangle$ and $\langle S_j(\hat{t}_{\alpha}) \rangle$ are computed to represent the matrices Y and A , respectively.

3.4 交替乘子法优化压缩感知矩阵

应用交替乘子算法^[36]优化矩阵, 解决上述问题。交替乘子法是一种将对偶上升的可分解性与乘法器优越的收敛性相结合的算法。它可以解决一个线性约束的最小优化问题, 该问题被明确地描述为

$$\begin{aligned} \min_{x,z} f(x) + \varphi(z) \\ \text{s.t. } Ax + Bz = c, \end{aligned} \quad (12)$$

其中 $x \in \mathbb{R}^n, z \in \mathbb{R}^m$, 等式中 $A \in \mathbb{R}^{p \times n}, B \in \mathbb{R}^{p \times m}, c \in \mathbb{R}^p, f(x)$ 和 $\varphi(z)$ 是可分离的凸函数。

通过对偶变量 η , 可以将式 (12) 中的目标函数与约束耦合在一起。同时, 为了使函数更具凸性以及增加其收敛速度, 增加了一个新的二次惩罚项 $\frac{\rho}{2} \|Ax + Bz - c\|_2^2$, 得到的增广拉格朗日 (Lagrange)

算子可以表示为

$$L_p(x, z, u) = f(x) + \varphi(z) + \eta^T(Ax + Bz - c) + \frac{\rho}{2} \|Ax + Bz - c\|_2^2. \quad (13)$$

一般将对偶变量定义为 $\gamma = \frac{\eta}{\rho}$, 可应用于增广拉格朗日, 其缩放形式如下:

$$\begin{aligned} L_p(x, z, u) &= f(x) + \varphi(z) + \rho\gamma^T(Ax + Bz - c) + \frac{\rho}{2} \|Ax + Bz - c + \gamma - \gamma\|_2^2 \\ &= f(x) + \varphi(z) + \rho\gamma^T(Ax + Bz - c) + \frac{\rho}{2} \{ \|Ax + Bz - c + \gamma\|_2^2 + \|\gamma\|_2^2 - 2\gamma^T(Ax + Bz - c + \gamma) \} \\ &= f(x) + \varphi(z) + \frac{\rho}{2} \|Ax + Bz - c + \gamma\|_2^2 + \frac{\rho}{2} \|\gamma\|_2^2 - \rho\|\gamma\|_2^2 \\ &= f(x) + \varphi(z) + \frac{\rho}{2} \|Ax + Bz - c + \gamma\|_2^2 - \frac{\rho}{2} \|\gamma\|_2^2. \end{aligned} \quad (14)$$

基于上述分析, 重构算法中的交替乘子法在不耦合变量的情况下通过将大规模重构问题分解为若干子问题来简化计算过程. 具体来说, 对于重构时的大问题等式 (2) 可以转化为一个带约束的优化问题: $\min_x f(x) + g(x)$, 其中 $f(x)$ 为观测误差的度量函数, $g(x)$ 为稀疏性的惩罚项, 在将其拆分为多个子问题时, 通常涉及引入辅助变量, 等式 (12) 中 z 为引入的辅助变量代表 x 的稀疏性, 同时使用等式 (13) 更新拉格朗日乘子. 接下来, 通过等式 (16) 对子问题进行迭代更新操作, 每次更新时, 仅对其中一个变量进行更新操作, 固定其余参数值, 重复执行这些步骤, 最终能够逐步逼近原始优化问题的解. 对于惩罚系数 ρ , 值过小会导致求解精度低, 值过大则不能满足稀疏性要求. 为了在准确性和稀疏性之间权衡, 参考了文献 [37] 中相关系数的取值, ρ 被设置为 1. 对于迭代的上限 $k_{\max}$, 可以在期望的结果精度没有明显变化的情况下确定. 本文可以更进一步将上述问题推广到更广义的优化问题, 将经过压缩感知操作的节点时间序列矩阵 Φ , 以及需要重构操作的目标节点矩阵 Y 分别带入到下式中, 所以就转化成了带正则的全局一致性优化问题, 具体问题如下:

$$\begin{aligned} \min_x \quad & \sum_{i=1}^N f_i(x_i) + \varphi(z) \\ \text{s.t.} \quad & x_i - z = 0, \quad i = 0, 1, \dots, N. \end{aligned} \quad (15)$$

其交替乘子法迭代的更新形式如下:

$$\begin{aligned} x_i^{k+1} &= \arg\min_x \left(f_i(x_i) + y_i^{T^k} (x_i - z^k) + \frac{\rho}{2} \|x_i - z^k\|_2^2 \right), \\ z^{k+1} &= \arg\min_z \left(\varphi(z) + \sum_{i=1}^N \left(-y_i^{T^k} + \frac{\rho}{2} \|x_i^{k+1} - z\|_2^2 \right) \right), \\ y_i^{k+1} &= y_i^k + \rho(x_i^{k+1} - z_i^{k+1}). \end{aligned} \quad (16)$$

表示为全局一致性优化问题:

$$\begin{aligned} \min_x \quad & \sum_{i=1}^N f_i(x_i) + \varphi(z) \\ \text{s.t.} \quad & x_i - z = 0, \quad i = 0, 1, \dots, N, \end{aligned} \quad (17)$$

其中, $f_i(x_i)$ 表示划分到每一个节点上的目标函数, x_i 表示局部模型参数, z 表示全局一致性变量. 把整个数据集划分到每一节点, 通过迭代更新, 最终收敛到一个一致的全局解. 具体形式如下:

$$\min_i \frac{1}{2} \|Ax - b\|_2^2 + \lambda \|x\|_1, \quad (18)$$

其中, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$. 更换成全局一致性优化问题为

$$\begin{aligned} \min_{x_i} \quad & \sum_{i=1}^N (\|Ax_i - b\|^2 + \lambda \|z\|_1) \\ \text{s.t.} \quad & x_i - z = 0, \quad i = 0, 1, \dots, N. \end{aligned} \quad (19)$$

全局优化问题的迭代更新形式如下:

$$\begin{aligned} x_i^{k+1} &= \underset{x_i}{\operatorname{argmin}} \left(\frac{1}{2} \|A_i x_i - b_i\|^2 + y_i^{T^k} (x_i - z^k) + \frac{\rho}{2} \|x_i - z^k\|_2^2 \right), \\ z^{k+1} &= \underset{z}{\operatorname{argmin}} \left(\lambda \|z\|_1 + \sum_{i=1}^N (-y_i^{T^k} z + \frac{\rho}{2} \|x_i^{k+1} - z\|_2^2) \right), \\ y_i^{k+1} &= y_i^k + \rho(x_i^{k+1} - z_i^{k+1}). \end{aligned} \quad (20)$$

对于 x_i^{k+1} 的更新, 右边括号内的函数 $\varphi = \frac{1}{2} \|A_i x_i - b_i\|^2 + y_i^{T^k} (x_i - z^k) + \frac{\rho}{2} \|x_i - z^k\|_2^2$ 对 x_i 求导可以得到: 首先 $\|A_i x_i - b_i\|^2 = (A_i x_i - b_i)^T (A_i x_i - b_i)$, 所以 $\frac{\partial (\|A_i x_i - b_i\|^2)}{\partial (x_i)} = A_i^T (A_i x_i - b_i)$,

$$\frac{\partial \varphi}{\partial x_i} = A_i^T (A_i x_i - b_i) + y_i^{T^k} + \rho(x_i - z^k), \quad (21)$$

令上式为 0 可以得到

$$A_i^T (A_i x_i - b_i) + y_i^k + \rho(x_i - z^k) = 0, \quad (22)$$

即可以得到

$$x_i^{k+1} = (A_i^T A_i + \rho I)^{-1} (A_i^T b_i + \rho z^k - y_i^k). \quad (23)$$

对于 z 的更新, 同理 x_i 可以计算为

$$z^{k+1} = \arg \min_z \left(\lambda \|z\|_1 + \frac{\rho N}{2} \left\| z - \frac{1}{\rho N} \sum_{i=1}^N (\rho x_i^{k+1} + y_i^k) \right\|_2^2 \right). \quad (24)$$

因此 z 的更新可以使用软阈值算子的方法, 其中 ζ 为阈值参数, 确保重构信号的稀疏性, 如下所示:

$$z^{k+1} = S_{\frac{\lambda}{\rho N}} \left(\frac{1}{\rho N} \sum_{i=1}^N (\rho x_i^{k+1} + y_i^k) \right), \quad (25)$$

$$S_{\zeta}(a) = \begin{cases} a - \zeta, & a > \zeta, \\ 0, & |a| \leq \zeta, \\ a + \zeta, & a < -\zeta. \end{cases} \quad (26)$$

文献 [38] 提出的停止准则是基于原始变量和对偶变量的残差来确定的, 即为原始残差 (r) 和对偶残差 (s):

$$r \leq \varepsilon_{\text{pri}}, \quad s \leq \varepsilon_{\text{dual}}, \quad (27)$$

其中 $r = \|Ax + Bz - c\|$, 这通常是衡量 $Ax + Bz - c$ 与零向量之间差异的量, 这个值越趋近于零, 其效果越好. 而 $s = \|\nabla f(x) + A^T y\|$, 相对而言, $\varepsilon_{\text{pri}} > 0$ 和 $\varepsilon_{\text{dual}} > 0$ 可以理解为原始可行性条件和对偶可行性条件之间的差别. 采用绝对标准和相对标准 [39], 这两个公差可表示为

$$\begin{aligned} \varepsilon_{\text{pri}} &= \sqrt{p} \varepsilon_{\text{abs}} + \varepsilon_{\text{rel}} \max(\|Ax^k\|_2, \|Bz_k\|_2, \|c\|_2), \\ \varepsilon_{\text{dual}} &= \sqrt{w} \varepsilon_{\text{abs}} + \varepsilon_{\text{rel}} \|A^T u^k\|_2, \end{aligned} \quad (28)$$

其中, ε_{abs} 为绝对误差容忍度, ε_{rel} 为相对误差容忍度, 一般而言, 两者都可以设为一个较小的正数, 例如 10^{-3} , 意味着算法需要更紧密地满足原始条件和对偶条件, 从而得到更接近全局最优解的结果.

基于上述分析, 在重构算法中, 引入交替乘子法的重要作用就是为了将大规模重构问题分解为几个可以在没有耦合变量情况下的子问题. 到目前为止, 已经实现了节点 i 的网络重构任务的分解处理操作. 在统计的网络中每个节点的时间状态矩阵基础上, 运用交替乘子法来分解优化传统压缩感知中求解 L_1 正则化最小值时所需的测量矩阵, 以此降低数据的维数, 进而实现将原问题划分成几个较小的子问题. 然后, 在这些子问题之间进行计算. 为了更进一步提升网络重构时的计算精度, 在原模型的基础上增添二进制约束, 并采用交替乘子法来求解每个子问题. 重构过程的具体流程如图 2 所示.

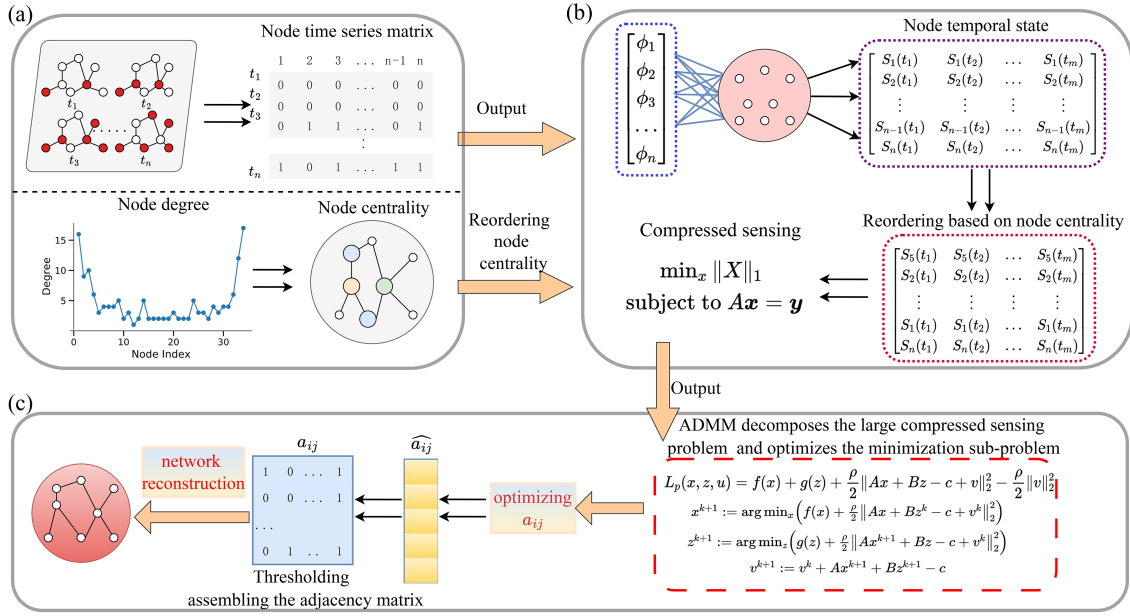


图 2 (网络版彩图) NCSA 算法重构网络流程. (a) 获取节点状态, 统计网络中节点的时间状态矩阵以及计算网络中节点的度, 便于对重构节点顺序重新排序; (b) 节点中心性压缩感知, 使用节点中心性处理矩阵, 将矩阵重新排序, 运用压缩感知对排序后的节点状态矩阵进行操作处理; (c) 交替乘子法优化重构矩阵, 在 (b) 压缩感知时使用交替乘子法对 L_1 正则化最小值优化问题进行问题分解和局部重构优化, 最终得到网络的邻接矩阵, 从而达到重构网络拓扑的目的.

Figure 2 (Color online) Network reconstruction process using the NCSA algorithm. (a) Obtain node states, calculate the temporal state matrix of nodes in the network, and compute the degree of each node, facilitating the reordering of nodes for reconstruction. (b) Node centrality-based compressed sensing: process the matrix using node centrality, reorder the matrix, and apply compressed sensing to the reordered node state matrix for further operations. (c) Alternating direction method of multipliers (ADMM) for optimizing the reconstruction matrix: during the compressed sensing in (b), use ADMM to decompose the problem of L_1 -regularized minimization and perform local reconstruction optimization. This ultimately yields the adjacency matrix of the network, achieving the goal of reconstructing the network topology.

3.5 算法复杂度分析

本文用到的基本伪代码在算法 1 和 2 中有详细步骤介绍. 算法 1 包含 2 个主要循环: 外层的最大迭代次数 $k_{\max}$ 、内层的交替乘子法更新步骤以及软阈值算子优化矩阵. 其中每次迭代过程可以简化为以下的操作步骤: 首先, 假设状态矩阵 A 的大小为 $N \times M$, N 为网络节点数, M 为测量矩阵的长度, 对于每个节点 i , 依次执行 N 次 x_i, y_i, z_i 的更新, 总时间复杂度可以表示为 $O(3N)$; 其次, 执行一次停止准则的判断, 时间复杂度为 $O(1)$; 最后, 执行一次软阈值算子的优化更新, 时间复杂度 $O(N \times M)$, 在算法 1 中的总时间复杂度可以表示为 $O(k_{\max}(3N + MN))$, 所以简化后的时间复杂度可近似表示为 $O(k_{\max}MN)$. 算法 2 中最开始对矩阵 A 行数 N 进行遍历, 但算法中主要的时间消耗在 5~18 行中对状态矩阵 A 的行数 N 和列数 M 进行遍历操作, 那么该时间复杂度就可以表示为 $O(MN^2)$, 两个算法的总复杂度为 $O(k_{\max}MN + MN^2)$. 使用压缩感知处理状态矩阵, 同时因为引入的交替乘子法对传统压缩感知中 L_1 正则化最小优化操作是迭代运算操作, 其中迭代次数 $k_{\max}$ 的取值相对于网络大小 N 而言值太小, 所以整个 NCSA 算法的时间复杂度可以表示为 $O(MN^2)$, 相比于代表性压缩感知方法的时间复杂度 ($O(MN^3)$) [8], 得到了显著的提高. 从上述算法复杂度分析可以得到, 本文所提出的 NCSA 算法可以用于规模较大的网络. 所以, 对于重构网络的规模以及重构的时间复杂度这两个重要指标而言, NCSA 的优势更为明显.

算法 1 交替乘子优化算法.**Require:** $\rho = 1$, 迭代上限 $k_{\max} = 100$, 正则化参数 $\lambda = 0.01 \times \sigma \times \|A^T \times \Phi\|_{\infty}$, $\sigma = 0.05$, 压缩感知矩阵 X ;

```

1: for  $i < k_{\max}$  do
2:   for  $j < X_{\text{row-num}}$  do
3:     根据等式 (20) 更新  $x_i^{k+1} = \operatorname{argmin}_{x_i} (\frac{1}{2} \|A_i x_i - b_i\|^2 + y_i^{T^k} (x_i - z^k) + \frac{\rho}{2} \|x_i - z^k\|_2^2)$ ;
4:     更新  $y_i^{k+1} = y_i^k + \rho(x_i^{k+1} - z_i^{k+1})$ ;
5:     更新  $z^{k+1} = \operatorname{argmin}_z (\lambda \|z\|_1 + \sum_{i=1}^N (-y_i^{T^k} z + \frac{\rho}{2} \|x_i^{k+1} - z\|_2^2))$ ;
6:   end for
7:   if 满足等式 (27) 的停止准则 then
8:     停止迭代操作;
9:   end if
10:  根据软阈值算子 (26) 优化矩阵;
11: end for
12: return  $\bar{X}$  (网络重构后的邻接矩阵).
```

算法 2 NCSA 算法.**Require:** 选取节点状态的参数阈值 Λ, Π , 惩罚系数 $A \in (N \times M)$, 状态矩阵、节点邻居矩阵 Φ ;

```

1: while  $i < N$  do
2:   从矩阵  $A$  中选取节点状态  $\omega_i$ ;
3:   建立  $\text{avgmatrix}_1 = [], \text{avgmatrix}_2 = []$  存储平均向量结果;
4:   提取第一个基向量  $N_{s1}, n = \text{len}(N_{s1})$ ;
5:   while  $i < M$  do
6:     if  $\omega_i == 0$  和  $(\omega_i == 0 \text{ 或 } \omega_i == 1)$  then
7:       提取其他基向量  $N_{s2}, N_{s2}, \dots, N_{sn}$ ;
8:     end if
9:     for  $i < n$  do
10:      计算矩阵  $A$  中每行与选取基向量的汉明距离  $H[\Delta]$ ;
11:      if  $H[\Delta] > \Pi, H[\Delta] < \Lambda$  then
12:        存储该节点向量  $\tau_1, \tau_2, \dots, \tau_n$ ;
13:      end if
14:      根据节点中心性对存储节点向量重新排序  $\tau_5, \tau_2, \tau_1, \dots, \tau_n$ ;
15:      对存储向量  $\tau_1, \tau_2, \dots, \tau_n$  进行压缩感知;
16:      使用算法 1 对  $x_i, y_i^{k+1}, z_i^{k+1}$  进行迭代更新同时求解最优值;
17:    end for
18:  end while
19: end while
20: return  $\hat{X}$  (网络重构后的邻接矩阵).
```

4 实验

为了验证提出 NCSA 算法的性能, 进行了一系列仿真实验. 首先, 介绍实验中的软件、硬件需求, 使用 Windows 11, NVIDIA 和 16 GB RAM, 算法用 Python 3.6 在 PyCharm 上进行编译, 相关依赖库有 Numpy, Pandas, Matplotlib 等. 在该方法中, 每个划分重构子问题是相互独立的. 本文采用 7 个真实网络数据集, 如表 1 所示. 本文以受试者工作特性曲线下面积 (area under the receiver operating characteristic curve, AUROC)、程序运行时间 (running time) 为标准指标评估方法的重构性能. 众所周知, AUROC 越大, 网络重构性能越好, 程序的运行时间越短, 说明该算法的效率越高. 为方便感兴趣的读者使用, 本文所用到的数据与代码可以在网站¹⁾找到.

通过将标准 (即真邻接矩阵) 与预测结果进行比较, 每个节点的结果可以被分类为真阳性 (true

1) github.com/mrking111/nca.

表 1 真实社交网络的描述.

Table 1 Description of real social networks.

Network	N	L	Description
Karate	34	78	The friendship social network of a karate club
Dolphins	62	159	The social network of a dolphin population
Cat-brain	65	1100	Edges represent fiber bundles connecting one vertex to another
Polbooks	105	441	A network of books about American politics
Football	115	613	The network of American football games from the 2000 season
Ca-netscience	379	914	Co-authorship network of network theorists and experimental scientists
Soc-wiki-vote	889	2900	All Wikipedia voting data from its inception to January 2008

positive, TP)、假阳性 (false positive, FP)、真阴性 (true negative, TN) 或假阴性 (false negative, FN). 从本文的实验中可以得到提出的 NCSA 方法准确度相比其他方法有了显著的提高, 同时相比传统的压缩感知方法, 该方法可以处理更大类型的网络拓扑结构的重构操作. 本文计算出该方法的精度为

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}.$$

指标 “Accuracy” 表示网络重构的结果中真阳性和真阴性两者之和与重构的最终结果比值, 该比值用以表示重构的正确率, 其值越高表明重构的效率越好. 受试者工作特征曲线下面积 (AUROC) 的取值是通过量化 ROC 曲线下的面积得出, 其中 ROC 曲线是以假阳性率 ($\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}$) 为横坐标, 真阳性率 ($\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}$) 为纵坐标绘制的曲线, 它能够综合评估模型中网络连接关系预测的准确性, 其值处于 0~1 之间, 越接近于 1 表示重构效果越好.

4.1 性能对比分析

本小节主要是对提出的 NCSA 算法的性能对比实验以及分析. 图 3 为无噪声情况下网络重构任务的执行效率, 图 3(a) 分析了所提出的基于节点中心性排序和交替乘法法的网络重构算法与代表性的压缩感知方法 (NCSA 和 CST) 下的运行时间消耗, 图 3(b) 表示将代表性的压缩感知方法运行时间归一化后 NCSA 的运行时间百分比. 实验结果表明, 在相同的计算机资源下, 提出的 NCSA 算法的时间消耗少了近一半, 其原因是引入交替乘法法会将原本压缩感知时的大问题分解为多个小问题进行处理, 在 L_1 正则化时为迭代运算操作, 而迭代次数 $k_{\max}$ 相比于网络中节点个数而言的取值可以忽略不计, 所以能够极大地减少代表性算法的时间复杂度, 从而使得运行时间得到减少. 将本文提出的 NCSA 与代表性的 CST 方法相比, 来说明该方法对效率的提升.

我们继续讨论参数对重构性能的影响, 并以 100 个节点的 WS 小世界网络为例, 来讨论参数变化对网络重构的影响. 讨论了其中的两个主要参数 Λ , Π (Λ 表示在不同时间片段 t 时刻内筛选其他节点时间状态 $S_j(t)$ ($j \neq i$) 的标准化汉明距离的阈值, 选取满足基本约束的一组字符串. Π 阈值用于识别满足条件的其他基本字符串, 从而选出多组满足约束的字符串组). 为了直观地分析一个超参数, 实验时将其他超参数固定. 一般来说, 在保证高精度的情况下, 设置迭代次数 (iteration = 100)、迭代阈值 (tol = 1E-4)、惩罚系数 ($\beta = 1$). 图 4 展示了 NCSA 算法随 Π 变化的重构精度, 可以看出, Π 取值的变化对 NCSA 的性能影响较大. 当达到最优值后, 随着 Π 值的增加, 重构精度变低, 其原因很可能是超参数 Π 控制网络结构的稀疏度. Π 的值增加使得选取节点中的时间片段减少, 从整体中来看增加了非零元素的数量, 从而导致重构精度的减小. 为了实现 NCSA 的高准确率, Π 的值应选取在 0.2 左右. 另一条曲线为 NCSA 随 Λ 阈值变化的重构精度, 随着 Λ 的增大, 网络重构精度逐渐减小, 原因是超参数 Λ 控制网络结构中基字符串数, 当值增加后, 基字符串选取数量会随之减少, 使其交替乘法法优化更加困难. 随着 Λ 值的增加对网络重构精度有较大的影响, 通常, Λ 的取值在 0.15 左右. 由于不同网

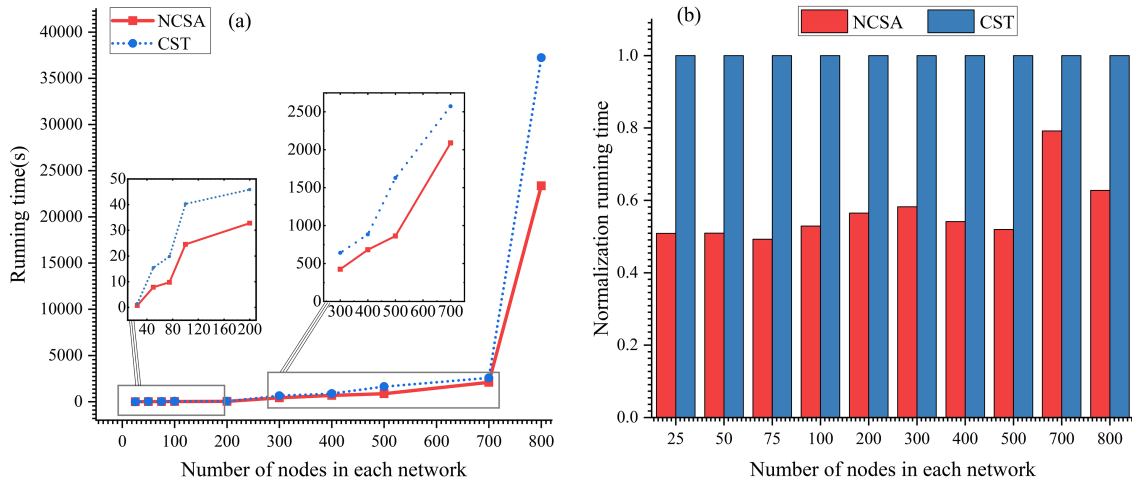


图 3 (网络版彩图) NCSA 和 CST, 即提出的基于节点中心性排序和交替乘子法的网络重构算法与代表性的压缩感知框架下的运行时间比较. (a) 两种方法在不同网络大小运行时间的折线对比图; (b) 以 CST 方法的运行时间归一化为参照标准, NCSA 算法的运行时间柱状图.

Figure 3 (Color online) Comparison of the running time between the proposed node centrality sorting and alternating direction method of multipliers (NCSA) network reconstruction algorithm and the representative compressed sensing framework (CST). (a) Line chart comparing the running time of both methods for different network sizes; (b) bar chart showing the running time of the NCSA algorithm with the CST method's running time normalized as the reference standard.

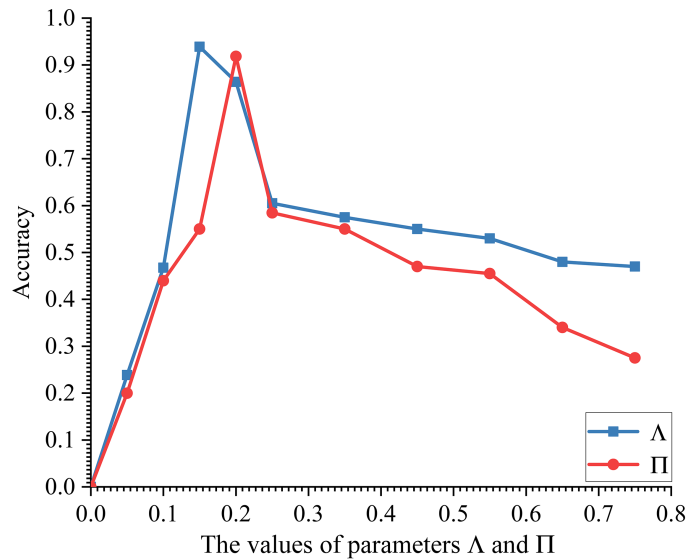


图 4 (网络版彩图) 在 100 节点的 WS 网络上, 重构精度随 NCSA 算法中参数 Π , Λ 的变化图.

Figure 4 (Color online) Reconstruction accuracy as a function of parameters Π and Λ in the NCSA algorithm on a 100-node WS network.

网络的稀疏度不同, 所以在进行最初观测值筛选时就可能导致每个网络用于算法实现的数据稀疏度不相同, 就需要设置不同的阈值 Λ, Π 的取值去将观测数据调整为满足条件的稀疏度, 来达到较好的实验结果. 所以对于 Λ, Π 的取值可以建立在上述关于 WS 网络中相关取值的基础上来进行相关阈值的选取操作.

为了进一步验证算法性能, 本文设置了在 4 个人工网络 (如表 2 所示, 网络平均度 $\langle k \rangle \approx 3$, 网络边数分别为 164, 176, 1867, 1955) 和 7 个真实网络上的对比实验. 从表 2 中分析得到 A-CST 其效果相比于纯粹的 CST 的效果更差是因为在每次迭代过程中, 都需要进行多个子问题的求解和变量更新,

表 2 NCSA 与其他方法的 AUROC 比较. 粗体表示最优结果.

Table 2 Comparison of AUROC between NCSA and other methods. The best results are in bold.

Dataset	N	CST	C-CST	A-CST	NCSA
50Nodes-WS	50	0.9701	0.9818	0.7003	0.9801
50Nodes-BA	50	0.9379	0.9489	0.7347	0.9796
600Nodes-WS	600	0.8567	0.8283	0.5614	0.8938
600Nodes-BA	600	0.8535	0.8283	0.7424	0.8879
Karate	34	0.9871	0.9873	0.6603	0.9935
Dolphins	62	0.9874	0.9832	0.7704	0.9874
Cat-brain	65	0.9705	0.9835	0.7958	0.9828
Polbooks	105	0.9410	0.9410	0.9297	0.9705
Football	115	0.9714	0.9721	0.7887	0.9787
Ca-netscience	379	0.9590	0.9655	0.9496	0.9814
Soc-wiki-vote	889	0.9456	0.9449	0.7816	0.9783

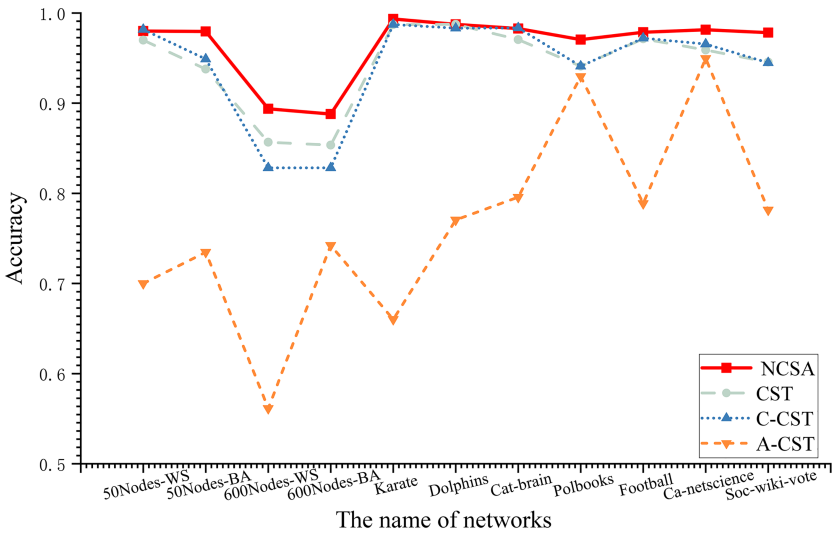


图 5 (网络版彩图) 消融实验图. 比较了提出的 NCSA 算法、代表性的压缩感知方法 (CST)、节点中心性的压缩感知方法 (C-CST) 以及交替乘法结合压缩感知方法 (A-CST) 这 4 种方法在 4 个人工网络和 7 个真实网络上的重构实验性能对比.

Figure 5 (Color online) The ablation study compares the performance of four methods (NCSA, a representative compressed sensing method (CST), a compressed sensing method based on node centrality (C-CST), and an alternating direction method of multipliers combined with compressed sensing (A-CST)) in reconstruction experiments on four synthetic networks and seven real-world networks.

不可避免地会导致计算误差, 使得误差可能在后续的迭代过程中累积. 而引入节点中心性排序, 效果得到显著提升, 是因为优先对重要节点进行处理, 减少其余节点带来的噪声干扰, 降低了进行交替乘子时累积的误差, 在一定程度上降低重构算法的计算复杂度, 同时也相当于为重构操作提供额外的约束条件, 减少了不确定性. 从表 2 可以得到本文提出的算法无论在数据量较小的网络还是大网络上, 其重构效果大多数情况下都优于代表性压缩感知算法的效果, 且无论网络的大小都具有较高的重构精度, 其中 N_M 表示实验数据长度与网络大小的比值, 其可视化如图 5 所示.

4.2 鲁棒性分析

本小节讨论 NCSA 算法在网络重构中的鲁棒性, 即节点状态时间长度的选取以及抗噪声的相关

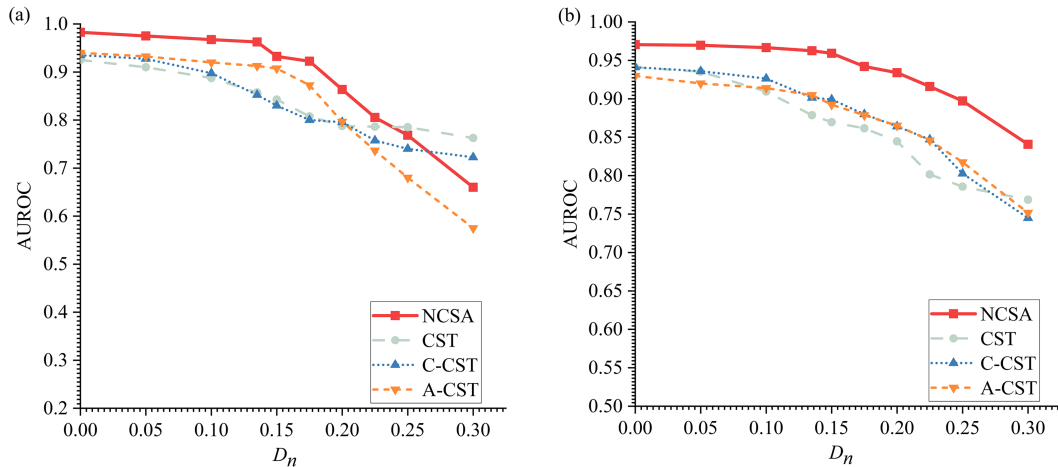


图 6 (网络版彩图) 验证引入噪声后所提算法对重构效果的鲁棒性. 横坐标为网络中引入的噪声与网络大小的比值, 纵坐标为网络重构指标 AUROC 值. 在 (a) 100 个节点的 WS 人工网络和 (b) Polbooks 的真实网络中, 算法对噪声的干扰都具有更好的鲁棒性.

Figure 6 (Color online) Assessment of the robustness of the proposed algorithm to reconstruction performance after introducing noise. The horizontal axis represents the noise-to-network size ratio, while the vertical axis shows the network reconstruction metric, AUROC value. In (a) the WS synthetic network with 100 nodes and (b) the real-world network Polbooks, the algorithm demonstrates better robustness against noise interference.

实验. 为了进一步研究方法对观测噪声的抗干扰能力, 使用人工网络 (100 节点的 WS) 和真实网络 (Polbooks) 为重构目标网络进行对比实验, 在观测向量中加入分布为 $\mathcal{N}(0, 1)$ 的随机高斯白噪声. 在 $RD = 0.4$ 和不同信噪比情况下进行网络重构实验. 图 6 为引入高斯噪声后, 该方法对重构效果的影响, 图中 D_n 表示数据中的噪声比, 纵坐标为重构精度. 可以得出, 当噪声值小于 0.1 时, NCSA 算法的重构精度趋于稳定, 但随着噪声值的增加, 精度逐渐降低, 最后趋于稳定, 但真实网络的抗噪效果相比更好, 原因可能是网络结构矩阵相比人工网络更加稀疏, 其精度最低趋于 0.85 左右, 这些网络结构本身具有一定的鲁棒性, 虽然高斯白噪声的引入可能会使个别节点的度发生小幅度变化, 但对其进行重构时可以利用其他未受噪声干扰的节点信息来进行重构网络结构, 同时压缩感知在一定条件下可以有效处理噪声干扰. 实验结果表明, NCSA 算法对噪声干扰具有较强的鲁棒性.

虽然与需要重构网络的规模相比, 基字符串的数量相对较少, 但实际中需要一组不同时间片段的基字符串来制定重构的数学框架. 所以进一步研究了选取的时间序列长度对重构精度的影响, 即所需观测参数的大小与重构精度之间的关系. 图 7 展示了在人工网络 (100 节点的 WS) 和真实网络 (Polbooks) 上时间序列相对长度 Len_t 对重构精度的影响, Len_t 代表时间序列从传播过程开始的总长度除以网络大小, 纵坐标为重构精度. 结果表明, 即使 Len_t 值较小, 大多数链路也已经可以被识别, 这反映在图 7 所示的 AUROC 值上. 当时间序列相对长度 Len_t 低于 0.1 时, 网络的重构效果不是很理想, 原因是选取的观测数据太少, 不能很好地包含整个网络的特征状况, 进而导致精度偏低. 而随着时间序列相对长度 Len_t 值的增加, 在实验结果中可以看见观测数据选取在 0.6 ~ 0.8 时, 其精度可以趋近于 90% 以上. 本文提出的方法在较小规模的数据量上进行重构操作仍具有较高的准确性和效率, 鲁棒性相对较高. 随着时间序列长度的增加, 重构精度也相应提升, 最终趋于稳定, 且稳定时的精度相对较高, 其原因是随着时间序列长度增加, 重构所需的节点观测数据更加精确完整, 从而使得其精度得到提高.

5 结论

本文提出了一种新的网络重构算法 (NCSA) (该算法能够对大规模网络进行有效重构), 并对其进

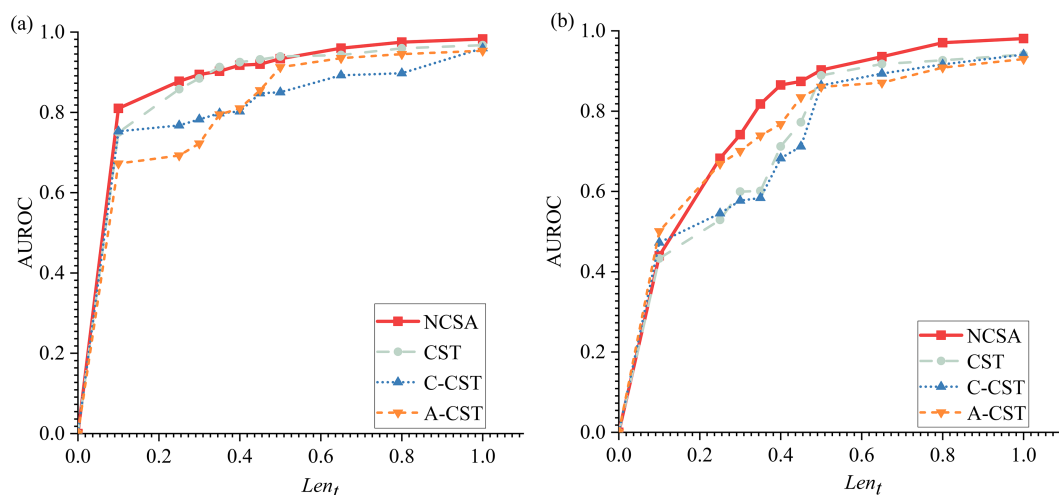


图 7 (网络版彩图) 对比分析算法时间长度对重构效果的影响. 横坐标为节点时间长度变化值, 纵坐标为网络重构指标 AUROC 值. 在 (a) 100 个节点的 WS 人工网络和 (b) Polbooks 的真实网络中, 都可以通过较少节点时间序列信息对网络进行重构操作.

Figure 7 (Color online) Comparative analysis of the impact of algorithm runtime on reconstruction performance. The horizontal axis represents the variation in node time length, while the vertical axis shows the network reconstruction metric, AUROC value. In (a) the WS synthetic network with 100 nodes and (b) the real-world network Polbooks, network reconstruction can be performed with a minimal amount of node time series information.

行了相关的实验验证和分析. NCSA 算法综合考虑了大规模网络的相关特征, 包括网络节点中心性对网络拓扑重构的影响、压缩感知对稀疏矩阵处理的优势以及交替乘法对大规模优化问题的应用. 本文在 7 个真实网络和 2 个人工网络上进行了多组实验综合对比分析. 实验结果表明 NCSA 算法在重构精度、抗噪声鲁棒性和重构运行时间等方面表现出优于代表性压缩感知重构方法的性能, 验证了该算法的优势. 此外, 该算法具有较低的时间复杂度, 可以适用于处理大规模稀疏网络, 因此在实际运用中具备良好的可扩展性和效率.

本文的研究虽只涉及无权无向网络的重构, 但所提出的算法可以扩展到具有不同类型约束的加权网络的重构中. 本文的算法取得了较好的效果, 但仍然值得进一步探索. 例如, 如何完成对其他类型网络的重构任务, 即如何运用其他网络的特征, 如时效网络或者多层网络的特征, 将网络重构问题也分解为多个子问题进行分布式求解, 同时确保全局的重构性能, 是未来的研究方向.

致谢 本文的数值计算得到了西南石油大学高性能计算中心的计算支持和帮助.

参考文献

- 1 Gao C, Zhu J, Zhang F, et al. A novel representation learning for dynamic graphs based on graph convolutional networks. *IEEE Trans Cybern*, 2022, 53: 3599–3612
- 2 Zhao J C, Lu J A, Wu X Q. Pinning control of general complex dynamical networks with optimization. *Sci China Inf Sci*, 2010, 53: 813–822
- 3 Yu W, DeLellis P, Chen G, et al. Distributed adaptive control of synchronization in complex networks. *IEEE Trans Automat Contr*, 2012, 57: 2153–2158
- 4 Wang J, Feng J, Xu C, et al. The synchronization of instantaneously coupled harmonic oscillators using sampled data with measurement noise. *Automatica*, 2016, 66: 155–162
- 5 Liu X, Cao J, Yu W, et al. Nonsmooth finite-time synchronization of switched coupled neural networks. *IEEE Trans Cybern*, 2015, 46: 2360–2371
- 6 Liu H, Lu J A, Lü J, et al. Structure identification of uncertain general complex dynamical networks with time delay. *Automatica*, 2009, 45: 1799–1807

- 7 Wu X, Zhao X, Lu J, et al. Identifying topologies of complex dynamical networks with stochastic perturbations. *IEEE Trans Control Netw Syst*, 2015, 3: 379–389
- 8 Shen Z, Wang W X, Fan Y, et al. Reconstructing propagation networks with natural diversity and identifying hidden sources. *Nat Commun*, 2014, 5: 4323
- 9 Caldarelli G, Chessa A, Pammolli F, et al. Reconstructing a credit network. *Nat Phys*, 2013, 9: 125–126
- 10 Levnajić Z, Pikovsky A. Network reconstruction from random phase resetting. *Phys Rev Lett*, 2011, 107: 034101
- 11 Hempel S, Koseska A, Kurths J, et al. Inner composition alignment for inferring directed networks from short time series. *Phys Rev Lett*, 2011, 107: 054101
- 12 Mei G, Wu X, Wang Y, et al. Compressive-sensing-based structure identification for multilayer networks. *IEEE Trans Cybern*, 2017, 48: 754–764
- 13 Sachs K, Perez O, Pe'er D, et al. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 2005, 308: 523–529
- 14 Xu J, Zheng S, Cai G L. Topology identification of weighted complex dynamical networks with non-delayed and time-varying delayed coupling. *Chin J Phys*, 2010, 48: 482–492
- 15 Wu X, Zhou C, Chen G, et al. Detecting the topologies of complex networks with stochastic perturbations. *Chaos*, 2011, 21: 043129
- 16 Liu J, Mei G F, Wu X Q, et al. Robust reconstruction of continuously time-varying topologies of weighted networks. *IEEE Trans Circ Syst Regular Papers*, 2018, 65: 2970–2982
- 17 Ma C, Chen H S, Li X, et al. Data based reconstruction of duplex networks. *SIAM J Appl Dyn Syst*, 2020, 19: 124–150
- 18 Wu J, Liu Y, Lu J, et al. Optimal reconstruction of probabilistic boolean networks. *IEEE Trans Cybern*, 2024, 54: 7656–7667
- 19 Wang H, Ma C, Chen H S, et al. Full reconstruction of simplicial complexes from binary contagion and Ising data. *Nat Commun*, 2022, 13: 3043
- 20 Chen W, Wassell I J. Energy efficient signal acquisition via compressive sensing in wireless sensor networks. In: *Proceedings of IEEE International Symposium on Wireless Pervasive Computing (ISWPC)*, Hong Kong, 2011. 1–6
- 21 Xie D, Peng H, Li L, et al. Semi-tensor compressed sensing. *Digital Signal Processing*, 2016, 58: 85–92
- 22 Zhang Y, Yang C, Huang K, et al. Reconstructing heterogeneous networks via compressive sensing and clustering. *IEEE Trans Emerg Top Comput Intell*, 2020, 5: 920–930
- 23 Dai J, Huang K, Liu Y, et al. Global reconstruction of complex network topology via structured compressive sensing. *IEEE Syst J*, 2020, 15: 1959–1969
- 24 Zhou J, Lu J. Topology identification of weighted complex dynamical networks. *Phys A-Stat Mech Its Appl*, 2007, 386: 481–491
- 25 Li X, Li X. Reconstruction of stochastic temporal networks through diffusive arrival times. *Nat Commun*, 2017, 8: 15729
- 26 Casadiego J, Nitzan M, Hallerberg S, et al. Model-free inference of direct network interactions from nonlinear collective dynamics. *Nat Commun*, 2017, 8: 2192
- 27 Zhao X, Zhou J, Zhu S, et al. Topology identification of multiplex delayed networks. *IEEE Trans Circuits Syst II*, 2020, 67: 290–294
- 28 Candes E J, Wakin M B. An introduction to compressive sampling. *IEEE Signal Process Mag*, 2008, 25: 21–30
- 29 Han X, Shen Z, Wang W X, et al. Robust reconstruction of complex networks from sparse data. *Phys Rev Lett*, 2015, 114: 028701
- 30 Wang W X, Yang R, Lai Y C, et al. Predicting catastrophes in nonlinear dynamical systems by compressive sensing. *Phys Rev Lett*, 2011, 106: 154101
- 31 Candès E, Romberg J. Sparsity and incoherence in compressive sampling. *Inverse Problems*, 2007, 23: 969–985
- 32 Zhang H F, Wang W X. Complex system reconstruction. *Acta Phys Sin*, 2020, 8: 281–291 [张海峰, 王文旭. 复杂系统重构. *物理学报*, 2020, 8: 281–291]
- 33 Candes E J, Tao T. Decoding by linear programming. *IEEE Trans Inform Theor*, 2005, 51: 4203–4215
- 34 Mei G, Wu X, Ning D, et al. Finite-time stabilization of complex dynamical networks via optimal control. *Complexity*, 2016, 21: 417–425
- 35 Mei G, Wu X, Chen G, et al. Identifying structures of continuously-varying weighted networks. *Sci Rep*, 2016, 6: 26649
- 36 Shi W, Ling Q, Yuan K, et al. On the linear convergence of the ADMM in decentralized consensus optimization. *IEEE Trans Signal Process*, 2014, 62: 1750–1761

- 37 Wang M, Wei S, Zhou Z, et al. Efficient ADMM framework based on functional measurement model for mmW 3-D SAR imaging. *IEEE Trans Geosci Remote Sens*, 2022, 60: 1–17
- 38 Boyd S, Parikh N, Chu E, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *FNT Machine Learn*, 2010, 3: 1–122
- 39 Liu Y, Huang K, Yang C, et al. Distributed network reconstruction based on binary compressed sensing via ADMM. *IEEE Trans Netw Sci Eng*, 2023, 10: 2141–2153

Network reconstruction based on node centrality sorting and alternating direction method of multipliers

Jianbo WANG^{1,2*}, Jianghua LIU¹, Zhanwei DU^{2*}, Ping LI¹ & Xiaoke XU^{3,4*}

1. *School of Computer Science and Software Engineering, Southwest Petroleum University, Chengdu 610500, China*

2. *School of Public Health, The University of Hong Kong, Hong Kong 999077, China*

3. *Computational Communication Research Center, Beijing Normal University, Zhuhai 519087, China*

4. *School of Journalism and Communication, Beijing Normal University, Beijing 100875, China*

* Corresponding author. E-mail: jwang@swpu.edu.cn, zwdu@hku.hk, xuxiaoke@bnu.edu.cn

Abstract Reconstructing complex network structures from observational time series is a prominent and challenging issue in network science. Recent advancements have seen extensive application of statistical inference and compressed sensing methods for network reconstruction. However, there remains significant room for improvement in terms of the accuracy, speed, and noise robustness of existing representative reconstruction methods. Building on classical compressed sensing algorithms, this paper introduces a network reconstruction algorithm that leverages node centrality ranking and the alternating direction method of multipliers (ADMM). The algorithm first reorders the node state feature matrix based on node centrality assessments, then applies compressed sensing to the reordered matrix, and finally employs ADMM to optimize the matrix for network topology reconstruction. Theoretical analysis demonstrates that the proposed algorithm has lower complexity compared to representative compressed sensing reconstruction methods. Experimental results show that the proposed algorithm enhances reconstruction accuracy, reduces reconstruction time, and offers greater noise robustness compared to existing methods. This research advances the accuracy, applicability, and reliability of network reconstruction algorithms for large-scale networks and has the potential for extension to weighted network reconstruction with various constraints in the future.

Keywords network reconstruction, compressive sensing, time series, node centrality, ADMM