软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2017,28(3):648−662 [doi: 10.13328/j.cnki.jos.005165]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于深度稀疏自动编码器的社区发现算法*

尚敬文, 王朝坤, 辛 欣, 应 翔

(清华大学 软件学院,北京 100084)

通讯作者: 王朝坤, E-mail: chaokun@tsinghua.edu.cn

摘 要: 社区结构是复杂网络的重要特征之一,社区发现对研究网络结构有重要的应用价值.*k*-均值等经典聚类算法是解决社区发现问题的一类基本方法.然而,在处理网络的高维矩阵时,使用这些经典聚类方法得到的社区往往不够准确.提出一种基于深度稀疏自动编码器的社区发现算法 CoDDA(a community detection algorithm based on deep sparse autoencoder),尝试提高使用这些经典方法处理高维邻接矩阵进行社区发现的准确性.首先,提出基于跳数的处理方法,对稀疏的邻接矩阵进行优化处理,得到的相似度矩阵不仅能够反映网络拓扑结构中相连节点间的相似关系,同时还反映了不相连节点间的相似关系.然后,基于无监督深度学习方法构建深度稀疏自动编码器,对相似度矩阵进行特征提取,得到低维的特征矩阵.与邻接矩阵相比,特征矩阵对网络拓扑结构有更强的特征表达能力.最后,使用 *k*-均值算法对低维特征矩阵聚类得到社区结构.实验结果显示:与 6 种典型的社区发现算法相比,CoDDA 算法能够发现更准确的社区结构.同时,参数实验结果显示,CoDDA 算法发现的社区结构比直接使用高维邻接矩阵的基本 *k*-均值算法发现的社区结构更为准确.

关键词: 社区发现;深度学习;CoDDA;*s*-跳;深度稀疏自动编码器

中图法分类号: TP311

中文引用格式: 尚敬文,王朝坤,辛欣,应翔.基于深度稀疏自动编码器的社区发现算法.软件学报,2017,28(3):648−662. http://www.jos.org.cn/1000-9825/5165.htm

英文引用格式: Shang JW, Wang CK, Xin X, Ying X. Community detection algorithm based on deep sparse autoencoder. Ruan Jian Xue Bao/Journal of Software, 2017,28(3):648−662 (in Chinese). http://www.jos.org.cn/1000-9825/5165.htm

Community Detection Algorithm Based on Deep Sparse Autoencoder

SHANG Jing-Wen, WANG Chao-Kun, XIN Xin, YING Xiang

(School of Software, Tsinghua University, Beijing 100084, China)

Abstract: Community structure is one of the most important features of complex network. Community detection is of great significance in exploring the network structure. Classical clustering algorithms such as *k*-means are the basic methods for community detection. However, the detection results are often not accurate enough when dealing with high-dimensional matrix when using these classical methods. In this study, a community detection algorithm based on deep sparse autoencoder (CoDDA) is proposed to improve the accuracy of community detection using high-dimensional adjacent matrix with the classical methods. First, a hop-based operation for sparse adjacent matrix is provided to obtain the similarity matrix, which can express not only the relations between nodes that are linked but also the relations between nodes that are not linked. Then, a deep sparse autoencoder based on unsupervised deep learning methods is designed to extract the features of similarity matrix and obtain the low-dimensional feature matrix which can represent the features of network topology better than similarity matrix. Finally, *k*-means is used to identify the communities according to the feature matrix. Experimental results show that CoDDA can obtain more accurate communities than the six baseline methods. Besides, the parameter analysis indicates

* 基金项目: 国家自然科学基金(61373023)
Foundation item: National Natural Science Foundation of China (61373023)
收稿时间: 2016-07-31; 修改时间: 2016-09-14; 采用时间: 2016-11-01; jos 在线出版时间: 2016-11-29
CNKI 网络优先出版: 2016-11-29 13:35:13, http://www.cnki.net/kcms/detail/11.2560.TP.20161129.1335.017.html

that CoDDA can result in more accurate communities than the *k*-means algorithm which finds the communities according to the high-dimensional matrix directly.

Key words: community detection; deep learning; CoDDA; *s*-hop; deep sparse autoencoder

复杂网络是由大量节点以及节点之间错综复杂的关系共同构成的网络结构[1,2].除了小世界和无标度等特性外,复杂网络还呈现出明显的社区结构.所谓社区(community),是指网络中的节点聚集而成的子图.同一社区的节点之间连接紧密,不同社区的节点之间连接稀疏[1,3].图 1 所示的示例网络包含 3 个社区(虚线框出),其中,相同颜色的节点属于同一社区.现实世界的许多网络都具有社区结构,例如:在社交网络中,人与人之间建立朋友关系,形成了不同类型的朋友圈;在学术合作网络中,研究者们共同发表学术著作,形成各个研究领域的学术圈;在蛋白质网络中,蛋白质之间频繁地交互,形成了结构单元.于是,近年来,复杂网络中的社区发现问题得到社会学、生物学与计算机科学等多个学科的广泛关注和深入研究[1,4−6],社区发现的研究成果也被成功应用于诸如好友推荐、个性化商品推介、蛋白质功能预测、舆情分析与处理等众多领域[1,2].

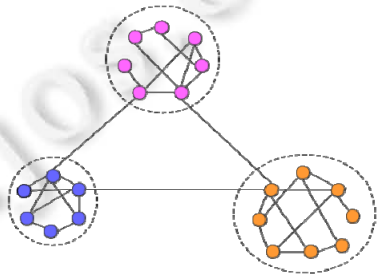


Fig.1 A sample network with community structures

图 1 一个具有社区结构的示例网络

社区发现致力于有效地寻找到复杂网络中准确的社区结构.目前,主流的社区发现方法有层次聚类方法(Radicchi[7])、矩阵分块方法(MB-DSGE[8])、骨架图方法(gCluSkeleton[9])、标签传播方法(LPA[10],HANP[11])以及图嵌入方法(DeepWalk[12])等,这些方法根据网络的拓扑结构,从不同的角度出发,解决复杂网络中的社区发现问题.

k-均值等经典的聚类算法是解决社区发现问题的一类基本方法[13],然而在处理网络的高维相似度矩阵时,使用这些经典的聚类算法进行社区发现存在如下问题.

(1) 使用邻接矩阵作为网络的相似度矩阵,不能全面地反映每个节点的局部信息.网络中除了直接相连的节点之间存在相似关系外,不直接相连的节点之间也存在不同程度的相似关系.例如:可以经过 *s* 跳到达彼此的节点对之间存在一定的相似关系,使用邻接矩阵作为网络的相似度矩阵只能简单地呈现直接相连的节点之间的相似关系,无法表示不直接相连的节点之间的相似关系.所以,邻接矩阵丢失了许多节点对的相似关系信息,不能全面地反映每个节点的局部信息.邻接矩阵包含信息有限的问题,影响社区发现的准确性;

(2) 高维相似度矩阵不能反映网络拓扑结构的主要特征,对网络拓扑结构中的社区结构没有很好的表达能力,在使用 *k*-均值等经典的聚类方法对高维相似度矩阵进行社区发现时,存在结果社区不准确的问题.

针对上述问题,本文提出了基于深度稀疏自动编码器的社区发现算法 CoDDA(a community detection algorithm based on deep sparse autoencoder).CoDDA 算法能够根据网络拓扑结构,实现较为准确的社区发现.本文的主要贡献有如下 3 点.

(1) 提出基于 *s*-跳的预处理方法,对稀疏的邻接矩阵进行优化处理.预处理得到的相似度矩阵不仅可以表示直接相连的节点对之间的相似关系,也可以为不直接相连的节点对添加一定的相似关系;

(2) 基于深度稀疏自动编码器这一无监督的深度学习方法,对相似度矩阵进行特征提取.提取特征后得到的低维特征矩阵能够更好地反映网络拓扑结构的主要特征,提高 *k*-均值等经典聚类算法的准确性;

(3) 在多个真实数据集上进行了大量实验.实验结果表明,本文提出的 CoDDA 算法可以得到更加准确的社区结构.

本文第 1 节介绍相关工作,包括已有的社区发现方法和深度学习方法.第 2 节给出基于跳数的相似度矩阵预处理操作的具体过程.第 3 节详尽阐述本文设计的深度稀疏自动编码器.第 4 节提出一种基于深度稀疏自动编码器的社区发现算法 CoDDA.第 5 节在多个真实的数据集上,从不同方面对 CoDDA 算法的有效性进行验证.第 6 节总结全文.

1 相关工作

本节介绍网络中社区的定义,并总结目前主流的 5 类社区发现方法.介绍深度学习的主要思想,并简介目前主流的有监督和无监督的深度学习方法.

1.1 社区发现

给定一个网络,社区发现根据网络中节点间的相互关系,将所有节点聚合成一系列子结构,即社区[1,3,14−17].与不同社区间节点之间的连接关系相比,同一社区内的节点之间通常具有较强的连接关系.目前,主流的社区发现算法分为层次聚类方法(Radicchi[7],GN[4],Newman[18],CNM[19])、矩阵分块方法(MB-DSGE[8],OQC[20])、骨架图方法(gCluSkeleton[9])、标签传播方法(LPA[10],HANP[11],SLPA[21])、图嵌入方法(Deepwalk[12],LE[22],GraRep[23]).其中,图嵌入方法与本文提出的 CoDDA 算法的思路最相似,都是先对矩阵进行降维操作,再得到社区.

基于层次聚类的方法根据网络图的层次化特点发现社区,包括自顶向下的分裂层次聚类方法和自底向上的凝聚层次聚类方法.分裂层次聚类方法将网络图分解为不同层次的子图,并从中寻找符合条件的社区.典型的分裂层次聚类算法有 Radicchi 算法[7]和 GN 算法[4].Radicchi 算法根据每条边所在的三角形数量计算边的聚集系数,并在每次迭代中删除聚集系数最小的边,直到生成符合条件的社区[7].凝聚层次聚类方法从网络图中每个节点出发,将满足条件的节点合并为不同层次的子图,并从中寻找符合条件的社区.典型的凝聚层次聚类算法有 Newman 算法[18]和 CNM[19]算法.矩阵分块方法通过重排网络图的邻接矩阵,将具有紧密连接关系的节点重新排列,并从中找出比较稠密的矩阵块,即社区.典型的矩阵分块算法有 MB-DSGE 算法[8]和 OQC 算法[20].MB-DSGE 算法通过构建一棵完整的层次树,根据密度阈值找到社区结构[8].骨架图聚类方法从网络图中提取出骨架图(skeleton graph),并根据骨架图得到社区,例如 gCluSkeleton 算法[9].gCluSkeleton 算法依据网络图的拓扑结构生成骨架图,将网络图包含的结构信息聚集到骨干网络中去,从冗杂的连接关系中提取出有效的社区结构.标签传播方法模拟社交关系在网络图中的传播过程生成社区,典型的标签传播算法有 LPA 算法[10]、HANP 算法[11]和 SLPA 算法[21].LPA 算法首先为每个节点赋予唯一的标签,进而模拟社交关系在网络图中的传播过程;待标签分布稳定后,相同标签的节点被视为同一社区内的成员.在 LPA 的基础上,HANP 算法通过添加节点偏好和衰减因子来控制标签的传播过程.图嵌入方法先对网络图的相似度矩阵进行降维操作,再计算社区结构,例如 Deepwalk 算法[12]、LE 算法[22]和 GraRep 算法[23].Deepwalk 算法使用随机游走和 skip-gram 模型得到网络图的表示矩阵,并计算出社区.

本文提出基于深度稀疏自动编码器这一无监督深度学习方法的社区发现算法 CoDDA,提高了使用 *k*-均值等经典聚类算法进行社区发现的准确性.

1.2 深度学习

深度学习方法[24−27]的主要思想是:通过建立具有多个层次的神经网络,实现对输入数据的深层次表达,从而实现更好的分类与特征抽取.其中,前一个层次的输出为后一个层次的输入.使用深度学习方法可以让计算机自动地学习到人工方法难以发现的重要特征.一个具有两个隐藏层的深度神经网络如图 2 所示.

深度学习方法在机器学习和人工智能等领域得到了充分的研究[24,28].针对深度学习的理论研究、算法设计

和应用系统在许多领域被广泛提出,例如语音识别、图像分类、自然语言处理等[29,30].目前,主流的深度学习方法有自动编码器(autoencoder,简称 AE)[31]、限制玻尔兹曼机(restricted Boltzmann machines,简称 RBM)[32]、深度置信网络(deep belief network,简称 DBN)[33]、卷积神经网络(convolutional neural network,简称 CNN)[34]、循环神经网络(recurrent neural network,简称 RNN)[35,36]等.其中,AE,RBM,DBN 是无监督的深度学习方法,CNN, RNN 是有监督的深度学习方法.

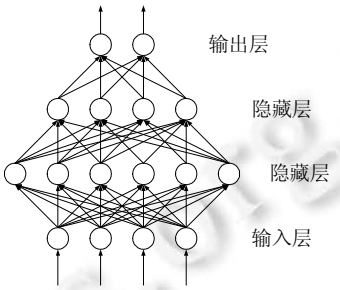


Fig.2 A deep neural network with 2 hidden layers

图 2 一个具有 2 层隐藏层的深度神经网络

AE 是 3 层神经网络,将输入表达编码为一个新的表达,然后解码,通过反向传播方法训练网络,使输出表达等于输入表达.在 RBM 中,我们可以通过输入表达 v 和条件概率 $p(h|v)$得到隐含层表达 h;反过来,也可以通过隐含层表达 h 和条件概率 $p(v|h)$得到一个新的输入表达.如此反复下去,通过调整参数,使输出表达等于输入表达.2006 年,Hinton 等人[37−39]提出了 DBN,在保证学习效果的同时,降低了学习隐藏层参数的难度.作为一个由多层 RBM 组成的神经网络,DBN 使用非监督的逐层贪婪方法与反向传播方法训练获得权值.

1998 年,Lecun 等人[40]提出的 CNN 提高了反向传播方法的训练性能.CNN 是多层神经网络,包括卷积层(alternating convolutional layer)和池化层(pooling layer),每层包括多个二维平面,每个平面包括多个独立神经元.作为另一种多层神经网络,RNN 的隐藏层之间的节点也存在有连接,并且隐藏层的输入不仅包括输入层的输出还包括上一时刻隐藏层的输出,通过这样的结构设计来处理序列数据.

CNN 等有监督的深度学习方法主要用于分类问题,AE 等无监督的深度学习方法在学习特征方面存在诸多优势.与 CNN 训练过程会丢失大量信息的特点不同,AE 致力于在特征提取时尽可能地减少信息的丢失.本文尝试利用 AE 的这一特点,将相似度高的节点聚集,同时将相似度低的节点分开.

2 矩阵预处理

本节给出 CoDDA 算法在矩阵预处理阶段的详细过程.

给定一个网络图 $G=(V,E)$,$V=\{v_1,v_2,\ldots,v_n\}$是节点集合,$E=\{e_1,e_2,\ldots,e_m\}$是边集合,$N(u)$是节点 u 的邻居节点集合.邻接矩阵 $A=[a_{ij}]_{n\times n}$ 表示节点间的连接关系,矩阵对应元素的值表示边是否存在:如果 v_i 和 v_j 之间存在边,则 $a_{ij}=1$;v_i 和 v_j 之间不存在边,则 $a_{ij}=0$.

通常,可以把邻接矩阵 A 作为网络的相似度矩阵来刻画网络中节点间的相似关系.然而,网络中除了直接相连的节点之间存在相似关系以外,不直接相连的节点之间也存在不同程度的相似关系.例如:经过有限跳数可以彼此到达的两个节点之间存在一定的相似关系.使用邻接矩阵作为网络的相似度矩阵只能简单地呈现直接相连的节点之间的相似关系,无法表示不直接相连的节点之间的相似关系.所以,邻接矩阵丢失了许多节点之间的相似关系信息,不能反映每个节点完整的局部信息.邻接矩阵包含信息有限的问题,影响社区发现的准确性.

因此,为了更充分地描述每个节点的局部信息,本节提出基于跳数的方法,根据网络的邻接矩阵 A,计算节点间的相似关系,得到新的相似度矩阵.下面给出 s-跳和本文提出的相似度矩阵的定义.

定义 1(s-跳). 给定一个网络图 $G=(V,E)$,$u\in V$,对点集内的任意节点 $u\in V$,若节点 u 可以在最少经过 s 跳之后

到达节点 v,即:节点 u 到节点 v 的最短路径的长度是 s,则称节点 u 可以经过 s-跳(s-hop)到达节点 v.

如图 3 所示,在一个网络图中,节点 v_1 可以经过 2 跳(2-hop)到达节点 v_4,可以经过 3 跳(3-hop)到达节点 v_5,可以经过 2 跳(2-hop)到达节点 v_6.

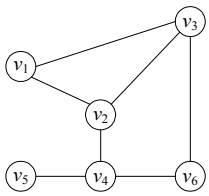


Fig.3 A sample network

图 3 一个示例网络图

定义 2(节点相似度). 给定一个网络图 $G=(V,E)$,$u,v\in V$,u 和 v 之间的相似度 $Sim(u,v)$定义为

$$Sim(u,v)=e^{\sigma(1-s)} \tag{1}$$

其中,节点 u 经过 s-跳到达节点 v,$s\geqslant 1$.σ称为衰减因子,$\sigma\in(0,1)$.随着跳数 s 的增加,相似度不断减小.σ控制了相似度的衰减程度,σ越大,节点间相似关系衰减得越快.

定义 3(相似度矩阵 X). 给定一个网络图 $G=(V,E)$,$X=[x_{ij}]_{n\times n}$ 是与 G 对应的一个矩阵.如果使用公式(1)计算 X 中对应两个节点 v_i 和 v_j 之间的相似度 $x_{ij}=Sim(v_i,v_j)$,$v_i,v_j\in V$,则称 X 为 G 的相似度矩阵.

使用跳数和衰减因子计算两个不直接相连的节点之间的相似关系,可以更好地反映社区的拓扑结构,提高社区发现的准确性.但是当跳数大于一定阈值时,两个不在同一个社区内的节点也会得到一定的相似度值,这使得社区结构的边界更加模糊.因此,设置跳数阈值 S,只计算在 S 跳内可以相互到达的节点之间的相似度,保证在增强图的拓扑结构信息的同时,不影响社区边界的划分.实验部分对跳数阈值 S 和衰减因子σ进行分析,研究不同的跳数阈值 S 和衰减因子σ取值对结果的影响.

3 特征提取

本节给出 CoDDA 算法在特征提取阶段的具体过程.首先介绍稀疏自动编码器的定义,然后构建深度稀疏自动编码器,对 $n\times n$ 的相似度矩阵 X 进行特征提取,得到 $n\times d$ 的低维特征矩阵 Y.特征矩阵 Y 能够显示更主要的图拓扑结构特征[41].

3.1 稀疏自动编码器

定义 4(自动编码器). 自动编码器(autoencoder,简称 AE)是一个 3 层的神经网络.AE 对用户输入进行编码,然后解码得到输出,使用反向传播算法来训练网络,使得输出等于输入,得到编码结果.

自动编码器通过尽可能去复现输入数据的方法,去提取输入数据的特征.它是一种无监督的模型,训练样本没有类别标签,结构如图 4 所示,形象表示如图 5 所示.

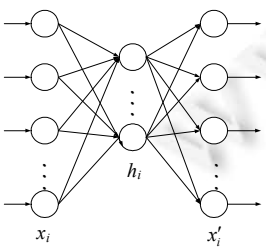


Fig.4 The structure of an autoencoder (AE)

图 4 自动编码器的结构

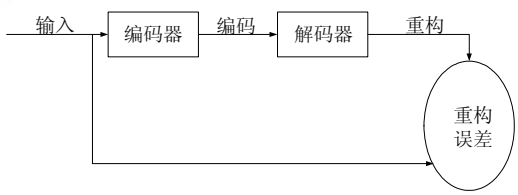


Fig.5 Autoencoder (AE) representation

图 5 自动编码器的形象表示

第 1 层到第 2 层相当于一个编码(encode)过程.给定网络图 G 的相似度矩阵 $X=\{x_1,x_2,\ldots,x_n\}$,我们输入 G 的一个节点在 X 中对应的向量 $x_i\in R^n$ 后,自动编码器编码得到这个节点对应的低维向量 $h_i\in R^d$.从第 2 层到第 3 层相当于一个解码(decode)过程.自动编码器将这个节点对应的低维向量 h_i 进行解码,得到与 x_i 具有相同维度的输出向量 x_i'.我们使用反向传播算法来训练网络,通过调整编码器和解码器的参数,最小化重构误差,使得输出向量 x_i' 等于输入向量 x_i.最后,我们把获得的编码层结果,即这个节点对应的低维向量 h_i 作为特征结果.

在自动编码器中,我们设定具体的训练过程如下.

我们将网络图 G 的相似度矩阵 X 作为自动编码器的输入矩阵,$x_i\in R^{n\times 1}$ 是相似度矩阵 X 的第 i 个节点对应的向量,我们将 x_i 作为自动编码器的第 i 个输入向量.将 x_i 输入到一个具有 d 个神经元的编码层,通过公式(2)得到编码 $h_i\in R^{d\times 1}$.

$$h_i = s_f(Wx_i + p) \tag{2}$$

其中,s_f 是编码器的一个激活函数,如 sigmoid 函数(sigmoid(x)=1/(1+exp(–x)).$W\in R^{d\times n}$ 是权重矩阵,$p\in R^{d\times 1}$ 是编码层偏置向量.

得到编码 $h_i\in R^{d\times 1}$ (即第 i 个节点对应的低维向量)后,我们将 h_i 输入到解码层,通过公式(3)得到解码结果 $x_i'\in R^{n\times 1}$ 作为输出信息:

$$x_i' = s_g(\tilde{W}h_i + q) \tag{3}$$

其中,s_g 是解码器的一个激活函数.$\tilde{W} = W^T \in R^{n\times d}$ 是权重矩阵,$q\in R^{n\times 1}$ 是解码层偏置向量.

通过训练,自动编码器自动调整权重矩阵和偏置向量这 4 个参数 $\theta=\{W,\tilde{W},p,q\}$,最小化 x_i 和 x_i' 的重构误差:

$$\underset{W,\tilde{W},p,q}{minimize}\sum_{i=1}^{n}\| s_g(\tilde{W}s_f(Wx_i + p) + q) - x_i \|_2^2 \tag{4}$$

此外,我们使用 KL 散度,为自动编码器添加稀疏性限制:

$$\sum_{j=1}^{d} KL\left(\rho \| \frac{1}{n}\sum_{i=1}^{n} h_i\right) \tag{5}$$

其中,使用 ρ_j 来表示隐藏层的平均活跃度 $\rho_j = \frac{1}{n}\sum_{i=1}^{n} h_i$. $KL(\rho\|\rho_j)$ 表示分别以 ρ_j 和ρ为均值的两个变量之间的相对熵.ρ是一个接近于 0 的常量,例如取 0.01. $KL(\rho\|\rho_j)$ 的计算公式如下:

$$KL(\rho\|\rho_j)=\rho\log\frac{\rho}{\rho_j} + (1-\rho)\log\frac{1-\rho}{1-\rho_j} \tag{6}$$

本文构建的自动编码器的重构误差可以总结为如下公式:

$$L(\theta) = \sum_{i=1}^{n}\| s_g(\tilde{W}s_f(Wx_i + p) + q) - x_i \|_2^2 + \alpha\sum_{j=1}^{d} KL\left(\rho \| \frac{1}{n}\sum_{i=1}^{n} s_f(Wx_i + p)\right) \tag{7}$$

其中,α是控制稀疏限制的权重因子.

3.2 深度稀疏自动编码器

本文构建的深度稀疏自动编码器由多层稀疏自动编码器构成,结构如图 6 所示,前一层自动编码器的输出是后一层的输入.我们使用逐层贪婪的训练方法,依次训练深度稀疏自动编码器的每一层,进而训练整个深度神经网络.

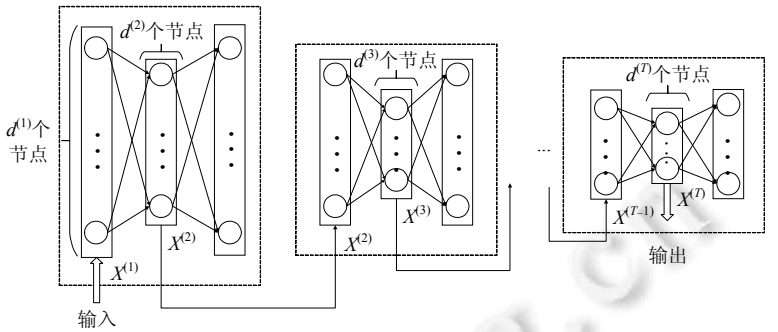


Fig.6 Deep sparse autoencoder

图 6 深度稀疏自动编码器的结构

本文基于深度稀疏自动编码器,实现对相似度矩阵 X 的特征提取.首先设置深度稀疏自动编码器的结构,包括层数 T 和每层的节点数 $\{d^{(1)},d^{(2)},\ldots,d^{(T)}\}$,其中,$d^{(1)}=n$;接着,我们将相似度矩阵 $X^{(1)}=X,X\in R^{n\times n}$ 作为深度稀疏自动编码器的第 1 层输入,将 $X^{(1)}$ 输入到有 $d^{(2)}$ 个节点的自动编码器,提取训练得到的编码层结果 $X^{(2)}\in R^{n\times d^{(2)}}$;然后,将矩阵 $X^{(2)}$ 输入到有 $d^{(3)}$ 个节点的自动编码器,提取训练得到的编码层结果 $X^{(3)}\in R^{n\times d^{(3)}}$;继续循环操作,直到从最后一个自动编码器中提取出训练得到的编码层结果 $X^{(T)}\in R^{n\times d^{(T)}}$;最后,输出深度稀疏自动编码器的低维特征矩阵 $X^{(T)}$.

4 基于深度稀疏自动编码器的社区发现算法

本节提出基于深度稀疏自动编码器的社区发现算法 CoDDA,主要包括 3 个步骤:基于跳数的邻接矩阵预处理,即,使用广度优先遍历 BFS,对邻接矩阵进行处理,得到相似度矩阵(已在第 2 节详细介绍);构建深度稀疏自动编码器对相似度矩阵进行特征提取,以得到特征明显的低维特征矩阵(已在第 3 节详细介绍);使用 k-均值方法聚类得到社区.具体的算法流程见算法 1.

算法 1. CoDDA.

输入:网络图 $G=(V,E)$的邻接矩阵 $A\in R^{n\times n}$,社区个数 k;跳数阈值 S,衰减因子σ,深度稀疏自动编码器的层数 T,每层节点数 $d^{(j)},j=1,\ldots T;d^{(1)}=n$;

输出:社区结果$\{C_1,C_2,\ldots,C_k\}$.

```
1.  FOR each x in V
2.    将图 G 中每个节点初始化为未访问状态;
3.    初始化队列 Queue=∅;初始化跳数集合 D=∅;
4.    将 x 标记为访问中状态,初始化 x 的跳数为 0,将 x 及跳数 0 写进跳数集合 D,并将 x 入队列 Queue;
5.    WHILE Queue≠∅
6.      从队列 Queue 中取出节点 u;
7.      FOR each v in N(u)
8.        IF u 在 x 的(S–1)-跳内 and v 处于未访问状态
9.          将 v 设置为正在访问状态;
10.         x 到 v 的跳数等于 x 到 u 的跳数加 1;
11.         将 v 及 x 到 v 的跳数写进跳数集合 D;
12.         将 v 加入队列 Queue;
13.       End
14.     End
15.     将 u 设置为访问结束状态;
```

16. **End**
17. **FOR** each v in V
18. 根据跳数集合 D 及公式(1)计算 x 和 v 的相似度 $Sim(x,v)$;
19. **End**
20. **End**
21. 得到基于跳数的相似度矩阵 X;
22. $X^{(1)}=X$;
23. **FOR** j=1 to T
24. 建立一个稀疏自动编码器;
25. 输入特征矩阵 $X^{(j)}$;
26. 通过优化公式(7)训练稀疏自动编码器;
27. 得到隐含层的表示 $H^{(j)}$;
28. $X^{(j+1)}=H^{(j)}$;
29. **End**
30. 对低维特征矩阵 $X^{(T)}\in R^{n\times d^{(T)}}$ 运行 k-均值,聚类得到结果社区 $Coms=\{C_1,C_2,\ldots,C_k\}$;
31. **Return** 社区结果 $Coms=\{C_1,C_2,\ldots,C_k\}$.

首先,CoDDA 算法的输入包括 3 部分:图 $G=(V,E)$的邻接矩阵 $A\in R^{n\times n}$,社区个数 k;基于跳数的邻接矩阵预处理阶段需要的参数,即,跳数阈值 S 和衰减因子σ;深度稀疏自动编码器特征提取阶段需要的参数,即,层数 T 和每层节点数 $d^{(j)},j=1,\ldots T,d^{(1)}=n$.

从算法 1 的第 1 行~第 21 行,使用基于跳数的邻接矩阵优化处理方法,生成相似度矩阵 X.其中,从第 5 行~第 16 行,对图 G 中的每个节点 x,使用 BFS 广度优先遍历的方法找到 x 在 S 跳内可达的节点 v,将 v 及 x 和 v 之间的跳数写进跳数集合 D.从第 17 行~第 19 行,计算 x 与点集 V 内其他节点的相似度:如果 v 在 D 内,那么使用公式(1)计算得到结果 $Sim(x,v)$;否则,$Sim(x,v)=0$.

从第 22 行~第 29 行,对相似度矩阵 X 进行特征提取.循环执行 T 次,每次使用一个稀疏自动编码器从编码层提取出低维特征矩阵 $H^{(j)}$,令 $X^{(j+1)}=H^{(j)}$作为下次循环的输入矩阵.循环结束,得到低维的特征矩阵 $X^{(T)}\in R^{n\times d^{(T)}}$.第 30 行对低维特征矩阵使用 k-均值方法,聚类得到结果社区$\{C_1,C_2,\ldots,C_k\}$.

5 实验结果与分析

本节使用真实数据集对 CoDDA 算法进行性能评价.我们首先介绍实验环境、数据集、评价标准和典型的比较算法,然后进行比较实验、可视化实验和参数实验.

5.1 实验准备

本文的实验环境设置为:Windows Server 2008 操作系统,Intel Xeon 2.0GHz CPU,256GB 内存.本文提出的算法及实验基于 Java 语言(JDK 1.7.0)和 Matlab(R2015b)编码实现.

使用 4 个真实数据集 Strike,Football[42],LiveJournal,Orkut[43]进行实验.Strike 是锯木工人关于罢工话题的通信网络;Football 是 2006 年 NCAA 足球比赛的对阵关系网络;LiveJournal 是用户之间可以标记朋友关系的在线博客网络;Orkut 是一个免费的在线社交网络,用户可以建立朋友关系,同时还可以建立朋友群组.

从 4 个数据集中分别抽取 3 个、11 个、8 个、12 个社区的节点及连接这些节点的边组成新的网络,用于本节的实验.详细信息见表 1,每个数据集包括网络拓扑结构和真实社区(ground-truth)信息.

本文构建的深度神经网络在不同数据集上的层数设置是不同的,详细信息见表 2.

对于小规模的数据集,选择层数小的深度神经网络即可达到很好的特征提取的效果.对于大规模数据集,按照 2 的幂次,依次降低深度神经网络中每层节点的个数.同时,在深度神经网络中,设置学习速率为 lr=0.1,考虑稀

疏性限制,设置ρ=0.01.这样的参数设置可以有效地利用深度自动编码器逐层贪婪训练的特点,保证在不同规模数据集上特征提取的准确性.

Table 1 Dataset information

表 1 数据集信息

数据集	节点数	边数	社区个数
Strike	24	38	3
Football	180	788	11
LiveJournal	6 000	89 019	8
Orkut	30 692	582 132	12

Table 2 Structure of deep neural network

表 2 深度神经网络的结构

数据集	节点数
Strike	24-16
Football	180-128
LiveJournal	6000-4096-2048
Orkut	30692-16384-8192-4096-2048

5.2 评价标准

根据真实社区 $GT=\{C'_1,C'_2,...,C'_t\}$,判断结果社区 $Coms=\{C_1,C_2,\ldots,C_k\}$的准确性,其中,$t$ 是真实社区(ground-truth)的数量,k 是结果社区的数量.使用 F_{same}[10]和 NMI(normalized mutual information)[44]这 2 个通用的社区评价标准对社区的精确度进行分析,同时考虑到在许多真实数据集中通常难以获得真实社区的信息,因此,我们使用 Q[19]这一评价指标对社区的质量进行度量.

(1) F_{same}:是对社区结果与真实社区的交叉和近似程度的衡量指标,计算公式如下.

$$F_{same}=\frac{1}{2n}\left(\sum_{i=1}^{k}\max_{j}|C_i\cap C'_j|+\sum_{j=1}^{t}\max_{i}|C_i\cap C'_j|\right) \tag{8}$$

其中,n 是图 G 中节点的数量.

(2) NMI:归一化互信息,计算公式如下.

$$NMI=\frac{-2\sum_{ij}N_{ij}\log\frac{N_{ij}N_t}{N_{i.}N_{.j}}}{\sum_i N_{i.}\log\frac{N_{i.}}{N_t}+\sum_j N_{.j}\log\frac{N_{.j}}{N_t}} \tag{9}$$

其中,N 是混淆矩阵,每个元素 N_{ij} 代表了 C_i 和 C'_j 的公共节点数,$N_{i.}$和 N_j 分别代表了矩阵中第 i 行和第 j 列中元素的和,且 $N_t=\sum_i\sum_j N_{ij}$.

(3) Q:模块性(modularity)是使用最广泛的衡量社区质量的评价标准之一,计算公式如下.

$$Q=\sum_{i=1}^{k}\left(\frac{E_i^{in}}{m}-\left(\frac{2E_i^{in}+E_i^{out}}{2m}\right)^2\right) \tag{10}$$

其中, E_i^{in} 是社区 C_i 的内部边的数量, E_i^{out} 是社区 C_i 的外部边的数量,m 是图 G 中边的个数.

5.3 比较算法

将本文提出的算法 CoDDA 与已有的典型算法进行比较.比较算法详细信息如下.

(1) Radicchi 算法:是一种分裂层次聚类算法,根据每条边所在的三角形数量计算边的聚集系数,每次迭代将删除聚集系数最小的边,直到产生不相连的子图作为社区结果;

(2) MB-DSGE 算法:是一种矩阵分块算法,通过构建一棵完整的层次树,根据密度阈值找到社区结构;

(3) gCluSkeleton 算法:是一种骨架图算法,依据网络图的拓扑结构生成骨架图,将网络图包含的结构信息聚集到骨干网络中去,从冗杂的连接关系中提取出有效的社区结构;

(4) LPA 算法:是一种标签传播算法,模拟社交关系在网络中的传播过程.待标签分布稳定后,相同标签的节点被分到同一社区中;

(5) HANP 算法:也是一种标签传播算法,是 LPA 算法的改进算法,通过添加节点偏好和衰减因子的方式,控制标签的传播过程;

(6) Deepwalk 算法:是一种图嵌入算法,通过使用随机游走和 skip-gram 模型,得到图的低维矩阵表示,并计算出社区.

5.4 实验结果

通过比较实验和可视化实验,将已有的算法与本文提出的 CoDDA 算法进行对比,展示本算法的优势;通过参数实验,展示基于深度稀疏自动编码器的 CoDDA 算法的特征提取过程对社区准确性的贡献.

5.4.1 比较实验

为验证算法在社区准确性方面的优势,使用 CoDDA 算法和已有的典型算法进行比较,得到的社区评价指标 F_{same},NMI 和 Q 的结果见表 3.

Table 3 Analysis of community detection results

表 3 社区发现结果分析

算法	Strike			Football			LiveJournal			Orkut		
	F_{same}	NMI	Q	F_{same}	NMI	Q	F_{same}	NMI	Q	F_{same}	NMI	Q
Radicchi	0.79	**1.0**	0.46	0.50	0.63	0.34	0.62	0.52	0.44	0.52	0.01	0.01
LPA	0.90	0.77	0.54	0.66	0.66	0.57	0.87	0.76	0.50	0.68	0.69	0.75
HANP	0.92	0.87	**0.55**	0.65	0.67	0.56	0.88	0.78	0.50	0.70	0.71	**0.78**
MB-DSGE	0.79	0.70	0.52	0.34	0.34	0.57	0.19	0.42	0.16	0.34	0.51	0.23
gCluSkeleton	0.96	0.87	**0.55**	0.27	0.27	0.55	0.70	0.49	0.45	0.27	0.49	0.32
Deepwalk	0.58	0.62	0.06	0.01	0.01	0.30	0.49	0.03	0.02	0.01	0.01	0.01
CoDDA	**1.0**	**1.0**	**0.55**	**0.91**	**0.92**	**0.58**	**0.91**	**0.83**	**0.51**	**0.77**	**0.75**	**0.78**

表 3 的结果显示,CoDDA 算法得到的社区比其他典型的算法更为准确.这是因为处理高维的邻接矩阵时,CoDDA 使用基于跳数的预处理方法,有效地完善了节点的局部信息,并使用深度稀疏自动编码器对相似度矩阵进行特征提取,得到特征更加明显的低维特征矩阵,进而聚类得到更准确的社区.Deepwalk 算法的效果不甚理想,这也与文献[23]中的情况吻合.

与基于层次聚类的 Radicchi 算法、基于矩阵分块的 MB-DSGE 算法、基于骨架图的 gCluSkeleton 算法、基于标签传播的 LPA 算法和 HANP 算法以及基于图嵌入的 Deepwalk 算法等比较算法不同,本文构建的由深度稀疏自动编码器构成的深度神经网络具有较好的特征表达能力.CoDDA 算法采用逐层贪婪训练法进行训练,即:前一层自动编码器的输出作为后一层自动编码器的输入,这样通常能够获取到输入数据的层次型分解结构.例如,如果网络的输入数据是反映图的拓扑结构的相似度矩阵,那么网络的第 1 层会学习如何去识别网络中的重要节点,第 2 层会学习如何去组合这些重要节点,从而构成局部密集的子图,更高层会学习如何去组合这些密集子图,生成更形象且更有意义的子图,最终学得如何形成准确的社区.

CoDDA 算法的运行主要分为离线和在线两个阶段,其中,离线阶段进行特征的提取,在线阶段进行 k-均值聚类.我们对算法在线阶段的运行时间进行统计,不同数据集上 CoDDA 算法和比较算法的运行时间见表 4.实验结果表明,本文提出的 CoDDA 算法在时间上与目前主流的社区发现方法具有一定的可比性.

测试离线阶段特征提取过程的时间开销,结果见表 5.虽然在大数据集上的运行时间稍长,但这也符合深度学习的特点.并且,离线阶段进行的特征提取不会对在线阶段社区发现的效率产生影响.

Table 4 Running time on different datasets

表 4 不同数据集上的运行时间

数据集	Strike (s)	Football (s)	LiveJournal (s)	Orkut (h)
Radicchi	0.001	0.003	1.58	0.009
LPA	0.012	0.025	1.46	0.01
HANP	0.056	0.076	4.7	0.04
MB-DSGE	0.011	0.021	17.51	0.08
gCluSkeleton	0.032	0.045	185.14	1.5
Deepwalk	0.07	0.008	12.71	5.50
CoDDA	0.005	0.004	11.75	4.45

Table 5 Offline training time for CoDDA

表 5 CoDDA 在离线阶段的运行时间

数据集	Strike	Football	LiveJournal	Orkut
CoDDA	0.01s	29.54s	30h	52h

5.4.2 可视化实验

针对 Strike 数据集和 Football 数据集,使用比较算法和 CoDDA 算法进行可视化分析.因为相比其他算法,Deepwalk 算法效果不够理想,所以没有对其进行可视化实验.

对 Strike 数据集使用比较算法和 CoDDA 算法得到的社区和真实社区如图 7 所示,相同颜色的节点属于相同社区,不同颜色的节点属于不同社区.

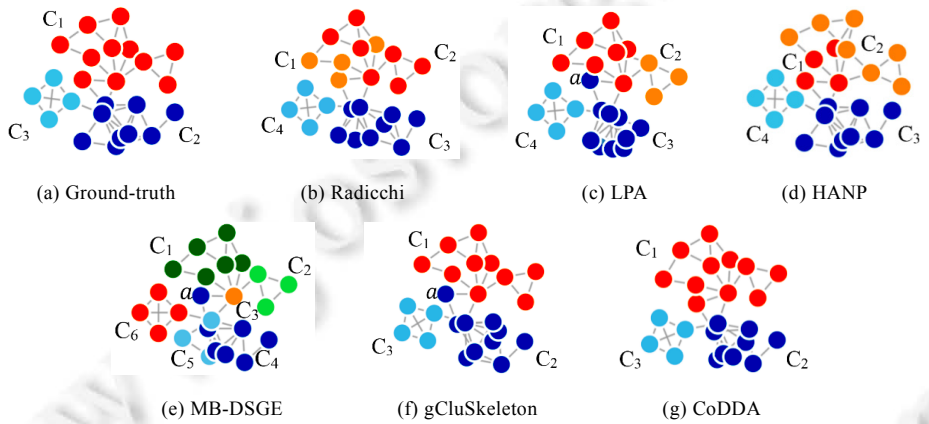


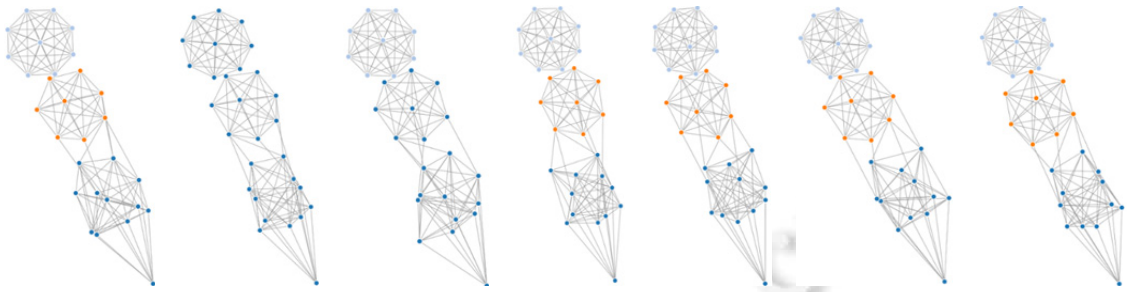
Fig.7 Comparation of ground-truth and communities from different methods in Strike

图 7 Strike 数据集的真实社区和使用不同算法得到的社区比较

实验结果表明,本文提出的 CoDDA 算法得到的社区结果最接近真实的社区结果.如图 7(a)所示,Strike 数据集含有的 24 个节点属于 3 个真实社区 C_1,C_2 和 C_3.如图 7(b)所示,Radicchi 算法将原本属于同一社区的节点分到了 C_1 和 C_2 两个社区.如图 7(c)所示,LPA 算法将原本属于同一社区的节点分到了 C_1 和 C_2 两个社区,并将节点 a 错误地分到了另一个社区 C_3,这显然是不合理的.如图 7(d)所示,HANP 算法将原本属于同一社区的节点分到了 C_1 和 C_2 两个社区.如图 7(e)所示,MB-DSGE 算法将原本属于同一社区的节点分到了 C_1,C_2 和 C_3 这 3 个社区,并将节点 a 错误地分到了另一个社区 C_4;同时,将原本属于同一社区的节点分到了 C_4 和 C_5 两个社区,得到的社区与真实社区相差较大.如图 7(f)所示,gCluSkeleton 算法将节点 a 错误地分到了另一个社区 C_2.如图 7(g)所示,CoDDA 算法得到的社区结果与真实社区完全一致.实验结果证实了本文提出的 CoDDA 算法的优势.

选择 Football 数据集的 3 个真实社区(社区结构最明显)的节点,使用比较算法和 CoDDA 算法得到的社区如图 8 所示.因为社区结构明显,所以使用 HANP 算法、MB-DSGE 算法、gCluSkeleton 算法和 CoDDA 算法得到的社区与真实社区是一致的.但是,即使对于社区结构明显的节点,Radicchi 算法仍将 3 个社区识别为同一个社区,LPA 算法也将其中 2 个社区识别为同一个社区,这显然是不合理的.本文提出的 CoDDA 算法得到的结果与真实社区是一致的,这验证了 CoDDA 算法的准确性.

实验结果显示:不管是 Strike 数据集,还是社区结构明显的 Football 数据集,本文提出的 CoDDA 算法都能够得到准确的社区,证实了 CoDDA 算法优势的稳定性.另一方面,比较算法使用不同的数据集得到的社区存在不同程度的不稳定性.



(a) Football 真实社区 (b) Radicchi (c) LPA (d) HANP (e) MB-DSGE (f) gCluSkeleton (g) CoDDA

Fig.8 Comparation of ground-truth and communities from different methods in Football

图 8 Football 数据集中,3 个真实社区的节点使用不同算法得到的社区比较

5.4.3 参数实验

本节分析跳数阈值 S、衰减因子σ和深度稀疏自动编码器的层数对实验结果的影响.比较 CoDDA 算法得到的社区与直接使用 k-均值算法对相似度矩阵聚类得到的社区,来展示 CoDDA 算法中的基于深度稀疏自动编码器的特征提取操作可以有效提高社区结果的准确性.

(1) 跳数阈值 S

针对 Strike 数据集,设置深度稀疏自动编码器各层的节点数是[24-16],衰减因子σ=0.5,分析不同跳数阈值 S 对 NMI 的影响.根据图 9 所示,在跳数阈值 S 的不同取值下,对比直接使用 k-均值算法对相似度矩阵进行聚类,CoDDA 算法得到的结果社区都更为准确.实验结果显示,CoDDA 算法基于深度稀疏自动编码器进行特征提取的操作可以很好地提高社区结果的准确性.

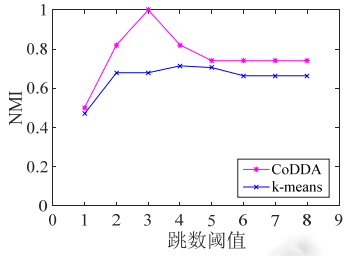


Fig.9 NMI of CoDDA and k-means with different values of S in Strike

图 9 Strike 数据集中不同跳数阈值 S 下,使用 CoDDA 算法与 k-均值算法的 NMI 值

同时,随着跳数阈值S的增加,NMI 呈现先增加后减小的趋势.这样的实验结果说明:考虑不直接相连但在一定跳数内可达的节点对的相似度,可以有效地反映每个节点局部的结构信息.但是如果跳数阈值过大,距离过远不在相同社区的节点之间也增加了一定的相似度值,这不利于社区边界的识别,使得社区准确性降低.对于小规模数据集 Strike 和 Football,分别选择小的跳数阈值 3-跳和 1-跳;对于大规模数据集 LiveJournal 和 Orkut,选择稍大的跳数阈值 8-跳,以达到最优的结果.

(2) 衰减因子σ

针对 Strike 数据集,设置深度稀疏自动编码器每层的节点数[24-16],跳数阈值 S=3,分析不同衰减因子σ的取值对实验结果的影响.根据图 10 所示,相比直接使用 k-均值算法对相似度矩阵的聚类操作,CoDDA 算法在不同衰减因子取值下得到的结果社区更精确.从实验的角度展示了 CoDDA 算法中基于深度稀疏自动编码器进行特征提取的操作在社区准确性方面的贡献.

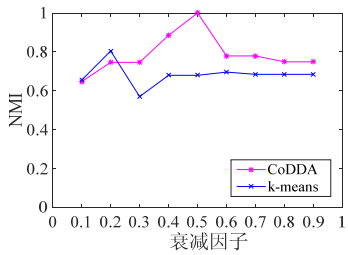


Fig.10 NMI of CoDDA and *k*-means with different values of σ in Strike

图 10 Strike 数据集中不同衰减因子σ下,使用 CoDDA 算法与 *k*-均值算法的 NMI 值

同时,随着衰减因子的增加,NMI 整体呈现先增加后减小的趋势.因为衰减因子控制相似度随着跳数的增加的衰减程度,所以对于小规模数据集 Strike 和 Football,选择稍大的衰减因子σ=0.5,以避免衰减因子过大时对社区边界产生的模糊作用;对于大规模数据集 LiveJournal 和 Orkut,选择小的衰减因子σ=0.1,增强节点的局部特征,以达到最优的结果.

(3) 深度稀疏自动编码器的层数

针对 LiveJournal 数据集,设置跳数阈值 S=8,衰减因子σ=0.1,分析不同层数的深度稀疏自动编码器对实验结果的影响,其中,1~6 层对应的节点数分别为 6 000,4 096,2 048,1 024,512 和 256.

根据图 11 所示,CoDDA 算法在 3 层深度稀疏自动编码器(每层节点是 6000-4096-2048)的设置下性能最好.随着深度稀疏自动编码器层数的增加,社区准确度先增加后减小.实验结果显示:深度稀疏自动编码器这一无监督深度学习方法可以提取出使社区结构更加明显的特征信息,提高结果社区的准确性.但是如果层数过高,会过滤掉一些特征信息,降低结果社区的准确性.因此,对于小规模数据集 Strike 和 Football,分别选择稍低的训练层数[24-16]和[180-128];对于大规模数据集 LiveJournal 和 Orkut,分别选择稍大的训练层数[6000-4096-2048]和[30692-16384-8192-4096-2048],以提供更准确的特征提取结果.

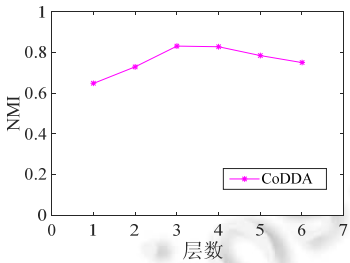


Fig.11 NMI of CoDDA with different numbers of deep spare autoencoder layers in Strike

图 11 Strike 数据集中在不同层数的深度稀疏自动编码器中使用 CoDDA 算法的 NMI 值

6 结束语

本文提出了基于深度稀疏自动编码器的社区发现算法 CoDDA,尝试使用无监督深度学习的方法,根据网络拓扑结构,提高使用例如 *k*-均值等经典的聚类方法进行社区发现的准确性.首先提出了基于 *s*-跳数的邻接矩阵预处理方法,得到能够更好地反映节点局部信息的相似度矩阵;然后构建深度稀疏自动编码器,对相似度矩阵进行深度特征提取;最后,使用 *k*-均值方法聚类得到社区.实验结果显示:与典型的社区发现算法相比,本文提出的 CoDDA 算法可以得到更准确的社区结构.并且,与直接使用高维相似度矩阵的 *k*-均值方法相比,CoDDA 算法得到的社区结果更为准确.

References:

[1] Fortunato S. Community detection in graphs. Physics Reports, 2010,486(3):75−174. [doi: 10.1016/j.physrep.2009.11.002]

[2] Huang FL, Zhang SC, Zhu XF. Discovering network community based on multi-objective optimization. Ruan Jian Xue Bao/Journal of Software, 2013,24(9):2062−2077 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4400.htm [doi: 10.3724/SP.J.1001.2013.04400]

[3] Lancichinetti A, Fortunato S, Radicchi F. Benchmark graphs for testing community detection algorithms. Physical Review E, 2008, 78(4):046110. [doi: 10.1103/PhysRevE.78.046110]

[4] Newman MEJ, Girvan M. Finding and evaluating community structure in networks. Physical Review E, 2004,69(2):026113. [doi: 10.1103/PhysRevE.69.026113]

[5] Newman MEJ. The structure and function of complex networks. SIAM Review, 2003,45(2):167−256. [doi: 10.1137/S00361445034 2480]

[6] Zhou XP, Liang X, Zhang HY. User community detection on micro-blog using R-C model. Ruan Jian Xue Bao/Journal of Software, 2014,25(12):2808−2823 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4720.htm [doi: 10.13328/j.cnki.jos. 004720]

[7] Radicchi F, Castellano C, Cecconi F, Loreto V, Parisi D. Defining and identifying communities in networks. Proc. of the National Academy of Sciences of the United States of America, 2004,101(9):2658−2663. [doi: 10.1073/pnas.0400054101]

[8] Chen J, Saad Y. Dense subgraph extraction with application to community detection. IEEE Trans. on Knowledge and Data Engineering, 2012,24(7):1216−1230. [doi: 10.1109/TKDE.2010.271]

[9] Huang J, Sun H, Song Q, Deng H, Han J. Revealing density-based clustering structure from the core-connected tree of a network. IEEE Trans. on Knowledge and Data Engineering, 2013,25(8):1876−1889. [doi: 10.1109/TKDE.2012.100]

[10] Raghavan UN, Albert R, Kumara S. Near linear time algorithm to detect community structures in large-scale networks. Physical Review E, 2007,76(3):036106. [doi: 10.1103/PhysRevE.76.036106]

[11] Leung IXY, Hui P, Lio P, Crowcroft J. Towards real-time community detection in large networks. Physical Review E, 2009,79(6): 066107. [doi: 10.1103/PhysRevE.79.066107]

[12] Perozzi B, Al-Rfou R, Skiena S. Deepwalk: Online learning of social representations. In: Proc. of the 20th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. ACM Press, 2014. 701−710. [doi: 10.1145/2623330.2623732]

[13] Hartigan JA, Wong MA. Algorithm AS 136: A *k*-means clustering algorithm. Journal of the Royal Statistical Society. Series C (Applied Statistics), 1979,28(1):100−108. [doi: 10.2307/2346830]

[14] Tang L, Liu H. Community detection and mining in social media. Synthesis Lectures on Data Mining and Knowledge Discovery, 2010,2(1):1−137.

[15] Gan WY, He N, Li DY, Wang JM. Community discovery method in networks based on topological potential. Ruan Jian Xue Bao/ Journal of Software, 2009,20(8):2241−2254 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3318.htm [doi: 10.3724/SP.J.1001.2009.03318]

[16] Yang N, Gong DZ, Li X, Meng XF. Survey of communicaties identification. Journal of Computer Research and Development, 2005, 42(3):439−447 (in Chinese with English abstract).

[17] Wang M, Wang C, Yu JX, Zhang J. Community detection in social networks: An in-depth benchmarking study with a procedure-oriented framework. Proc. of the VLDB Endowment, 2015,8(10):998−1009. [doi: 10.14778/2794367.2794370]

[18] Newman MEJ. Fast algorithm for detecting community structure in networks. Physical Review E, 2004,69(6):066133. [doi: 10.1103/PhysRevE.69.066133]

[19] Clauset A, Newman MEJ, Moore C. Finding community structure in very large networks. Physical Review E, 2004,70(6):066111. [doi: 10.1103/PhysRevE.70.066111]

[20] Tsourakakis C, Bonchi F, Gionis A, Gullo F, Tsiarli M. Denser than the densest subgraph: Extracting optimal quasi-cliques with quality guarantees. In: Proc. of the 19th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. ACM Press, 2013. 104−112. [doi: 10.1145/2487575.2487645]

[21] Xie J, Szymanski BK, Liu X. Slpa: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process. In: Proc. of the 2011 IEEE 11th Int'l Conf. on Data Mining Workshops. IEEE, 2011. 344−349. [doi: 10.1109/ICDMW.2011.154]

[22] Belkin M, Niyogi P. Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation, 2003,15(6): 1373−1396. [doi: 10.1162/089976603321780317]

[23] Cao S, Lu W, Xu Q. Grarep: Learning graph representations with global structural information. In: Proc. of the 24th ACM Int'l Conf. on Information and Knowledge Management. ACM Press, 2015. 891−900. [doi: 10.1145/2806416.2806512]

[24] Schmidhuber J. Deep learning in neural networks: An overview. Neural Networks, 2015,61:85−117. [doi: 10.1016/j.neunet.2014.09. 003]

[25] Bengio Y, Courville A, Vincent P. Representation learning: A review and new perspectives. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2013,35(8):1798−1828. [doi: 10.1109/TPAMI.2013.50]

[26] Deng L. A tutorial survey of architectures, algorithms, and applications for deep learning. APSIPA Trans. on Signal and Information Processing, 2014,3:e2. [doi: 10.1017/atsip.2013.9]

[27] Zhang X, Gao Y. Face recognition across pose: A review. Pattern Recognition, 2009,42(11):2876−2896. [doi: 10.1016/j.patcog. 2009.04.017]

[28] Bengio Y. Learning deep architectures for AI. Foundations and Trends in Machine Learning, 2009,2(1):1−127. [doi: 10.1561/2200 000006]

[29] Deng L, Yu D. Deep learning: Methods and applications. Foundations and Trends in Signal Processing, 2014,7(3-4):197−387. [doi: 10.1561/2000000039]
[30] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature, 2015,521(7553):436−444. [doi: 10.1038/nature14539]
[31] Le QV, Ngiam J, Coates A, Lahiri A, Prochnow BY, Ng AY. On optimization methods for deep learning. In: Proc. of the 28th Int'l Conf. on Machine Learning (ICML-11). 2011. 265−272.
[32] Salakhutdinov R, Hinton G. Deep Boltzmann Machines. In: Proc. of the 12th Int'l Conf. on Artificial Intelligence and Statistics (AISTATS). 2009.
[33] Ranzato MA, Boureau YL, LeCun Y. Sparse feature learning for deep belief networks. Advances in neural information processing systems, 2008. 1185−1192.
[34] Matsugu M, Mori K, Mitari Y, Kaneda Y. Subject independent facial expression recognition with robust face detection using a convolutional neural network. Neural Networks, 2003,16(5):555−559. [doi: 10.1016/S0893-6080(03)00115-1]
[35] Mikolov T, Karafiát M, Burget L, Khudanpur S. Recurrent neural network based language model. Interspeech, 2010,2:3.
[36] Funahashi K, Nakamura Y. Approximation of dynamical systems by continuous time recurrent neural networks. Neural Networks, 1993,6(6):801−806. [doi: 10.1016/S0893-6080(05)80125-X]
[37] Hinton GE, Osindero S, Teh YW. A fast learning algorithm for deep belief nets. Neural Computation, 2006,18(7):1527−1554. [doi: 10.1162/neco.2006.18.7.1527]
[38] Bengio Y, Lamblin P, Popovici D, Popovici D, Larochelle H. Greedy layer-wise training of deep networks. Advances in Neural Information Processing Systems, 2007,19:153.
[39] Lee H, Grosse R, Ranganath R, Ng AY. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In: Proc. of the 26th Annual Int'l Conf. on Machine Learning. ACM, 2009. 609−616. [doi: 10.1145/1553374.1553453]
[40] LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-Based learning applied to document recognition. Proc. of the IEEE, 1998, 86(11):2278−2324. [doi: 10.1109/5.726791]
[41] Shin HC, Orton MR, Collins DJ, Doran S, Leach M. Stacked autoencoders for unsupervised feature learning and multiple organ detection in a pilot study using 4D patient data. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2013,35(8):1930−1943. [doi: 10.1109/TPAMI.2012.277]
[42] Khorasgani RR, Chen J, Zaïane OR. Top leaders community detection approach in information networks. In: Proc. of the 4th SNA-KDD Workshop on Social Network Mining and Analysis. 2010.
[43] Yang J, Leskovec J. Defining and evaluating network communities based on ground-truth. Knowledge and Information Systems, 2015,42(1):181−213. [doi: 10.1007/s10115-013-0693-z]
[44] Danon L, Diaz-Guilera A, Duch J, Arenas A. Comparing community structure identification. Journal of Statistical Mechanics: Theory and Experiment, 2005,2005(9):P09008. [doi: 10.1088/1742-5468/2005/09/P09008]

附中文参考文献:

[2] 黄发良,张师超,朱晓峰.基于多目标优化的网络社区发现方法.软件学报,2013,24(9):2062−2077. http://www.jos.org.cn/1000-9825/4400.htm [doi: 10.3724/SP.J.1001.2013.04400]
[6] 周小平,梁循,张海燕.基于 R-C 模型的微博用户社区发现.软件学报,2014,25(12):2808−2823. http://www.jos.org.cn/1000-9825/4720.htm [doi: 10.13328/j.cnki.jos.004720]
[15] 淦文燕,赫南,李德毅,王建民.一种基于拓扑势的网络社区发现方法.软件学报,2009,20(8):2241−2254. http://www.jos.org.cn/1000-9825/3318.htm [doi: 10.3724/SP.J.1001.2009.03318]
[16] 杨楠,弓丹志,李忱,孟小峰.Web 社区发现技术综述.计算机研究与发展,2005,42(3):439−447.

尚敬文(1993－),女,山东济南人,硕士,主要研究领域为社交网络,社区发现.

辛欣(1990－),女,硕士,主要研究领域为机器学习,数据挖掘,社交网络,深度学习.

王朝坤(1976－),男,博士,副教授,博士生导师,CCF 会员,主要研究领域为图和社交数据管理,音乐计算,大数据系统.

应翔(1992－),男,硕士,主要研究领域为社区发现.