论文阅读笔记

1. 论文信息

论文名：node2vec Scalable Feature Learning for Networks
作者：Grover, Leskovec
年份：2016

2. 概述

网络预测任务一般是预测网络的节点和边。在一个节点分类任务中，预测节点的最可能的标签。比如在社交网络中预测用户兴趣爱好，在蛋白质网络中预测蛋白质的功能标签，在连接预测中预测一对节点是否会有边相连等。连接预测可用在发现基因互动，发现社交网络中的朋友关系。

进行网络预测需要构建特征向量来代表结点和边。其中一个方法是通过解决一个优化问题来学习特征。特征学习的困难在于定义目标函数，这涉及到计算复杂度与预测精度之间的平衡。但现在的技术还无法定义和最优化一个合理的目标以进行大规模的无监督特征学习。基于线性 or 非线性降维技术的传统算法比如 PCA（Principal Component Analysis）、多维缩放法（Multi-Dimensional Scaling）及其拓展，最优化得出一个有代表性的数据矩阵。因此这些算法总是涉及数据矩阵的特征分解，这对于大规模真是网络来说开销太大且效果还不好。

本文设计一个目标以保持节点的局部邻居信息，这个目标可以通过使用 SGD（随机梯度下降）以进行单隐藏层的 BPNN 来进行最优化。网络节点可以根据他们所属的社区（同质性 homophily）或节点的结构性作用（结构相同 structural equivalence）被组织起来。

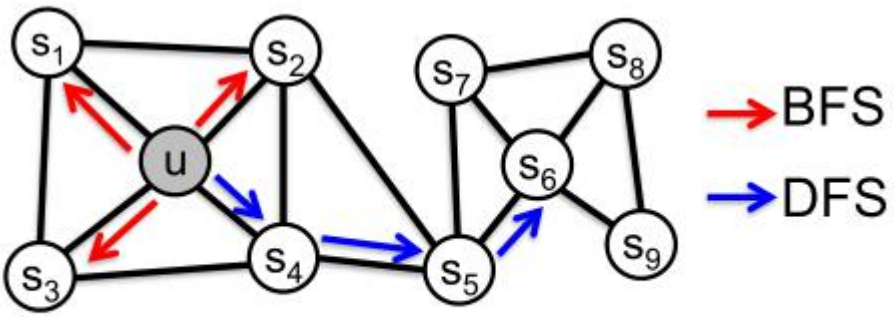


Figure 1: BFS and DFS search strategies from node u ($k = 3$).

比如 u 与 S1 属于同一个社区，u 与 S6 在各自的点集有相同的结构作用（中心点）。真实网络一般是以上两种相似的混合。本文设计的算法根据以下两个原则学习节点特征：既能使在同一个紧密相连的网络中的节点相似，又能使起相似结构作用的节点相似。这使得该特征学习算法在不同领域和预测任务间都能通用。

本文提出的 node2vec 算法是适用于大规模网络特征学习的半监督算法，使用了 nlp 领域上的 SGD（随机梯度下降）来最优化目标函数，得到的特征可以最大程度地在一个 d 维特征空间中保存节点的网络近邻信息，还设计了 biased random walk 方法生成节点的网络近邻序列。

总的来说，贡献如下：

1、提出 node2vec，一个高效、可规模化的网络特征学习算法，能通过 SGD 最优化目标（保存近邻信息）。

2、展示了 node2vec 与当前网络科学领域的已有原则相一致，且在探索网络特征时更具灵活性。

3、将 node2vec 特征学习方法，从节点拓展到节点对，用以进行基于边的预测任务。

4、在几个真实数据集上测试了 node2vec 在多元分类和连接预测任务上的效果。

3.模型

1）特征学习方法

把网络的特征学习看做最大似然最优化问题，

$G = (V, E)$ 代表网络，适用于所有的有向 or 无向，带权 or 不带权图；

$f : V \to \mathbb{R}^d$ 代表从节点到特征向量之间的映射函数，是需要学习的目标，用于下游预测任务，f 是一个|V|*d 大小的矩阵，其中 d 代表特征向量维度；

$N_S(u) \subset V$ 代表源点 u 通过采样策略 S 所得的网络近邻节点序列。

使用 Skip-gram 架构，需要最优化的目标函数为：

$$\max_f \sum_{u \in V} \log Pr(N_S(u)|f(u)).$$

（等式 1）

两个假设：

1、条件独立性。给定源点的特征向量的情况下，观察某一个近邻点的出现可能性与其他近邻点出现的可能性独立。

$$Pr(N_S(u)|f(u)) = \prod_{n_i \in N_S(u)} Pr(n_i|f(u)).$$

2、特征空间对称性。源点及其近邻点在相互之间的特征空间上有对称效应。因此对每个源点-近邻节点对的条件似然，使用一个 softmax 单元，softmax 值参数是他们的特征向量的点乘：

$$Pr(n_i|f(u)) = \frac{\exp(f(n_i) \cdot f(u))}{\sum_{v \in V} \exp(f(v) \cdot f(u))}.$$

所有比值之和能保证总和为 1

由上述两个假设可将目标函数简化为：

$$\max_f \sum_{u \in V} \left[-\log Z_u + \sum_{n_i \in N_S(u)} f(n_i) \cdot f(u) \right].$$

（等式 2）

$$Z_u = \sum_{v \in V} \exp(f(u) \cdot f(v))$$

Zu 的计算在大规模网络开销很大，因此要用负采样。
最后使用 SGD（随机梯度下降法）最优化模型参数（即特征 f）。
以上与 word2vec 过程类似。

2）采样策略

基于 Skip-gram 架构的特征学习方法最初发展于 nlp 领域。由于文本的线性特点，近邻的概念很自然的被定义为一个滑动窗口中的一段连续的单词。但网络不是线性的，因此需要更丰富的近邻概念。因此本文提出了一个随机过程，根据给定的源节点 u 采样得出许多不同的近邻节点。近邻集合 Ns(u)中的节点不仅仅局限于邻接点，还可以因为采用策略 S 的不同而有不同。

规定 Ns 集合节点数为 k，对同一个节点进行近邻点采样。
预测任务一般关注的两类相似性：同质性（homophily）和结构相同性（structural equivalence）。
1、同质性：距离相近且同属于相似节点簇节点具有同质性。比如 u 和 s1 同属于一个社区。
2、结构相同性：在各自节点簇中起相似的结构性作用的节点具有结构相同性（不要求邻接）。比如 u 与 s6 都是各自社区的中心节点。

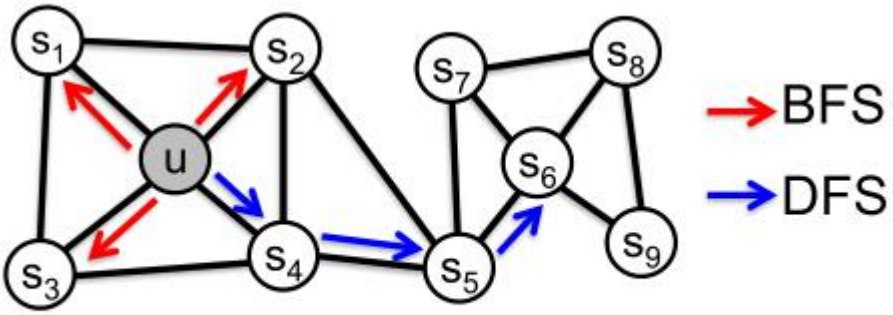


Figure 1: BFS and DFS search strategies from node u ($k = 3$).

经典采样方法：BFS、DFS

BFS 对应结构相同性，优点是准确描绘局部社区的特性，缺点是图都只被探索了很小一个部分。

DFS 对应同质性，优点所得近邻节点序列具有宏观性，缺点探索的过深会导致采样节点过远而不具代表性。

本文采样策略 biased random walk 综合了 BFS 和 DFS。

给定源点 u，模拟长度为 l 的 random walk 过程。ci 代表 walk 过程中的第 i 个节点，其中 c0=u。按以下规则选择 ci：

$$P(c_i = x \mid c_{i-1} = v) = \begin{cases} \frac{\pi_{vx}}{Z} & \text{if } (v, x) \in E \\ 0 & \text{otherwise} \end{cases}$$

PIvx 是节点 v 与 x 之间的非正则化转移概率，Z 是正则化常数。

最简单的 biased random walk 是根据固定的边权值 Wvx 来选下一个采样节点，比如直接使 PIvx=Wvx，但这不具有可解释性。为了兼顾结构相同性或同质性，本文提出的 biased random walk 有两个参数 p 和 q。考虑 random walk 遇到这种情况：刚从节点 t 经过 edge(t,v)到达节点 v，现在要决定下一个节点，因此需要考虑边 edges(v,x)的转移概率 PIvx，设

$$\pi_{vx} = \alpha_{pq}(t,x) \cdot w_{vx}$$

$$\alpha_{pq}(t,x) = \begin{cases} \frac{1}{p} & \text{if } d_{tx} = 0 \\ 1 & \text{if } d_{tx} = 1 \\ \frac{1}{q} & \text{if } d_{tx} = 2 \end{cases}$$

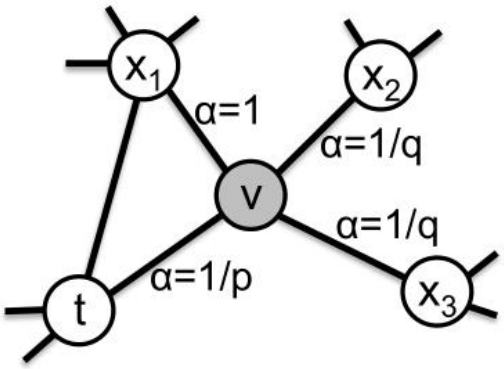


Figure 2: Illustration of the random walk procedure in *node2vec*. The walk just transitioned from t to v and is now evaluating its next step out of node v. Edge labels indicate search biases α.

dtx 代表结点 t 与 x 之间的最短路径距离，dtx 只能是{0,1,2}其中一个，
dtx=0，x=t
dtx=1，x 与 v、t 都邻接，即 x1
dtx=2，x 与 v 邻接，与 t 不邻接，即 x2，x3

Return parameter, p. 参数 p 控制返回上一个节点的可能性，当 p 越大时（p>max(q,1)），回到已访问节点的可能性越小（除非下一个节点没有其他邻接点，就只能返回上一个节点了）；当 p 越小时（p<min(q,1)），可能会回溯到上一个节点处，使得 walk 更具局部性（靠近源点）

In-out parameter, q. q 控制 walk 向里走或向外走，若 q>1，random walk 倾向于选接近 t 的节点，采样具有局部性（类似 BFS）。若 q<1，random walk 倾向于选远离 t 的节点（类似 DFS）。当 p=1，q=1 时，游走方式就等同于 DeepWalk 中的随机游走。

时间复杂度方面，每次进行一次长度为 l 的 random walk 时，可以得到 l-k 个节点的采样序列，其中采样序列长度为 k。因此采取每个样本的时间复杂度为 O(l/k*(l-k)) （因为这个过程共花费了 l 的时间，但采取了 k*(l-k)个样本）。

例如，l=6，k=3 时，采样得到{u,s4 ,s5 ,s6 ,s8 ,s9 }，则 Ns (u) = {s4 ,s5 ,s6 }, N s (s4) = {s5 ,s6 ,s8 } and Ns (s5) ={s6 ,s8 ,s9 }

3）node2vec 算法伪代码

Algorithm 1 The *node2vec* algorithm.

LearnFeatures (Graph $G = (V, E, W)$, Dimensions d, Walks per node r, Walk length l, Context size k, Return p, In-out q)
π = PreprocessModifiedWeights(G, p, q)
$G' = (V, E, \pi)$
Initialize $walks$ to Empty
for $iter = 1$ **to** r **do**
  for all nodes $u \in V$ **do**
    $walk$ = node2vecWalk(G', u, l)
    Append $walk$ to $walks$
f = StochasticGradientDescent(k, d, $walks$)
return f

node2vecWalk (Graph $G' = (V, E, \pi)$, Start node u, Length l)
Inititalize $walk$ to $[u]$
for $walk\_iter = 1$ **to** l **do**
  $curr = walk[-1]$
  V_{curr} = GetNeighbors($curr$, G')
  s = AliasSample(V_{curr}, π)
  Append s to $walk$
return $walk$

总共进行 r 次 random walk。对于 walk 的每一步，都根据转移概率 PIvx 进行采样。PIvx 可以预先计算，因此采样效率为 O(1)。node2vec 的三个阶段：预计算转移概率 PIvx、random walk、用 SGD 进行最优化按顺序进行。每个阶段可以异步、同时进行，因此 node2vec 适用于大规模数据集。

首先预计算 PI，即$\pi_{vx} = \alpha_{pq}(t, x) \cdot w_{vx}$。

然后开始 r 次 random walk，每次 walk 都要以每个节点为源节点进行 node2vecWalk，加入 walks 数组中，后面的 SGD 会用到。
对于每个 node2vecWalk，要从源节点 u 开始找 l 个节点，初始时数组 walk={u}，curr 代表前一个结点，Vcurr 代表当前节点，根据 AliasSample 采样取得下一个节点 s，将 s 加进数组 walk 中。最后返回 walk 加入 walks 数组中。
根据 walks 数组进行 SGD 最优化。

4）拓展到连接预测

node2vec 提供了网络节点特征学习的半监督方法。通过自扩展的方法可以将单节点的特征学习拓展到节点对。
给定节点 u 和 v，定义二元运算符 ◦ 使 g(u,v)=f(u)◦f(v)

$g : V \times V \rightarrow \mathbb{R}^{d'}$，d’ 代表 pair(u,v)的特征向量维度，运算符有以下选择：

Operator	Symbol	Definition
Average	$\boxplus$	$[f(u) \boxplus f(v)]_i = \frac{f_i(u)+f_i(v)}{2}$
Hadamard	$\boxdot$	$[f(u) \boxdot f(v)]_i = f_i(u) * f_i(v)$
Weighted-L1	$\|\cdot\|_{\bar{1}}$	$\|f(u) \cdot f(v)\|_{\bar{1}i} = \|f_i(u) - f_i(v)\|$
Weighted-L2	$\|\cdot\|_{\bar{2}}$	$\|f(u) \cdot f(v)\|_{\bar{2}i} = \|f_i(u) - f_i(v)\|^2$

4. 实验

1）小数据集样例可视化实验

取小说 Les Misérables 里的角色作节点，将同场景出现过的角色之间用线连接起来。共 77 个节点和 254 条边。设 d=16，为每个节点训练特征。使用 k-means 对特征进行聚类，然后二维可视化。

当 p=1，q=0.5 时，node2vec 把在各个情节中频繁互动的角色各归为了一个簇。表现出了角色的同质性。

当 p=1，q=2 时，node2vec 把起相同结构作用的节点归为一个簇，比如蓝色节点都是起着不同情节之间的沟通桥梁的作用。黄色节点代表边缘角色，与其他角色的互动很少。

这说明了 node2vec 适用于真实网络且能表现出同质性和结构相同性。

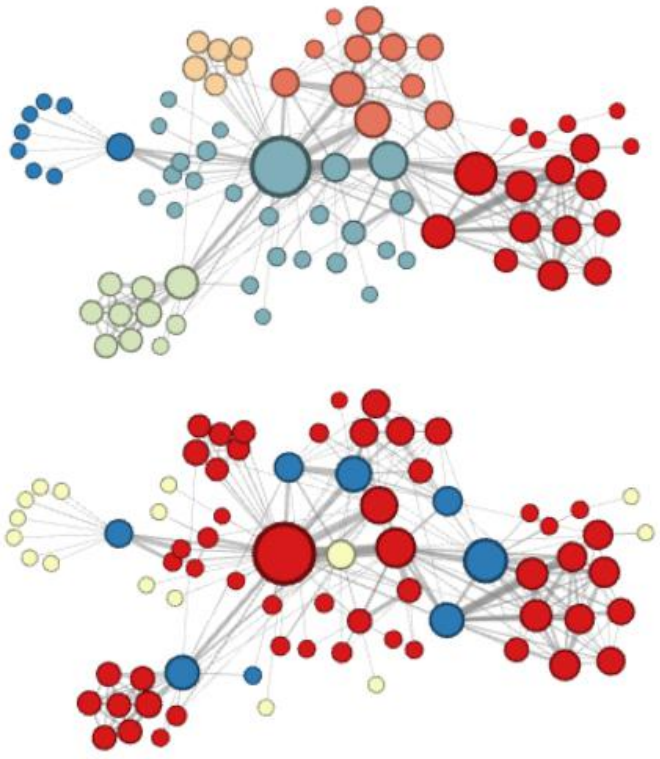


Figure 3: Complementary visualizations of Les Misérables coappearance network generated by *node2vec* with label colors reflecting homophily (top) and structural equivalence (bottom).

2）多元对比分类实验

参数设定：

d = 128,r = 10, l = 80, k = 10，最优化迭代一次，取 50%的有标签节点进行训练。通过十重交叉验证优化超参数， p,q ∈ {0.25,0.50,1,2,4}。

指标：

多元分类任务预测结果使用 Macro-F1 值评估，Macro-F1 即计算出每一个类的 Precison 和 Recall 后计算 F1，最后将 F1 平均。

数据集：BlogCatalog、PPI、Wikipedia

- BlogCatalog [38]: This is a network of social relationships of the bloggers listed on the BlogCatalog website. The labels represent blogger interests inferred through the metadata provided by the bloggers. The network has 10,312 nodes, 333,983 edges, and 39 different labels.
- Protein-Protein Interactions (PPI) [5]: We use a subgraph of the PPI network for Homo Sapiens. The subgraph corresponds to the graph induced by nodes for which we could obtain labels from the hallmark gene sets [19] and represent biological states. The network has 3,890 nodes, 76,584 edges, and 50 different labels.
- Wikipedia [20]: This is a cooccurrence network of words appearing in the first million bytes of the Wikipedia dump. The labels represent the Part-of-Speech (POS) tags inferred using the Stanford POS-Tagger [32]. The network has 4,777 nodes, 184,812 edges, and 40 different labels.

实验方法：

取一部分有标签的节点训练，预测其他节点的标签。将 node2vec 的效果与算法 Spectral clustering、 DeepWalk、LINE 对比。

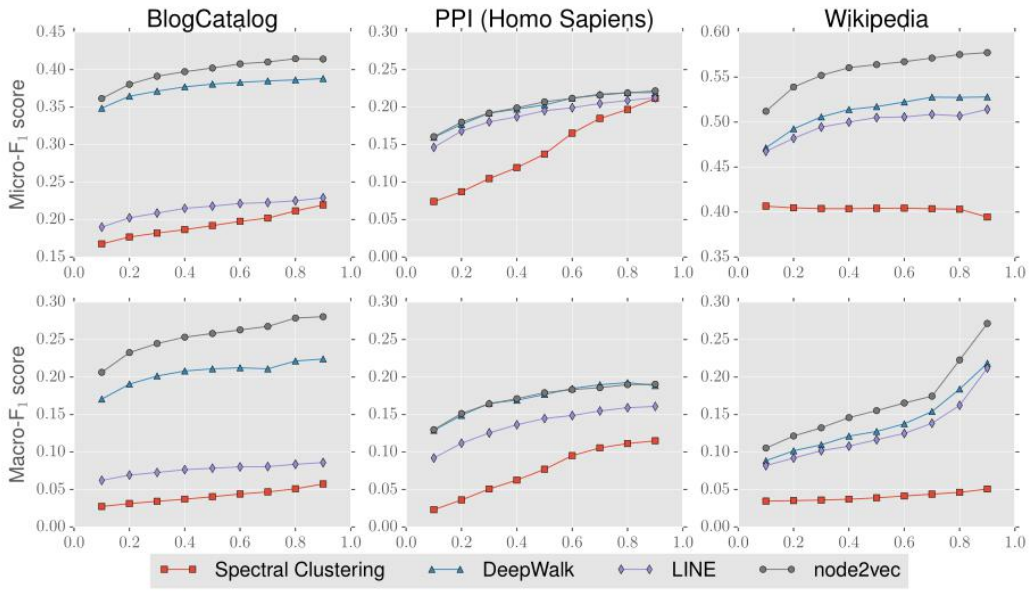
实验结果：

节点特征被输入到一个使用 L2 正则化的 one-vs-rest 逻辑回归分类器中，gain 代表 node2vec 对比其他算法中的最优效果所取得的进步。

Algorithm	**Dataset**		
	BlogCatalog	PPI	Wikipedia
Spectral Clustering	0.0405	0.0681	0.0395
DeepWalk	0.2110	0.1768	0.1274
LINE	0.0784	0.1447	0.1164
node2vec	**0.2581**	**0.1791**	**0.1552**
node2vec settings (p,q)	0.25, 0.25	4, 1	4, 0.5
Gain of *node2vec* [%]	**22.3**	**1.3**	**21.8**

Table 2: Macro-F_1 scores for multilabel classification on BlogCatalog, PPI (Homo sapiens) and Wikipedia word cooccurrence networks with 50% of the nodes labeled for training.

用作训练的有标签节点数从 10%上升到 90%的过程中四种算法的 Micor-F1 和 Macro-F1 值为：



由此可见 node2vec 除了在 PPI 任务中与 DeepWalk 效果相当外，都优于其他算法。

参数敏感度：

在 BlogCatalog 数据集上取 50%有标签数据作训练，使用控制变量法测试参数变化对实验结果的影响。当 p、q 减小时效果变好，因为较小的 q 使得采样向外探索，较低的 p 又能防止采样点离得太远。

此外，当 d=100 时已达到饱和，效果无法再提升，当 walk 的次数和长度提升时，效果也提升了。采样点数 k 增大后，也能提升效果，但伴随着较大的最优化时间以及提升效果不是很显著。

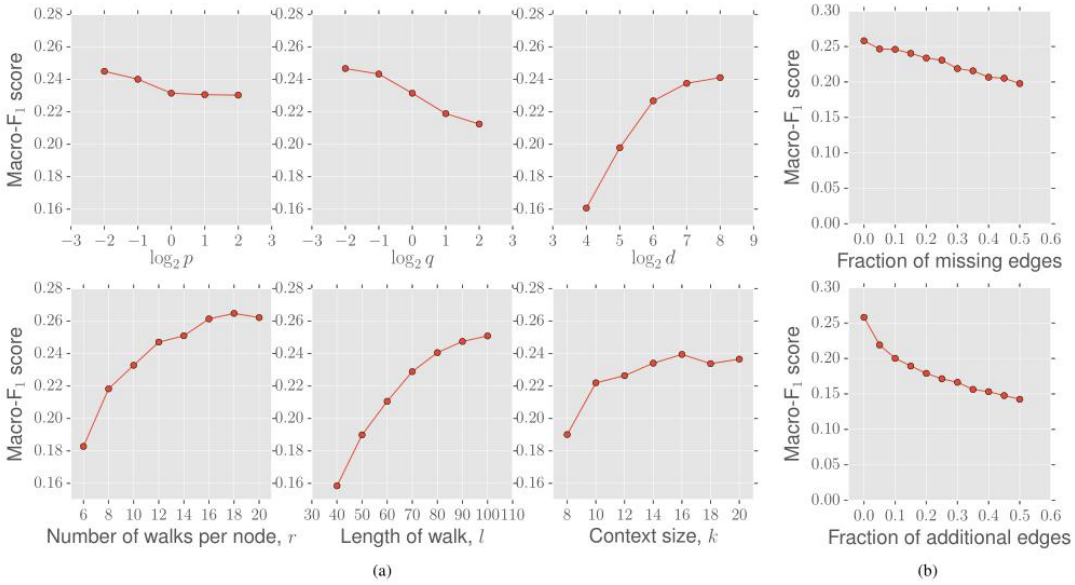
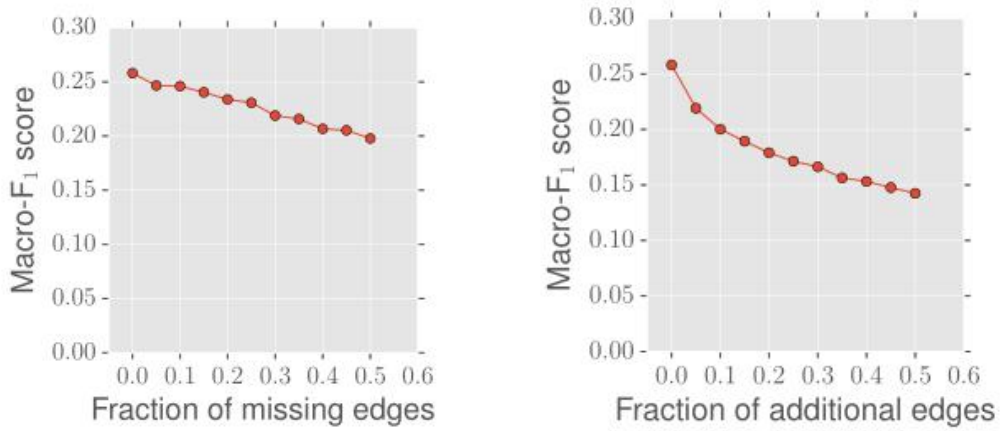


Figure 5: (a). Parameter sensitivity (b). Perturbation analysis for multilabel classification on the BlogCatalog network.

鲁棒性分析：

在真实网络中时常会有噪声，因此需考察算法的鲁棒性。

在 BlogCatalog 数据集上对比边缺失、边增多的噪声影响。



可见缺失边的情况下，随着缺失边数增加，Macro-F1 值近似线性下降，且下降幅度不大。

在边增多的情况下，随着增加边数增多，Macro-F1 值下降幅度较大但下降速率逐渐减慢。

规模化分析

图的节点数量由 100 增加到 1000000 个，保持平均度数为 10.

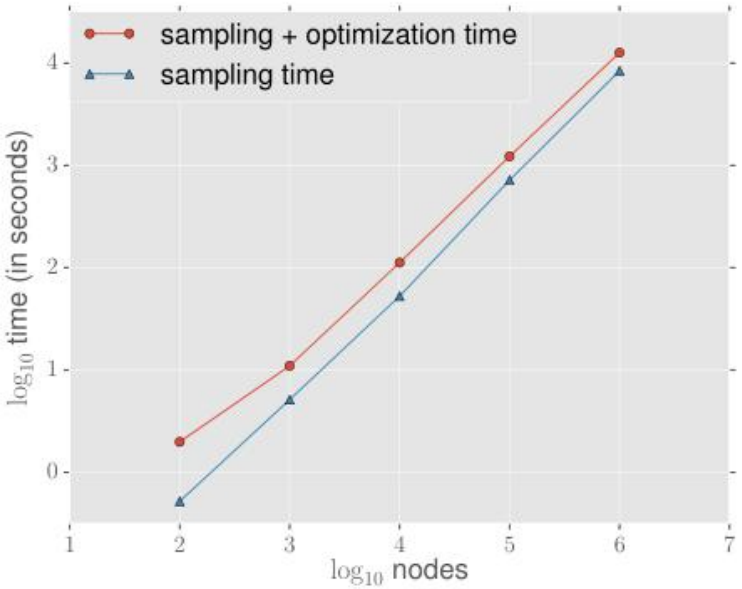


Figure 6: Scalability of *node2vec* on Erdos-Renyi graphs with an average degree of 10.

所需时间与节点数量近似呈线性关系，且一百万个节点的特征训练只需少于四个小时，说明 node2vec 也适用于大规模数据集。

3）连接预测任务

数据集：Facebook、PPI、arXiv

- Facebook [14]: In the Facebook network, nodes represent users, and edges represent a friendship relation between any two users. The network has 4,039 nodes and 88,234 edges.
- Protein-Protein Interactions (PPI) [5]: In the PPI network for Homo Sapiens, nodes represent proteins, and an edge indicates a biological interaction between a pair of proteins. The network has 19,706 nodes and 390,633 edges.
- arXiv ASTRO-PH [14]: This is a collaboration network generated from papers submitted to the e-print arXiv where nodes represent scientists, and an edge is present between two scientists if they have collaborated in a paper. The network has 18,722 nodes and 198,110 edges.

实验方法：去掉 50%的边且保持图连通，然后取相同数量的节点对（不邻接），进行连接预测。将 node2vec 与其他算法的预测效果对比，其中连接预测的几种启发式算法

Score	Definition
Common Neighbors	$\mid \mathcal{N}(u) \cap \mathcal{N}(v) \mid$
Jaccard's Coefficient	$\frac{\mid \mathcal{N}(u) \cap \mathcal{N}(v) \mid}{\mid \mathcal{N}(u) \cup \mathcal{N}(v) \mid}$
Adamic-Adar Score	$\sum_{t \in \mathcal{N}(u) \cap \mathcal{N}(v)} \frac{1}{\log \mid \mathcal{N}(t) \mid}$
Preferential Attachment	$\mid \mathcal{N}(u) \mid \cdot \mid \mathcal{N}(v) \mid$

Table 3: Link prediction heuristic scores for node pair (u, v) with immediate neighbor sets $\mathcal{N}(u)$ and $\mathcal{N}(v)$ respectively.

实验结果：

Op	Algorithm	Dataset		
		Facebook	PPI	arXiv
	Common Neighbors	0.8100	0.7142	0.8153
	Jaccard's Coefficient	0.8880	0.7018	0.8067
	Adamic-Adar	0.8289	0.7126	0.8315
	Pref. Attachment	0.7137	0.6670	0.6996
(a)	Spectral Clustering	0.5960	0.6588	0.5812
	DeepWalk	0.7238	0.6923	0.7066
	LINE	0.7029	0.6330	0.6516
	node2vec	0.7266	0.7543	0.7221
(b)	Spectral Clustering	0.6192	0.4920	0.5740
	DeepWalk	**0.9680**	0.7441	0.9340
	LINE	0.9490	0.7249	0.8902
	node2vec	**0.9680**	**0.7719**	**0.9366**
(c)	Spectral Clustering	0.7200	0.6356	0.7099
	DeepWalk	0.9574	0.6026	0.8282
	LINE	0.9483	0.7024	0.8809
	node2vec	0.9602	0.6292	0.8468
(d)	Spectral Clustering	0.7107	0.6026	0.6765
	DeepWalk	0.9584	0.6118	0.8305
	LINE	0.9460	0.7106	0.8862
	node2vec	0.9606	0.6236	0.8477

Table 4: Area Under Curve (AUC) scores for link prediction. Comparison with popular baselines and embedding based methods bootstapped using binary operators: (a) Average, (b) Hadamard, (c) Weighted-L1, and (d) Weighted-L2 (See Table 1 for definitions).

总的来说，node2vec 选 Hadamard 作二元运算符的情况下预测效果最稳定且几乎比其他所有算法的预测效果都要好。

5.评价

1）优点

a）random walk 中采样策略灵活。能通过调整参数 p、q 来将 BFS、DFS 综合起来，且参数 p、q 具有直观的可解释性，适当调参可适应不同类型的网络。
b）预测效果比目前相同领域的的算法好。
c）可适用于大规模数据集。百万级别节点数的网络也能在数个小时内训练完。
d）鲁棒性良好。即使网络发生变化，预测效果所受影响不大。

2）缺点

a）将 nlp 领域中的 word2vec 算法迁移类比到网络中这一点其实还是 deepwalk 先做的，我认为 node2vec 其实就是改进了近邻节点采样的过程。
b）只涉及到无向图网络，没有考虑节点之间单向联系的情况
c）能处理预测节点对连接存在与否的任务，难以预测连接类型，比如社交网络中两个节点间产生联系可以是通过点赞也可以是通过点踩。

3）改进思路

1）探究为什么 Hadamard 作运算符效果比其他的好
2）在有向图中的运用和改进。
3）比如在社交网络上以用户为节点，用户之间的边权值代表用户之间的交互类型，将对边权值的预测转化为分类问题。