

一种基于深度 RTRBM 的动态网络链路预测方法

潘嘉琪, 邹俊韬

(南京航空航天大学 计算机科学与技术学院, 江苏 南京 211106)

摘要:针对节点对的嵌入特征随时间演化而发生的骤变问题,提出了一种基于深度循环时序受限玻尔兹曼机(RTRBM)的链路预测方法。在样本集构建方面,利用网络嵌入学习自动化提取网络节点特征,并以嵌入特征空间中两个节点间的距离作为节点对样本属性;在学习模型选择方面,将 RTRBM 模型应用于动态网络链路预测,考虑到短时间段内节点在嵌入特征空间中的位置相对稳定,对 RTRBM 的能量函数及训练过程进行了改进。此外,为了提取节点对的深度时序特征,结合深度学习理论,通过纵向地堆叠多个改进后的 RTRBM 构成深度学习结构,并利用 Logistic 回归分类器对动态网络中的链路关系进行分类和预测。实验结果表明,改进后的 RTRBM 及其深度学习模型相比于其他方法在 AUC 指标下有着明显的性能提升。

关键词:动态网络;链路预测;网络嵌入;受限玻尔兹曼机

中图分类号:TP311

文献标识码:A

文章编号:1673-629X(2020)03-0001-06

doi:10.3969/j.issn.1673-629X.2020.03.001

A Dynamic Link Prediction Method Based on Deep Recurrent Temporal Restricted Boltzmann Machine

PAN Jia-qi, ZOU Jun-tao

(School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics,
Nanjing 211106, China)

Abstract: Aiming at the abrupt change of embedded features of node pairs with time evolution, a temporal link prediction method based on deep recurrent temporal restricted Boltzmann machine (RTRBM) is proposed. In the aspect of sample set construction, network embedding is used to extract node features over networks, and the distance between node pairs is taken as sample attributes. In terms of learning model selection, RTRBM is applied to temporal link prediction. Considering the position of the node in the embedded features space is relatively stable in a short time interval, the energy function and training process of the RTRBM are improved. In addition, In order to extract the depth-series features of node pairs, the deep learning theory is combined to form a deep leaning structure by stacking multiple improved RTRBM vertically, and a logistic regression classifier is utilized finally to classify and predict link relationships in dynamic networks. The experimental results show that the improved RTRBM and its deep learning model have significant performance improvement under the AUC index compared with other methods.

Key words: Temporal Network; Link Prediction; Network embedding; Restricted Boltzmann Machine

0 引言

链路预测不仅能够帮助人们挖掘网络内部的结构模式和分析网络动态演化的规律,而且还在学术推荐、犯罪治安监控、基因交互研究等诸多领域中体现出了实际的应用价值。

Sarkar 等人^[1]将非参数方法运用到动态网络的链路预测问题中,并使用局部敏感哈希加速了贝叶斯推断过程。Zhai 和 Zhang^[2]试图通过结合矩阵分解和自

动编码器来处理链路预测问题,并同时利用 Dropout 技术来进行模型的训练以防止过拟合现象。Zhu 等人^[3]通过矩阵分解来对动态网络进行链路预测,并且通过理论分析,提出了一种局部块坐标下降的优化策略。Zhang 等人^[4]提出了一种基于卷积神经网络的链路预测模型,该模型首先提取目标节点对的 k 步局部子图,然后再通过标签传播算法对子图内的每个节点进行哈希编码,最后将编码后的特征用于卷积神经网络

收稿日期:2019-02-28

修回日期:2019-06-28

网络出版时间:2019-11-07

基金项目:“十三五”重点基础科研项目(JCKY2016206B001);“十三五”装备预研项目(41401010201)

作者简介:潘嘉琪(1993-),男,硕士研究生,CCF 会员(88963G),研究方向为网络数据挖掘。

网络出版地址: <http://kns.cnki.net/kcms/detail/61.1450.tp.20191107.0910.030.html>

所有 RBM 都可以看成是相互独立的,因此,对于每个 RBM 而言,可以单独地执行分块 Gibbs 采样来近似求解模型参数的梯度。

对于一般的复杂网络,虽然网络内部的节点数量较多,但是节点对之间的连接却很少,这一方面表现出了网络的稀疏性,另一方面反映了除非特殊事件或异常情况的发生,大部分节点在网络中的状态应该基本保持不变,也就是说随着时间的推移,大部分节点对应的特征向量在潜层空间中所处的位置相对保持稳定^[9]。基于这一假设,文中在 RTRBM 的能量函数中加入了对于时序样本的平滑处理,即

$$E(\{v_t, h_t\}_{t=1}^T | \{r_t\}_{t=1}^{T-1}) = - \sum_{t=1}^T (h_t^T W v_t + b^T v_t + c^T h_t) - \sum_{t=2}^T h_t^T U r_{t-1} + \frac{\beta}{2} \sum_{t=2}^T \|r_t - r_{t-1}\|_2^2 \quad (8)$$

其中, β 为惩罚项系数, $\sum_{t=2}^T \|r_t - r_{t-1}\|_2^2$ 为惩罚项,用于防止样本在时间链上产生较大的输出波动,从而对模型的性能造成影响。

1.2 RTRBM 模型的学习过程

RTRBM 模型的学习过程涉及到输入数据的对数似然概率 $\log p(v_1, v_2, \dots, v_T)$ 关于模型参数 θ 求梯度的过程,而在使用 CD 算法求解模型该近似梯度时,又涉及到了式(8)中能量函数关于参数 θ 求梯度的过程。为了方便描述具体的梯度推导过程,文中将式(8)中的能量函数改写为 $E = -H - Q_2 - \beta \cdot L$ 的形式,其中

$$\begin{cases} H = \sum_{t=1}^T (h_t^T W v_t + b^T v_t + c^T h_t) \\ Q_2 = \sum_{t=2}^T h_t^T U r_{t-1} \\ L = \frac{1}{2} \sum_{t=2}^T \|r_t - r_{t-1}\|_2^2 \end{cases} \quad (9)$$

基于分块 K-CD 算法的 RTRBM 模型参数 θ 的梯度为:

$$\begin{aligned} \Delta\theta = & E_{h_1, \dots, h_T, v_1, \dots, v_T, r_1, \dots, r_{T-1}} [-\nabla_{\theta} E] - \\ & E_{v_1, \dots, v_T, h_1, \dots, h_T, r_1, \dots, r_{T-1}} [-\nabla_{\theta} E] = \\ & \underbrace{\langle \nabla_{\theta} H \rangle_0 - \langle \nabla_{\theta} H \rangle_K}_{\Delta H} + \\ & \underbrace{\langle \nabla_{\theta} Q_2 \rangle_0 - \langle \nabla_{\theta} Q_2 \rangle_K}_{\Delta Q_2} + \\ & \underbrace{\langle \nabla_{\theta} L \rangle_0 - \langle \nabla_{\theta} L \rangle_K}_{\Delta L} \end{aligned} \quad (10)$$

(1) H 关于参数 θ 的梯度 $\nabla_{\theta} H$ 。

H 关于参数 θ 的梯度求解比较简单,就相当于在每个 RBM 上分别对参数 θ 进行求导,即

$$\nabla_w H = \sum_{t=1}^T h_t^T v_t, \nabla_b H = 0,$$

$$\nabla_b H = \sum_{t=1}^T v_t, \nabla_c H = \sum_{t=1}^T h_t \quad (11)$$

(2) Q_2 关于参数 θ 的梯度 $\nabla_{\theta} Q_2$ 。

由于 Q_2 关于参数 θ 的梯度都依赖于 r_t ,所以在计算梯度时,需要先计算 r_t 关于参数 θ 的梯度,而 r_t 关于参数 θ 的梯度可以通过时序反向传播算法 (back propagation through time, BPTT)^[10] 来进行递归式的求解。

令 $Q_t = \sum_{\tau=2}^T h_{\tau}^T U r_{\tau-1} = Q_{t+1} + h_t^T U r_{t-1}$, 且 $Q_{T+1} = 0$, 则根据链式法则有:

$$\nabla_{r_t} Q_{t+1} = U^T (\nabla_{r_{t+1}} Q_{t+2} \odot r_{t+1} \odot (1 - r_{t+1}) + h_{t+1}) \quad (12)$$

其中 \odot 为元素对乘积。由于 Q_{t+1} 不是关于 r_1, r_2, \dots, r_{t-1} 的函数,故有 $\nabla_{r_t} Q_2 = \nabla_{r_t} Q_{t+1}$, 因此 Q_2 关于 $r_t (t \geq 1)$ 的偏导数可以通过递归的形式逐一求解。

同理, Q_2 关于参数 U 的偏导数可以根据链式法则递归求解,即

$$\nabla_U Q_2 = \sum_{t=2}^T [\nabla_{r_t} Q_{t+1} \odot r_t \odot (1 - r_t) + h_t] \cdot r_{t-1}^T \quad (13)$$

根据式(10)和式(13),可以求出 RTRBM 模型关于参数 U 的梯度。令 $D_t = \langle \nabla_{r_t} Q_{t+1} \rangle_0 - \langle \nabla_{r_t} Q_{t+1} \rangle_K$, $2 \leq t \leq T+1$, 且 $D_{T+1} = 0$, 则

$$\begin{aligned} \Delta_U Q_2 = & \sum_{t=2}^T [D_t \odot r_t \odot (1 - r_t) + \langle h_t \rangle_0 - \langle h_t \rangle_K] \cdot r_{t-1}^T \end{aligned} \quad (14)$$

其中, $\langle \nabla_{r_t} Q_{t+1} \rangle_0 = E_{h_1, \dots, h_T, v_1, \dots, v_T, r_1, \dots, r_{T-1}} [\nabla_{r_t} Q_{t+1}]$ 表示“数据分布”在 $\{r_t\}_{t=1}^{T-1}$ 条件下的期望, $\langle \nabla_{r_t} Q_{t+1} \rangle_K = E_{v_1, \dots, v_T, h_1, \dots, h_T, r_1, \dots, r_{T-1}} [\nabla_{r_t} Q_{t+1}]$ 表示执行 K 步分块 CD 算法后,“模型分布”在 $\{r_t\}_{t=1}^{T-1}$ 条件下的期望。

因为 $\nabla_{r_t} Q_{t+1}$ 是通过递归的方式求解的,故 D_t 也是以递归的形式求解的,即

$$D_t = U^T [D_{t+1} \odot r_{t+1} \odot (1 - r_{t+1}) + \langle h_{t+1} \rangle_0 - \langle h_{t+1} \rangle_K] \quad (15)$$

其中 $1 \leq t \leq T-1$ 。

同理, RTRBM 模型关于参数 W, b, c 的梯度分别为:

$$\nabla_w Q_2 = \sum_{t=2}^T [D_t \odot r_t \odot (1 - r_t)] \cdot v_t^T \quad (16)$$

$$\nabla_b Q_2 = 0 \quad (17)$$

$$\nabla_c Q_2 = \sum_{t=2}^T [D_t \odot r_t \odot (1 - r_t)] \quad (18)$$

(3) L 关于参数 θ 的梯度 $\nabla_{\theta} L$ 。

L 关于参数 θ 的梯度需要先计算 L 关于 r_t 的梯度,其计算过程可以根据链式法则进行求解,即:

$$\nabla_{r_t} L = \frac{\partial L}{\partial r_t} + U^T \left[\frac{\partial L}{\partial r_{t+1}} \odot r_{t+1} \odot (1 - r_{t+1}) \right] \quad (19)$$

而

$$\frac{\partial L}{\partial r_t} = \begin{cases} r_t - r_{t+1}, & t = 1 \\ 2r_t - r_{t-1} - r_{t+1}, & 2 \leq t \leq T-2 \\ r_t - r_{t-1}, & t = T-1 \end{cases} \quad (20)$$

其中 $r_T = 0$ 。所以 L 关于参数 W, U, b, c 的梯度分别为:

$$\nabla_W L = \sum_{t=1}^{T-1} [\nabla_{r_t} L \odot r_t \odot (1 - r_t)] \cdot v_t^T \quad (21)$$

$$\nabla_U L = \sum_{t=2}^{T-1} [\nabla_{r_t} L \odot r_t \odot (1 - r_t)] \cdot r_{t-1}^T \quad (22)$$

$$\nabla_b L = 0 \quad (23)$$

$$\nabla_c L = \sum_{t=1}^{T-1} \nabla_{r_t} L \odot r_t \odot (1 - r_t) \quad (24)$$

RTRBM 具体的实现如算法 1 所示:

算法 1: 基于分块 CD-K 的 RTRBM 训练流程。

输入: 时序样本数据 v_1, v_2, \dots, v_T , 最大训练迭代次数 \maxIter , 学习速率 η , CD 算法执行步长 K , 惩罚项系数 β , 以及每个 RBM 共享的模型参数: 可见层神经元个数 n_{visible} , 隐含层神经元个数 n_{hidden}

输出: RTRBM 模型的最优参数 $\theta^* = \{W^*, U^*, b^*, c^*\}$

1. 随机初始化模型参数 W, U, b, c ;
2. for iter = 1 to \maxIter do
3. 根据式(3)计算出第 t 个 RBM 在时间链上的输出期望 r_t ;
4. for $t = 1$ to T do
5. for $k = 1$ to K do
6. 根据式(6)计算出第 t 个 RBM 隐含层的采样概率并采样 $h_{t,j}^{(k)} \sim p(h_{t,j} = 1 | v_{t,j}^{(k)}, r_{t-1})$
7. 根据式(7)计算出第 t 个 RBM 可见层的采样概率并采样 $v_{t,i}^{(k+1)} \sim p(v_{t,i} = 1 | h_{t,i}^{(k)})$
8. end for
9. 根据式(15)计算 $D_t (2 \leq t \leq T)$
10. end for
11. 根据式(11)计算出 H 关于参数 W, U, b, c 的梯度 $\nabla_W H, \nabla_U H, \nabla_b H, \nabla_c H$
12. 根据式(14)、(16)、(17)和(18)计算模型在 Q_2 上关于参数 W, U, b, c 的梯度 $\Delta_W Q_2, \Delta_U Q_2, \Delta_b Q_2, \Delta_c Q_2$
13. 根据式(19)和(20)计算 $\nabla_{r_t} L (1 \leq t \leq T-1)$
14. 根据式(21)~式(24)计算 L 关于参数 W, U, b, c 的梯度 $\nabla_W L, \nabla_U L, \nabla_b L, \nabla_c L$
15. 根据式(10)计算参数的梯度 $\Delta\theta$
16. 利用随机梯度下降法更新参数 $\theta^{(\text{iter}+1)} = \theta^{(\text{iter})} + \eta \cdot \Delta\theta$
17. end for

2 基于深度 RTRBM 的动态网络链路预测方法

2.1 基于深度 RTRBM 的动态网络链路预测模型框架

如图 2 所示, 基于深度 RTRBM 模型的动态网络链路预测框架总共分为数据预处理和模型训练两个部分。数据预处理部分主要负责样本集的构建和训练测试集的划分, 而模型训练部分则主要负责训练深度 RTRBM 模型。具体的预测流程为:

步骤 1: 对原始网络数据进行预处理得到时序样本集, 其中样本属性由 $t_1 \sim t_{T-1}$ 时刻的节点对特征组成, 样本标签由节点对在 t_T 时刻的状态构成, 有连边则为 1, 否则为 0; 然后再采用随机抽样或十折交叉验证法对样本集进行划分, 得到训练集 trainSet 和测试集 testSet ;

步骤 2: 将训练集 trainSet 输入至深度 RTRBM 模型, 通过横向 BPTT 算法逐层训练, 获得第 n 层的隐含特征;

步骤 3: 将第 n 层的隐含特征作为 Logistic 回归模型的输入, 通过随机梯度下降和反向传播 (back propagation, BP) 对第 n 层的模型参数进行微调;

步骤 4: 将测试集 testSet 输入至已训练好的预测模型, 得出最后的预测结果。

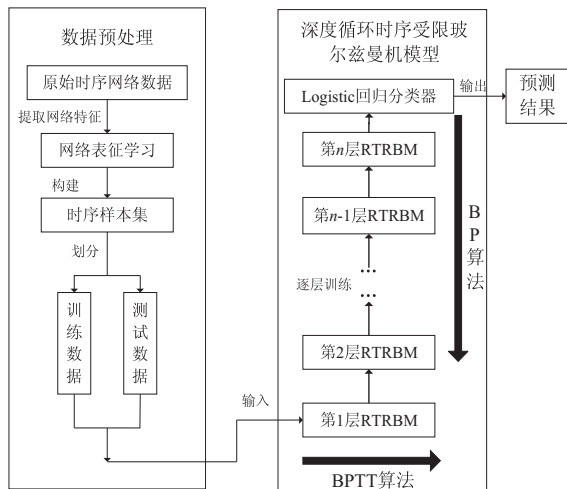


图 2 基于深度 RTRBM 的动态网络链路预测框架

2.2 深度 RTRBM 模型结构

图 3 所示的是一个具有两层结构的深度 RTRBM 模型, 该模型采用贪婪式的训练方法, 它将第 $n-1$ 层的隐含特征 $h_1^{(n-1)}, h_2^{(n-1)}, \dots, h_r^{(n-1)}$ 作为第 n 层 RTRBM 的输入, 经训练得到深度 RTRBM 模型在第 n 层上的隐含特征 $h_1^{(n)}, h_2^{(n)}, \dots, h_r^{(n)}$, 直到深度 RTRBM 模型的最后一层 RTRBM 训练完毕为止。

假设每个时序样本含有 T 个时间快照, 深度 RTRBM 模型在第 d 层上的输入输出维度分别为 $N_v^{(d)}$

和 $N_h^{(d)}$, 深度为 D , 分块 CD 算法执行步长为 K , 训练周期为 I , 则在贪婪训练方式下总共所需要的时间复杂度为 $O(\sum_{d=1}^D 2I((K+1)N_v^{(d)}N_h^{(d)} + TN_v^{(d)}N_h^{(d)3}))$ 。

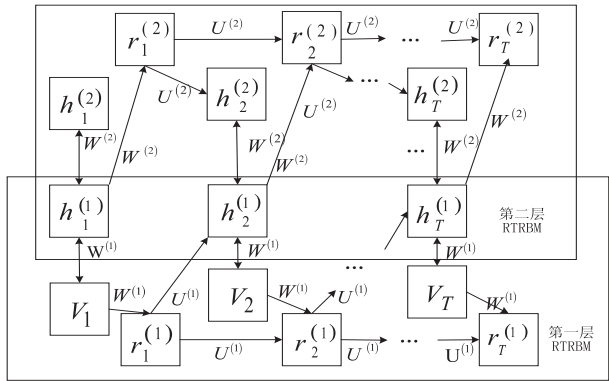


图3 两层 RTRBM 结构示意图

2.3 时序样本集的构建

给定一组网络的时间快照 $\{G_1, G_2, \dots, G_T\}$, 文中首先利用 Node2Vec 算法^[11]提取各个网络快照状态下的节点特征, 然后在此基础上构建节点对的特征, 并且为了保证样本集拥有足够的区分度, 还添加了 CN、RA、AA 相似性度量指标得分作为扩充属性集, 最后将每个时间快照的样本数据合并为最终的时序样本集。

3 实验仿真与结果分析

3.1 实验设置

3.1.1 数据集

实验部分使用的数据集全部来源于 Koblenz Network Collection (<http://konect.uni-koblenz.de/>) 和 Stanford Large Network Dataset (<http://snap.stanford.edu/data/>), 其中包含 2 个人际关系网络和 5 个邮件传递网络, 其相关统计信息如表 1 所示。

表1 动态网络的相关统计信息

数据集	节点数	边数	时间样本个数
Email	986	332 334	5
Email-D1	309	61 046	5
Email-D2	162	46 772	5
Email-D3	89	12 216	5
Email-D4	142	48 141	5
Haggle	274	28 244	4
Infect	410	17 298	5

3.1.2 对比算法

为了验证提出方法的有效性, 使用两种动态网络链路预测方法与之进行对比实验, 即基于相似性度量的方法和基于深度学习的方法。

(1) 基于相似性度量的方法。

对于给定的一组网络时间快照序列 $\{G_1, G_2, \dots, G_T\}$, 基于相似性度量的方法首先将前 $T-1$ 个网络快照 $\{G_1, G_2, \dots, G_{T-1}\}$ 压缩成对应的概念图 $G_{1,T-1}$, 然后在 $G_{1,T-1}$ 上做静态网络链路预测, 预测 G_T 的链路情况^[12]。 $G_{1,T-1}$ 的邻接矩阵 $A_{1,T-1}$ 定义如下:

A_{1,T-1} = \begin{cases} 1, & \text{if } \exists k \in [1, T-1], A_k(i, j) = 1 \\ 0, & \text{otherwise} \end{cases} \quad (25)

其中 A_k 为图 G_k 的邻接矩阵。文中采用的度量指标为 CN、RA 和 AA。

(2) 基于深度学习的方法。

基于深度学习的方法主要区别在于所使用的学习模型, 文中采用的学习模型有: 条件时序受限玻尔兹曼机 (conditional temporal Restricted Boltzmann Machine, ctRBM)^[13] 和深度条件置信网 (conditional Deep Belief Network, cDBN)^[14]。

3.1.3 超参数设置

在模型参数的设置上, RBM 作为 DRTRBM 和 cDBN 的基本单元, 其权重矩阵 W 采用文献[15]中的方式进行随机初始化, 学习速率 η 设为 0.01, CD 算法的执行步长 K 设为 1, 最大训练次数 maxIter 设为 100。另外, DRTRBM 的惩罚项系数 β 设为 0.001。

在实验结果上, 文中采用十折交叉验证法, 对每个网络分别重复实验 30 次, 并取平均值作为最终的实验结果。

3.2 实验结果

文中使用 AUC 指标来评估各个算法在不同数据集上的预测性能, 实验结果如图 4 所示。

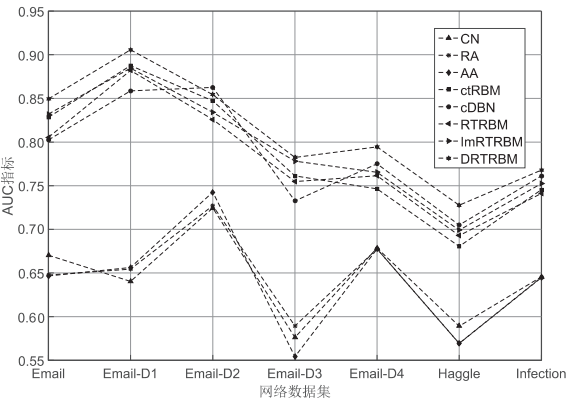


图4 算法关于动态网络链路预测的 AUC 得分

对于 Email 数据集, DRTRBM 的平均 AUC 得分最高, 相对于 CN、RA、AA、ctRBM、cDBN 和 RTRBM 分别将基准线提高了 26.70%、31.14%、31.36%、2.54%、5.84% 和 5.46%。此外, 改进后的 RTRBM (ImRTRBM) 性能比未改进的 RTRBM 提高了 3.32%, 说明 Email 网络在短时间间隔内节点对之间的关联相对较为稳定。DRTRBM 的平均 AUC 得分比 ImRTRBM 提高了近 2.08%, 说明通过堆叠多个 RTRBM 提取动态网络的深

度时序特征能够有效地提高链路预测的精度。

对于 Email-D1 数据集, ctRBM、cDBN、RTRBM、ImRTRBM 和 DRTRBM 的平均 AUC 得分都在 0.85 以上, 而 CN、RA 和 AA 的平均 AUC 得分都在 0.65 左右。其中 DRTRBM 的平均 AUC 得分最高, 相比于 ctRBM、cDBN、RTRBM 和 ImRTRBM 分别将基准线提高了 2.04%、5.49%、2.68% 和 2.39%。ImRTRBM 的平均 AUC 得分比 RTRBM 只提高了 0.28%, 说明对于 RTRBM 的时间平滑处理效果并不是很明显。另外, ctRBM 的平均 AUC 得分次高, 说明网络前一时刻的状态对于后一时刻的状态影响很大。

此外, 在 Email-D2 ~ D4、Haggle、Infection 数据集上, ImRTRBM 和 DRTRBM 的平均 AUC 得分都比其他算法要高。

总的来说, 基于机器学习的方法在 AUC 指标上均比基于相似性度量的方法要高; 其次, 文中提出的基于深度 RTRBM 的动态网络链路预测方法不仅可以有效地提取动态网络的深度时序特征, 而且可以处理节点对特征随时间演化而发生的骤变问题, 进而提高链路预测的准确性。

4 结束语

将 RTRBM 应用于动态网络的链路预测, 对 RTRBM 模型的能量函数和训练过程进行了改进。在所提出的动态网络链路预测框架中, 首先利用 Node2Vec 算法提取网络节点的嵌入特征, 并基于该特征构建用于后续学习分类的时序样本集, 然后结合深度学习理论将多个改进后的 RTRBM 纵向堆叠以构成深度学习结构来提取网络的深度时序特征, 最后结合 Logistic 回归分类器对动态网络中的链路进行分类和预测。实验结果表明, 该方法相比于其他方法有着明显的性能提升。然而, 深度 RTRBM 模型中的超参数需要根据网络类型和规模而决定, 因此如何高效地选择模型的超参数是今后要研究的主要问题。

参考文献:

- [1] SARKAR P, CHAKRABARTI D, JORDAN M I. Nonparametric link prediction in dynamic networks[C]//Proceedings of the 29th international conference on machine learning. Edinburgh, Scotland: Omnipress, 2012: 1897–1904.
- [2] ZHAI S, ZHANG Z. Dropout training of matrix factorization and autoencoder for link prediction in sparse graphs[C]//Proceedings of the 2015 international conference on data mining. Vancouver, BC, Canada: SIAM, 2015: 451–459.
- [3] ZHU L, GUO D, YIN J, et al. Scalable temporal latent space inference for link prediction in dynamic social networks[J]. IEEE Transactions on Knowledge & Data Engineering, 2016, 28(10): 2765–2777.
- [4] ZHANG M, CHEN Y. Link prediction based on graph neural networks[J]. CoRR, 2018, abs/1802.09691: 5171–5181.
- [5] KEIKHA M M, RAHGOZAR M, ASADPOUR M. DeepLink: a novel link prediction framework based on deep learning[J]. CoRR, 2018, abs/1807.10494: 1–19.
- [6] ZHANG C, ZHANG H, YUAN D, et al. Deep learning based link prediction with social pattern and external attribute knowledge in bibliographic networks[C]//2016 IEEE international conference on internet of things (iThings) and IEEE green computing and communications (GreenCom) and IEEE cyber, physical and social computing (CPSCom) and IEEE smart data (SmartData). Chengdu: IEEE, 2016: 815–821.
- [7] BOULANGER-LEWANDOWSKI N, BENGIO Y, VINCENT P. Modeling temporal dependencies in high-dimensional sequences: application to polyphonic music generation and transcription[J]. Chemistry A European Journal, 2012, 18(13): 3981–3991.
- [8] 张娟, 杨建功, 汪西莉. 条件深度玻尔兹曼机人脸图像分割模型[J]. 小型微型计算机系统, 2017, 38(5): 1130–1133.
- [9] LI T, ZHANG J, YUP S, et al. Deep dynamic network embedding for link prediction[J]. IEEE Access, 2018, 6: 29219–29230.
- [10] RUMELHART D, MCCLELLAND J. Learning internal representations by error propagation[J]. Readings in Cognitive Science, 1988, 323(6088): 399–421.
- [11] GROVER A, LESKOVEC J. node2vec: scalable feature learning for networks[C]//Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. San Francisco, California, USA: ACM, 2016: 855–864.
- [12] HUANG Z, LIN D K J. The time-series link prediction problem with applications in communication surveillance[J]. Journal on Computing, 2009, 21(2): 286–303.
- [13] LI X, DU N, LI H, et al. A deep learning approach to link prediction in dynamic network[C]//Proceedings of 2014 international conference on data mining. Philadelphia, Pennsylvania, USA: SIAM, 2014: 289–297.
- [14] 周风余, 尹建芹, 杨阳, 等. 基于时序深度置信网络的在线人体动作识别[J]. 自动化学报, 2016, 42(7): 1030–1039.
- [15] GLOROT X, BENGIO Y. Understanding the difficulty of training deep feedforward neural networks[C]//International conference on artificial intelligence and statistics. Chia Laguna Resort, Sardinia, Italy: JMLR, 2010: 249–256.