

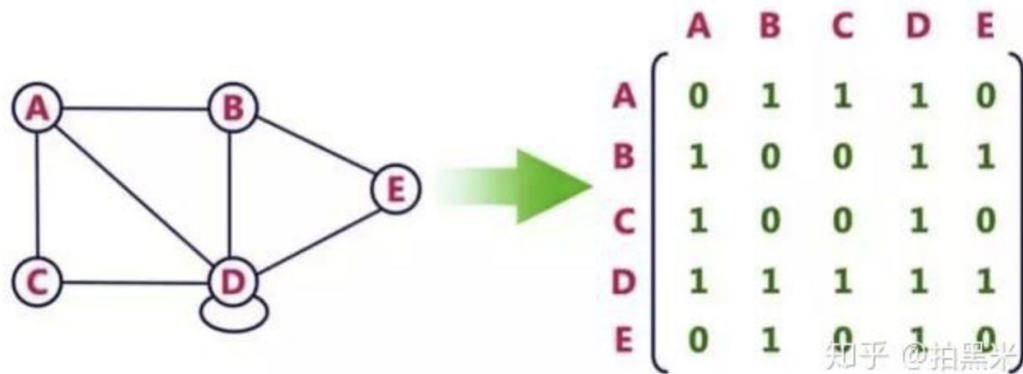


Network Embedding



如何描述一个网络

面对多个对象，我们首先根据对象之间普遍存在的某种关系将它们连接起来，从而以网络的形式获得整个对象群体。我们知道在数据结构中，最早使用的描述网络的语言是邻接矩阵，它同义于one-hot编码。



尽管邻接矩阵可以捕捉到网络的全部信息，但信息的抽象程度很低，数据过于稀疏也使得邻接矩阵的维度太高（=节点数量的平方），节点对应的one-hot向量不适合作为节点的特征输入模型。Embedding致力于将庞大的节点网络（十万级以上）作为输入，最终为每个节点学习出可以有效代表该节点的低维（百维左右）的向量，最终将向量或向量生成的变量分数作为其他机器学习任务的输入变量。

NE-臨近關係v.s.網路結構

复杂网络Embedding发展至今，大致可以分为两种基本类型：一类重在挖掘节点临近关系，以Deepwalk[1]、node2vec[2]、LINE[3]为代表，被称为基于临近关系的表示学习，其二是重在挖掘节点结构特征的，以struc2vec[4]为代表，被称为基于网络结构的表示学习。

- 基于临近关系的表示学习

在自然语言处理领域，随着Word2vec算法的提出，我们通过考察单词与背景文本共同出现的概率，可以将单词映射为有限维欧式空间中的向量，从而实现对文本的数值化分析。沿着这个思路，Deepwalk提出使用类似的方法。它考察节点与临近节点共同出现的概率，

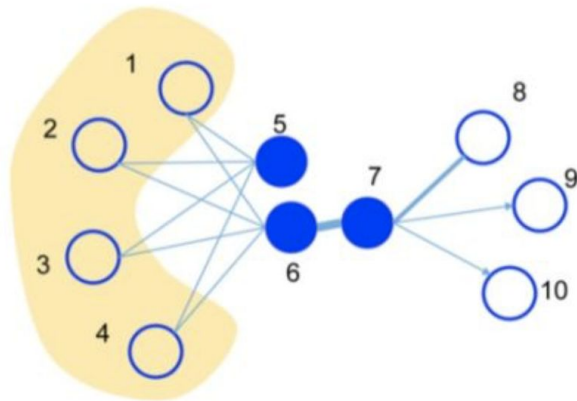
常見Network Embedding Method

Deepwalk	LINE
2阶相似性	1、2阶相似性
深度优先	广度优先
无权图	无权图、带权图

1階、2階相似

本文将下面两种情况的两个顶点归结为相似顶点：

1. 如果两个顶点之间有一条强连接的边（权重很大的边），那么这两个顶点就是相似的。图中顶点6与7就是这种相似。
2. 如果两个顶点共享了很多相同的邻居顶点，那么这两个顶点也是相似的。图中顶点5和6虽然没有直接相连，但是他们同时连接到了顶点1234，所以顶点5和6也是相似的。



1階相似

Algorithm

首先来看1阶相似性，前面说到1阶相似性是用来衡量两个直接相连的顶点的相似性的，边的权值越大相似性越高。那么怎么将这样的概念用可计算的形式表示出来呢？文中给出的公式是这样的：

$$p_1(v_i, v_j) = \frac{1}{1 + \exp(-\vec{u}_i^T \cdot \vec{u}_j)}$$

$$\vec{u}_i \in \mathbb{R}^d$$

知乎 @陌上疏影凉

其实 p_1 就是两个顶点的向量表示乘积的simoid函数。为什么是这种形式文中也没给出详细的解释。接下来考虑1阶相似性的经验概率，其实就是 (i, j) 这条边的权重 w_{ij} 占所有权重之和的比例。从变化趋势来分析， w_{ij} 越大， p_1 越大，与1阶相似性的变化趋势相同，反之亦然。经验概率的公式如下：

$$\hat{p}_1(i, j) = \frac{w_{ij}}{W}, \quad W = \sum_{(i, j) \in E} w_{ij}$$

知乎 @陌上疏影凉

2階相似

接下来看2阶相似性的计算，2阶相似性描述了两个顶点的邻居相似度，文中定义2阶相似度的公式如下：

$$p_2(v_j | v_i) = \frac{\exp(\vec{u}_j'^T \cdot \vec{u}_i)}{\sum_{k=1}^{|V|} \exp(\vec{u}_k'^T \cdot \vec{u}_i)}$$

of vertices or "context"

\vec{u}_i the representation of v_i treated as vertex

\vec{u}_i' the representation of v_i treated as context

知乎 @陌上疏影凉

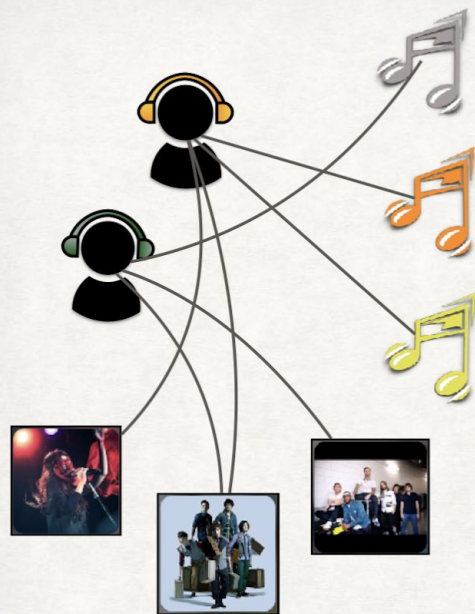







proNet:

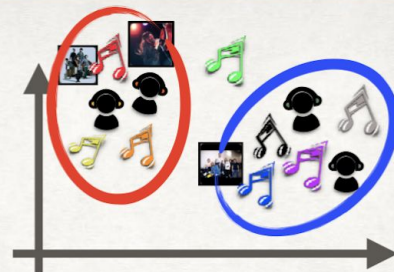
Pair-wise Representations Optimization Network



<https://github.com/cnclabs/proNet-core>



	$[0.08 \ 0.02 \ 0.28]$
	$[-0.2 \ 0.25 \ 0.21]$
	$[0.02 \ 0.13 \ 0.21]$
	$[-0.31 \ -0.1 \ 0.1]$
	$[-0.15 \ -0.3 \ 0.2]$



Computable Relationship



Clustering

Classification

Recommendation

Prediction Tasks

**Network/
Knowledge Graph**

Representation

Network Representation Learning

Application

Command Line Interface

Directly call the execution file to see the usage like:

```
./cli/deepwalk  
./cli/walklets  
./cli/line  
./cli/hpe  
./cli/app  
./cli/mf  
./cli/bpr  
./cli/warp  
./cli/hoprec
```

Options	Description:
-train <string>	Train the Network data
-save <string>	Save the representation data
-dimensions <int>	Dimension of vertex representation; default is 64
-undirected <int>	Whether the edge is undirected; default is 1
-negative_samples <int>	Number of negative examples; default is 5
-window_size <int>	Size of skip-gram window; default is 5
-walk_times <int>	Times of being staring vertex; default is 10
-walk_steps <int>	Step of random walk; default is 40
-threads <int>	Number of training threads; default is 1
-alpha <float>	Init learning rate; default is 0.025

Usage:

```
./deepwalk -train net.txt -save rep.txt -undirected 1 -dimensions 64 -walk_times 10 -walk_steps 40  
-window_size 5 -negative_samples 5 -alpha 0.025 -threads 1
```



DEMO

Experiments

Embedding Method	Interval	Target
Deepwalk	1, 3, 6 month	Flam1(實付金額)
Line	1, 3, 6 month	Flam1(實付金額)
Walklets	1, 3, 6 month	Flam1(實付金額)
MF	1, 3, 6 month	Flam1(實付金額)