Beijing Forest Studio
北京理工大学信息系统及安全对抗实验中心

网络表示学习–Deepwalk

硕士研究生　杨俊楠

2019年3月17日

内容提要

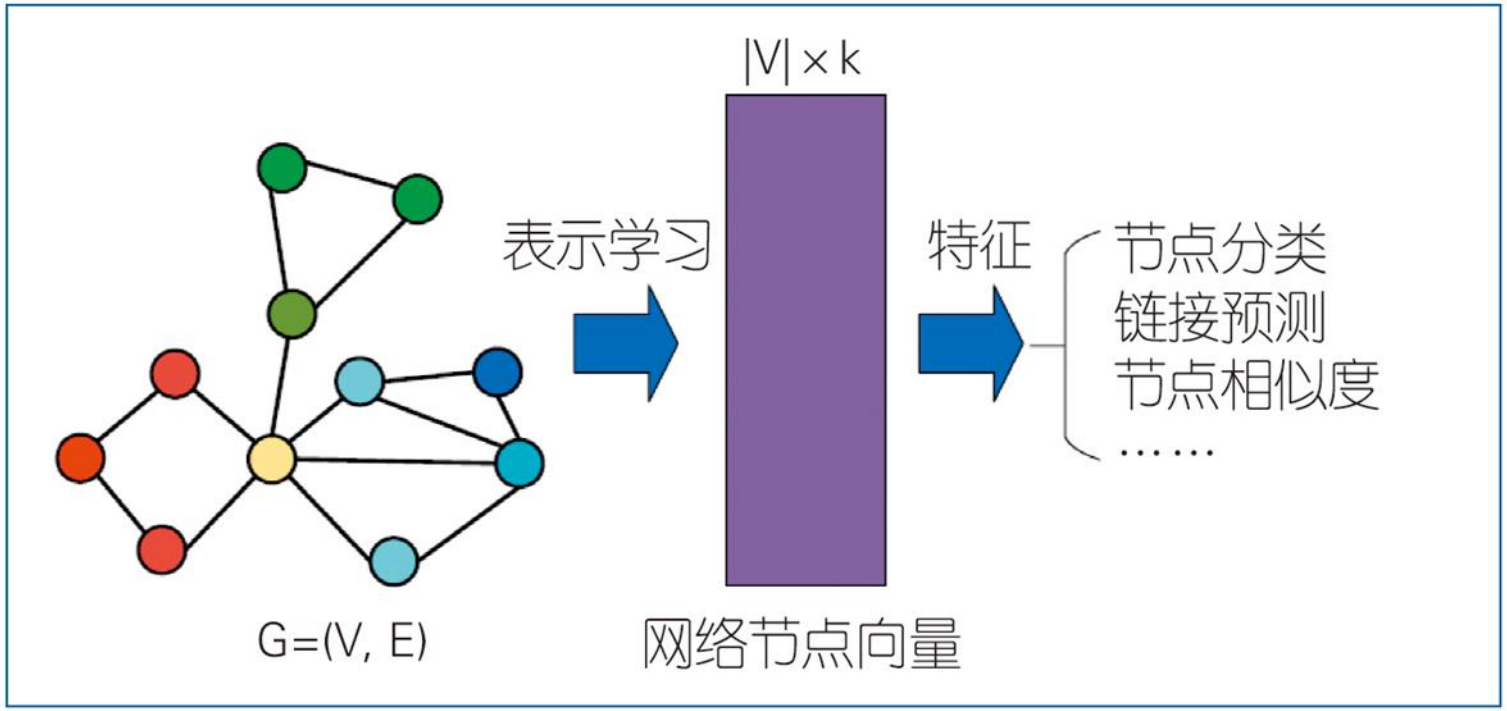
- 背景简介
- 基本概念
- 算法原理
- 优劣分析
- 模型总结
- 参考文献

预期收获

- 预期收获
  - 1. 了解网络表示学习基本思想
  - 2. 理解Deepwalk设计思路以及框架原理

背景介绍

- **网络嵌入**（Network Embedding）[1]



[1] 涂存超等：网络表示综述

背景介绍

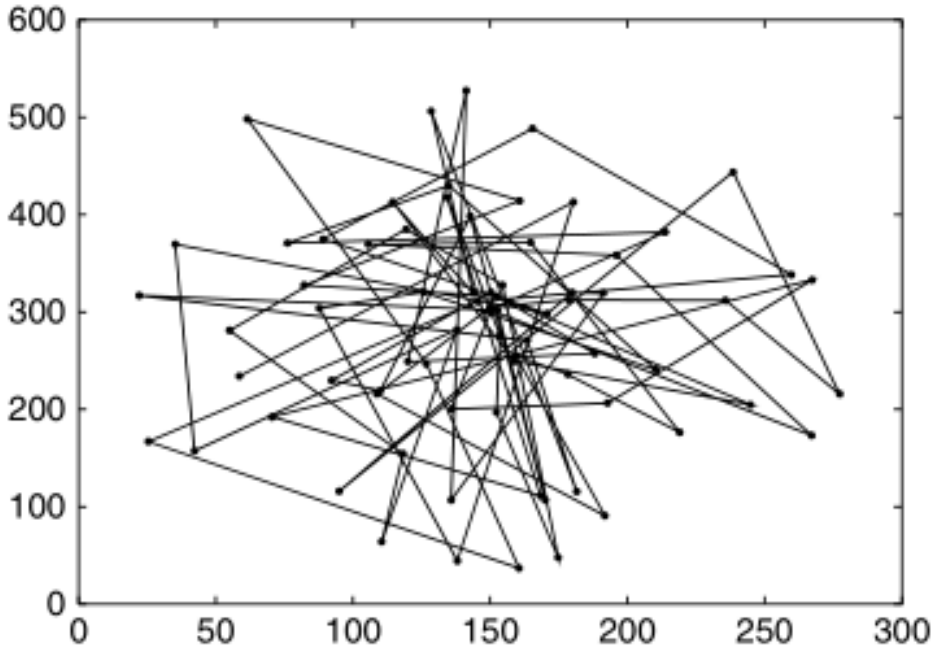
- 网络表示学习方法

基本概念

基本概念

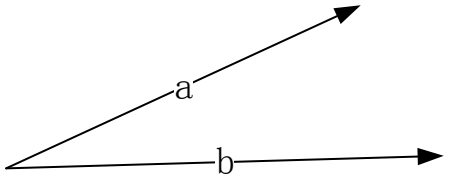
- 随机游走

  基于过去的表现，无法预测将来的发展步骤和方向。

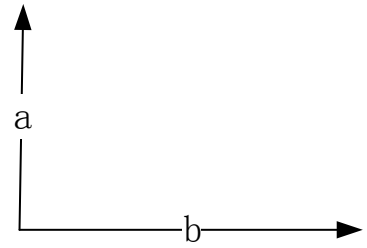


基本概念

- 向量内积
  - $\boldsymbol{a} = [a_1, a_2, \dots a_n]$ $\quad\quad$ $\boldsymbol{b} = [b_1, b_2, \dots b_n]$
  - $\mathbf{a} \cdot \boldsymbol{b} = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$
  - $\mathbf{a} \cdot \boldsymbol{b} = |a||b|cos\theta$



a与b相似

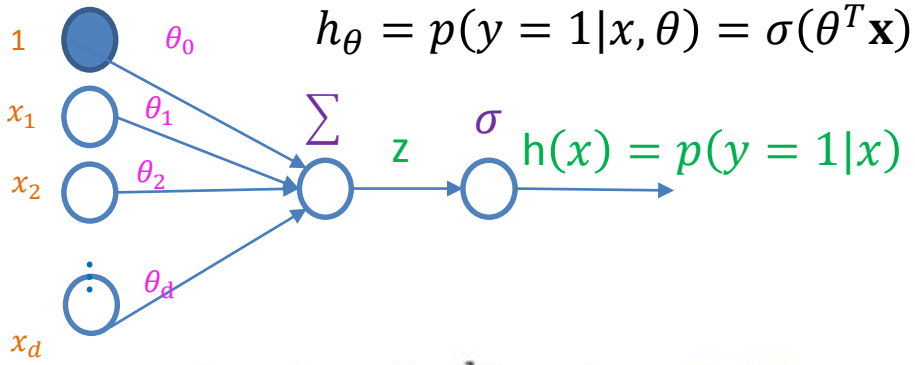


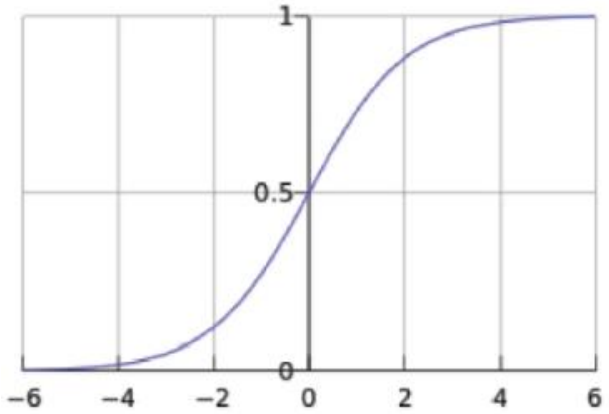
a与b不相似

所以向量内积可以用来表示向量之间的相似度

基本概念

- 逻辑回归
  - 用于二分类问题





Sigmoid函数

训练样本： $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$

输入特征： $\mathrm{x}^{(i)} \in R^{\mathrm{n}+1}$ **类标记：** $\mathrm{y}^{(i)} \epsilon \{0,1\}$

假设函数：

$$h_\theta(x) = \sigma(\theta^T x) = \frac{1}{1 + e^{-\theta^T X}}$$

$$\mathrm{y}(\mathbf{x}) = \begin{cases} 1, & h_\theta(\mathbf{x}) \geq 0.5; \\ 0, & h_\theta(\mathbf{x}) < 0.5. \end{cases}$$

损失函数：

$$J_\theta(x) = -\frac{1}{m}\sum_{i=0}^{m}\left(y^{(i)} log h_\theta\left(x^{(i)}\right) + \left(1 - y^{(i)}\right)\log(1 - h_\theta(x^{(i)}))\right)$$

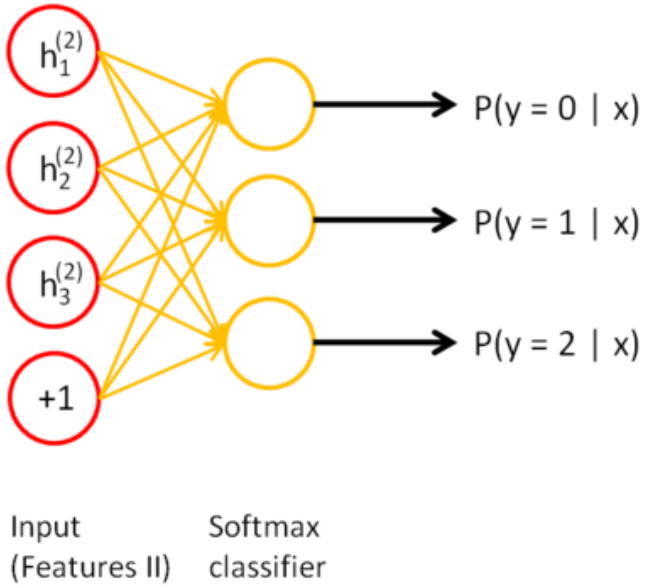
基本概念

- Softmax回归
  - 解决多分类问题
    - 训练集：$\{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$
    - 类标记：$y^{(i)} \in \{1,2, \ldots, k\}$
    - 假设函数：

$$h_\theta(x^{(i)}) = \begin{bmatrix} p(y^{(i)} = 1|x^{(i)};\theta) \\ p(y^{(i)} = 2|x^{(i)};\theta) \\ \vdots \\ p(y^{(i)} = k|x^{(i)};\theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}} \begin{bmatrix} e^{\theta_1^T x^{(i)}} \\ e^{\theta_2^T x^{(i)}} \\ \vdots \\ e^{\theta_k^T x^{(i)}} \end{bmatrix}$$

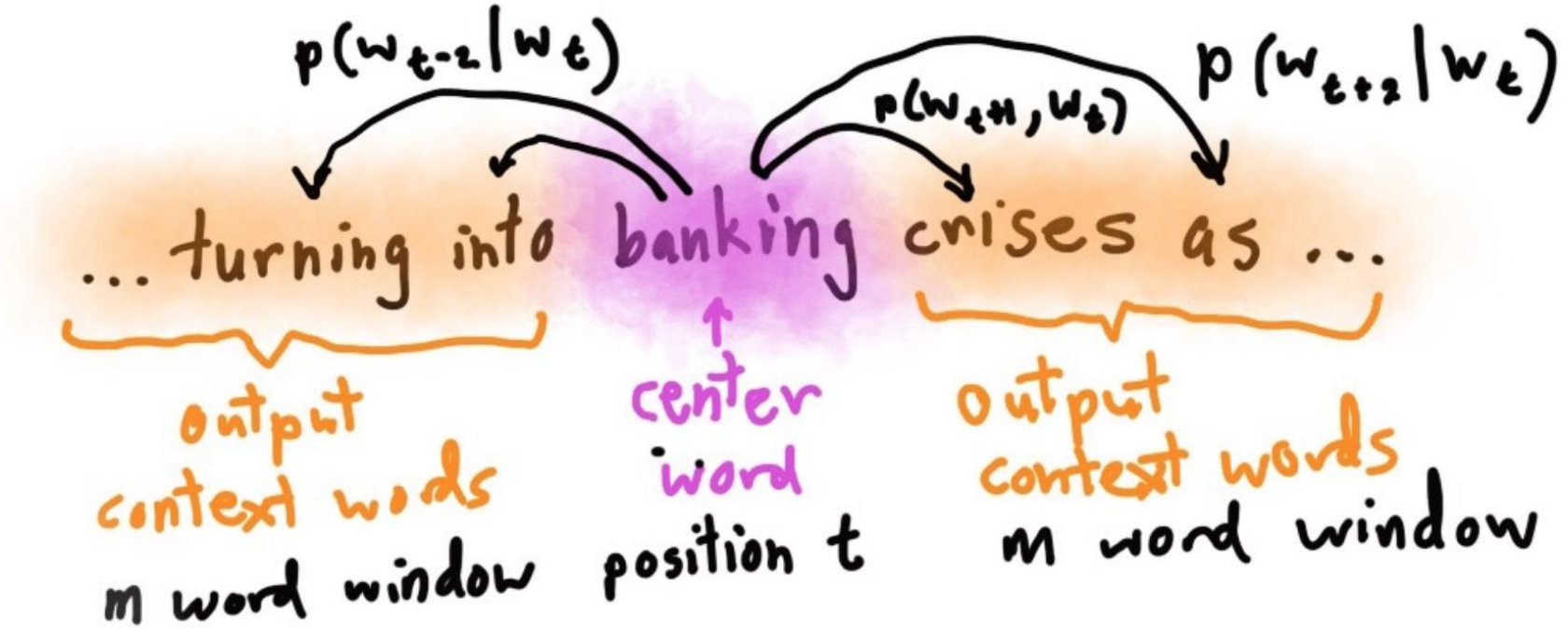
    - 代价函数：

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{j=1}^{k} 1\{y^{(i)} = j\} \log \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^{k} e^{\theta_l^T x^{(i)}}}\right]$$



基本知识

- SkipGram模型
  - 通过中心词预测上下文，该模型给出了给定中心词后，上下文中某个词出现的概率，即图中的$P(w_{t-2}|w_t)$, $P(w_{t-1}|w_t)$, $P(w_{t+1}|w_t)$, $P(w_{t+2}|w_t)$, SkipGram要做的事情就是最大化这些概率。



- SkipGram模型

将目标函数定义为所有位置的预测结果的乘积：

$$J'(\theta) = \prod_{t=1}^{T} \prod_{\substack{-m \le j \le m \\ j \ne 0}} \left(p(w_{t+j}|w_t;\theta)\right)$$

对目标函数取对数并添加负号，就得到了上式，目标就变成了最小化J(Θ)：

$$\mathrm{J}(\theta) = -\frac{1}{T}\sum_{t=1}^{T} \sum_{\substack{-m \le j \le m \\ j \ne 0}} \log(p(w_{t+j}|w_t))$$

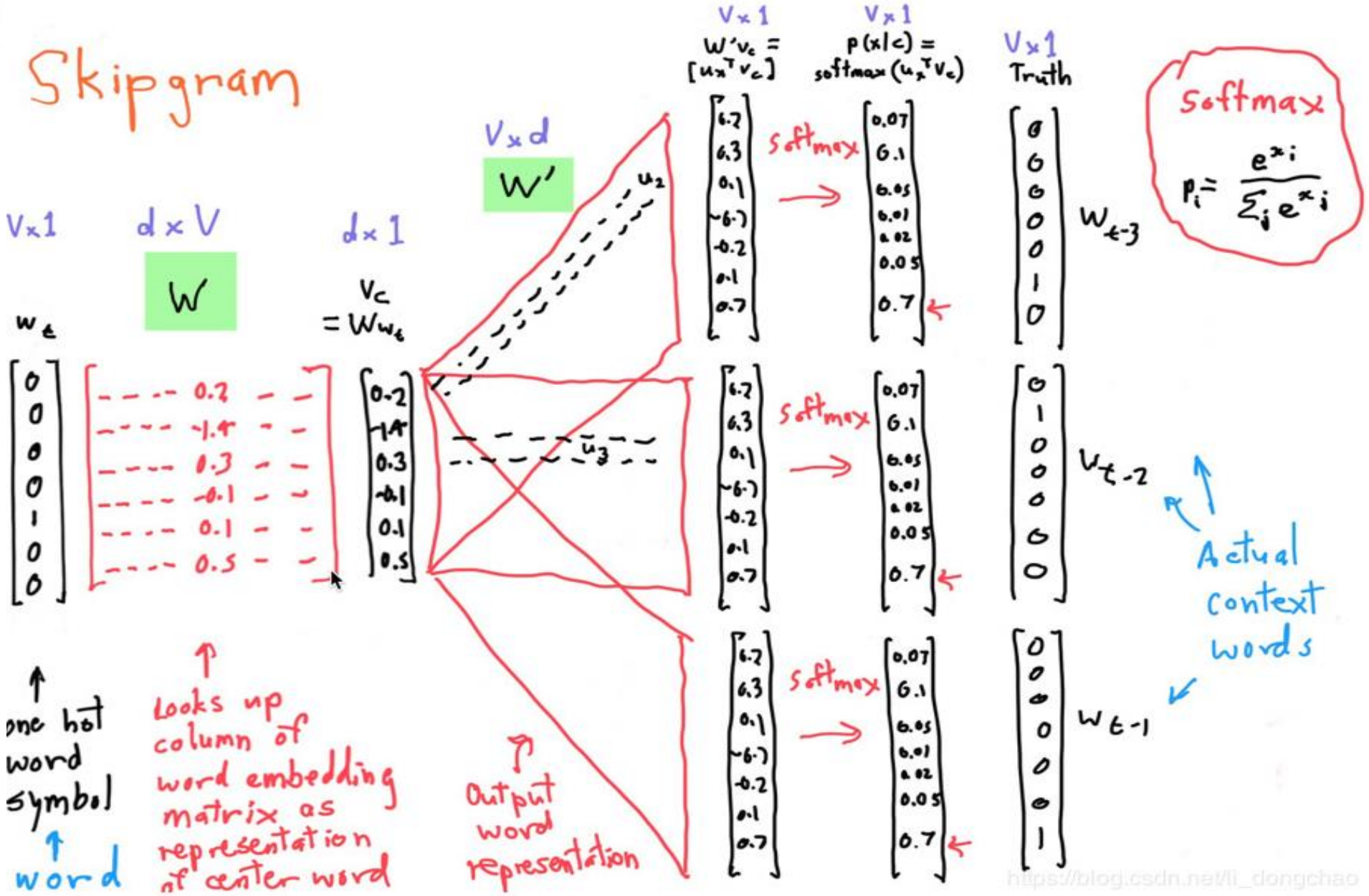
- SkipGram模型

式中$P(w_{t+\mathrm{j}}|w_t)$,是根据softmax得到的：

$$\mathrm{p(o|c)} = \frac{\exp(u_o^T v_c)}{\sum_{w=1}^{V} \exp(u_w^T v_c)}$$

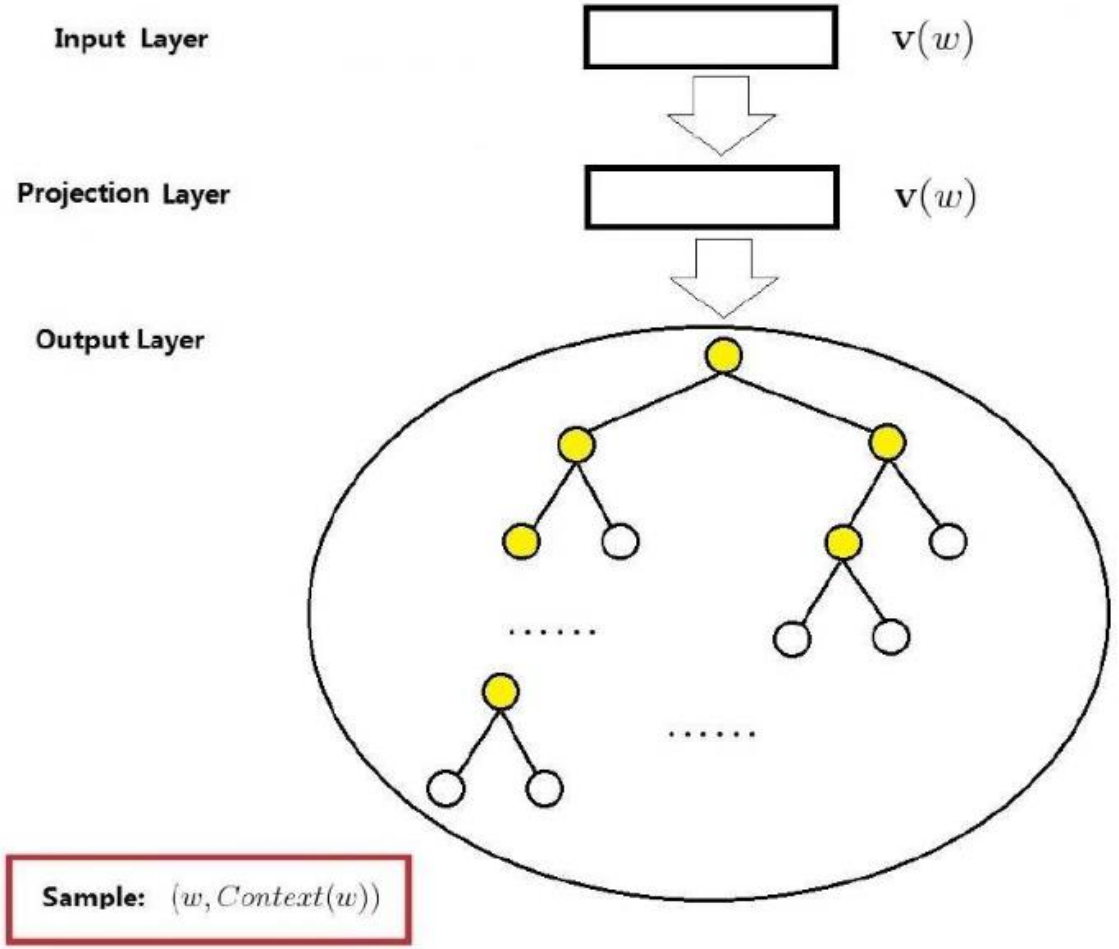
其中，c代表中心词，o代表该中心词上下文中的某一个单词，u和v代表的单词的向量表示。所以就是通过计算中心词与上下文单词的相似性来确定上下文某个单词出现的概率。

基本知识



基本知识

- 基于Hierarchical softmax的SkipGram模型



- **基于Hierarchical softmax的SkipGram模型**

Hierarchical softmax的中心思想：

对于词典$\mathcal{D}$中的任意词w，Huffman树中必存在一条从根节点到词w对应节点的路径p^{w}（且这条路径是唯一的）。路径p^{w}上存在$l^{\mathrm{w}}-1$个分支，将每个分支看做一次二分类，每一次分类就产生一个概率，将这些概率进行累乘，就是所需的$p(context(w)|w)$。

- **基于Hierarchical softmax的SkipGram模型**

一个节点分到左边为负类（1），分到右边为正类（0）：

分为正类的概率：

$$\sigma(v(m)^T\theta) = \frac{1}{1+e^{-v(m)^T\theta}}$$

分为负类的概率：

$$1-\sigma(v(m)^T\theta)$$

- **基于Hierarchical softmax的SkipGram模型**

条件概率函数定义为：

$$\mathrm{p}(\mathrm{context(w)|w}) = \prod_{u \in context(w)} \mathrm{p(u|w)}$$

按照Hierarchical softmax，p(u|w)定义为：

$$\mathrm{p(u|w)} = \prod_{j=2}^{l^u} p(d_j^u | v(m), \theta_{j-1}^u)$$

其中：

$$\mathrm{p}\left(d_j^u | v(m), \theta_{j-1}^u\right) = \begin{cases} \sigma\left(v(m)^T \theta_{j-1}^u\right), & d_j^u = 0 \\ 1 - \sigma\left(v(m)^T \theta_{j-1}^u\right), & d_j^u = 1 \end{cases}$$

- **基于Hierarchical softmax的SkipGram模型**

  即：

$$p\left(d_j^u \middle| v(m), \theta_{j-1}^u\right) = \left[\sigma\left(v(m)^T \theta_{j-1}^u\right)\right]^{1-d_j^u} * \left[1 - \sigma\left(v(m)^T \theta_{j-1}^u\right)\right]^{d_j^u}$$

网络表示学习–Deepwalk

算法原理

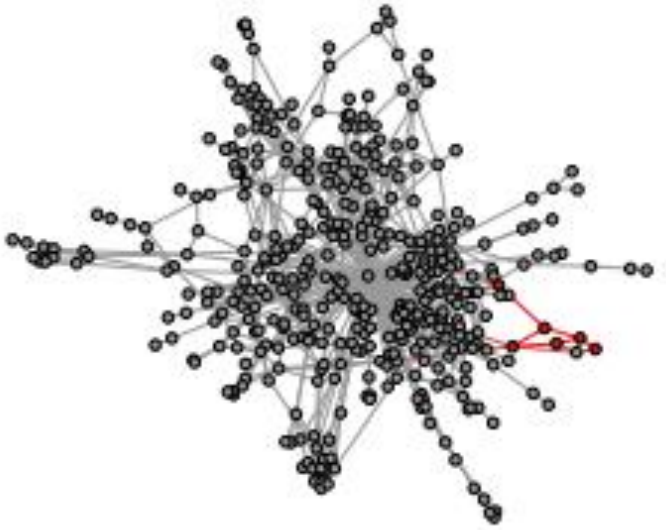
算法原理

P	如何在缺少信息、标记数据稀疏的情况下得到较好的网络表示
C	Word2vec在无标签文本上可以得到很好的词向量
D	如何将Word2vec的方法应用到网络表示中
L	CCF A+（KDD）

算法原理

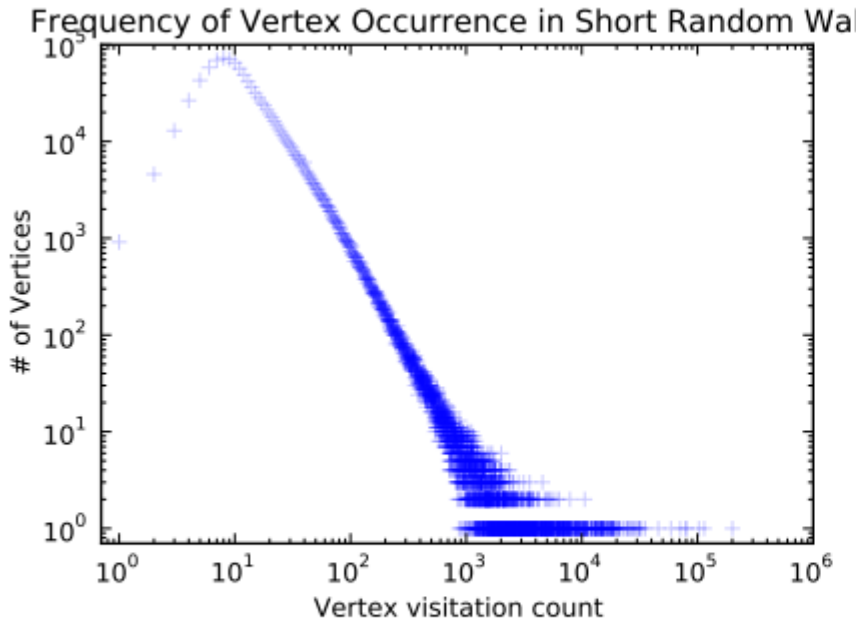
SkipGram中的词 → 网络中的节点

SkipGram中的句子 → 随机游走序列

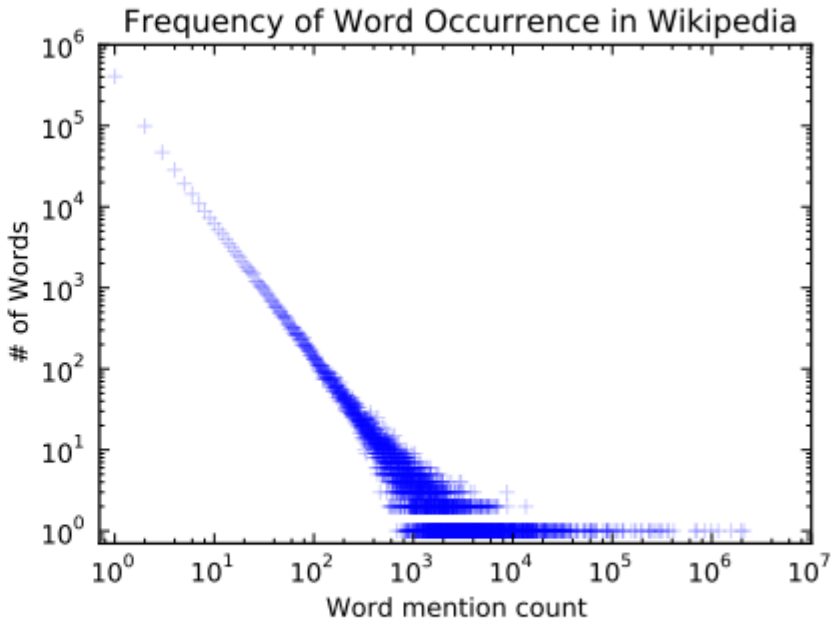


随机游走发生器

算法原理



(a) YouTube Social Graph



(b) Wikipedia Article Text

自然语言中的词频分布规律和网络中节点频率的分布规律都服从幂律分布，因此可以使用此方法！

算法原理

T	将网络中的节点映射成为低维向量表示
I	网络的节点、边

P	For{
	1. 随机游走得到节点随机游走序列
	2. SkipGram模型用中间节点预测上下文节点
	}

O	每个节点的低维稠密表示向量

算法原理

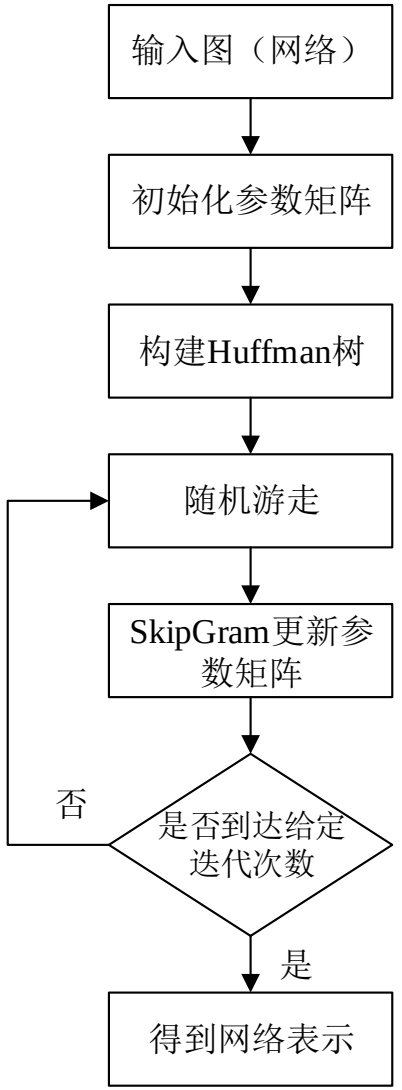
算法步骤与流程图

Algorithm 1 DEEPWALK(G, w, d, γ, t)

Input: graph $G(V, E)$
window size w
embedding size d
walks per vertex γ
walk length t

Output: matrix of vertex representations $\Phi \in \mathbb{R}^{|V| \times d}$

1: Initialization: Sample Φ from $\mathcal{U}^{|V| \times d}$
2: Build a binary Tree T from V
3: **for** $i = 0$ to γ **do**
4: $\mathcal{O} = \text{Shuffle}(V)$
5: **for each** $v_i \in \mathcal{O}$ **do**
6: $\mathcal{W}_{v_i} = RandomWalk(G, v_i, t)$
7: SkipGram(Φ, $\mathcal{W}_{v_i}$, w)
8: **end for**
9: **end for**



优劣分析

- **数据集：**BlogCatalog，Flickr，YouTube
- **对比方法：**SpectralClustering，Modularity，EdgeCluster，wvRN，Majority

	% Labeled Nodes	10%	20%	30%	40%	50%	60%	70%	80%	90%
Micro-F1(%)	DeepWalk	**36.00**	**38.20**	**39.60**	**40.30**	**41.00**	**41.30**	41.50	41.50	42.00
	SpectralClustering	31.06	34.95	37.27	38.93	39.97	40.99	**41.66**	**42.42**	**42.62**
	EdgeCluster	27.94	30.76	31.85	32.99	34.12	35.00	34.63	35.99	36.29
	Modularity	27.35	30.74	31.77	32.97	34.09	36.13	36.08	37.23	38.18
	wvRN	19.51	24.34	25.62	28.82	30.37	31.81	32.19	33.33	34.28
	Majority	16.51	16.66	16.61	16.70	16.91	16.99	16.92	16.49	17.26
Macro-F1(%)	DeepWalk	**21.30**	**23.80**	25.30	26.30	27.30	27.60	27.90	28.20	28.90
	SpectralClustering	19.14	23.57	**25.97**	**27.46**	**28.31**	**29.46**	**30.13**	**31.38**	**31.78**
	EdgeCluster	16.16	19.16	20.48	22.00	23.00	23.64	23.82	24.61	24.92
	Modularity	17.36	20.00	20.80	21.85	22.65	23.41	23.89	24.20	24.97
	wvRN	6.25	10.13	11.64	14.24	15.86	17.18	17.98	18.86	19.57
	Majority	2.52	2.55	2.52	2.58	2.58	2.63	2.61	2.48	2.62

Table 2: Multi-label classification results in BlogCatalog

优劣分析

	% Labeled Nodes	1%	2%	3%	4%	5%	6%	7%	8%	9%	10%
Micro-F1(%)	DeepWalk	**37.95**	**39.28**	**40.08**	**40.78**	**41.32**	**41.72**	**42.12**	**42.48**	**42.78**	**43.05**
	SpectralClustering	—	—	—	—	—	—	—	—	—	—
	EdgeCluster	23.90	31.68	35.53	36.76	37.81	38.63	38.94	39.46	39.92	40.07
	Modularity	—	—	—	—	—	—	—	—	—	—
	wvRN	26.79	29.18	33.1	32.88	35.76	37.38	38.21	37.75	38.68	39.42
	Majority	24.90	24.84	25.25	25.23	25.22	25.33	25.31	25.34	25.38	25.38
Macro-F1(%)	DeepWalk	**29.22**	**31.83**	**33.06**	**33.90**	**34.35**	**34.66**	**34.96**	**35.22**	**35.42**	**35.67**
	SpectralClustering	—	—	—	—	—	—	—	—	—	—
	EdgeCluster	19.48	25.01	28.15	29.17	29.82	30.65	30.75	31.23	31.45	31.54
	Modularity	—	—	—	—	—	—	—	—	—	—
	wvRN	13.15	15.78	19.66	20.9	23.31	25.43	27.08	26.48	28.33	28.89
	Majority	6.12	5.86	6.21	6.1	6.07	6.19	6.17	6.16	6.18	6.19

Table 4: Multi-label classification results in YouTube

Deepwalk总结

优势：

随机游走序列只依赖于局部信息,所以可以适用于分布式和在线系统；

系统具有鲁棒性，对参数不敏感

劣势：

随机游走过程设置简单，只是随机进行深搜；

应用：

节点分类、链路预测、社区发现、推荐系统.....

参考文献

- B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: Online learning of social representations. 2014.
- Word2vec中的数学原理详解 https://www.cnblogs.com/peghoty/p/3857839.html
- 刀口木博客：CS224n课程笔记2 word2vec介绍 https://blog.csdn.net/li_dongchao/article/details/83589692#_38

道德经

知人者智，自知者明。

胜人者有力，自胜者强。

知足者富，强行者有志。

不失其所者久，死而不亡者，寿。

谢谢！