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-SDNE

周妍汝 博士研究生

2019年03月24日

内容提要

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

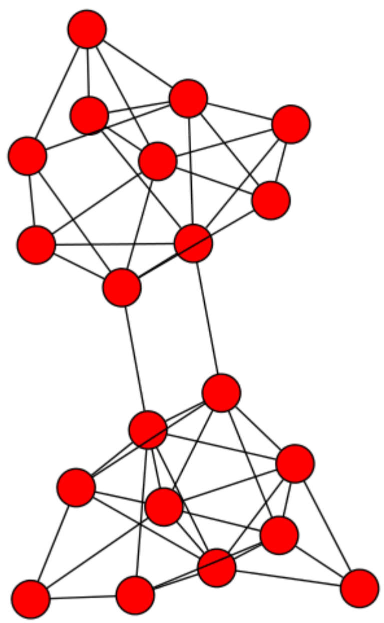
背景简介

- 预期收获
  - 1. 了解网络表示学习基本思想
  - 2. 理解SDNE的算法原理
  - 3. 了解网络表示学习的应用

背景简介

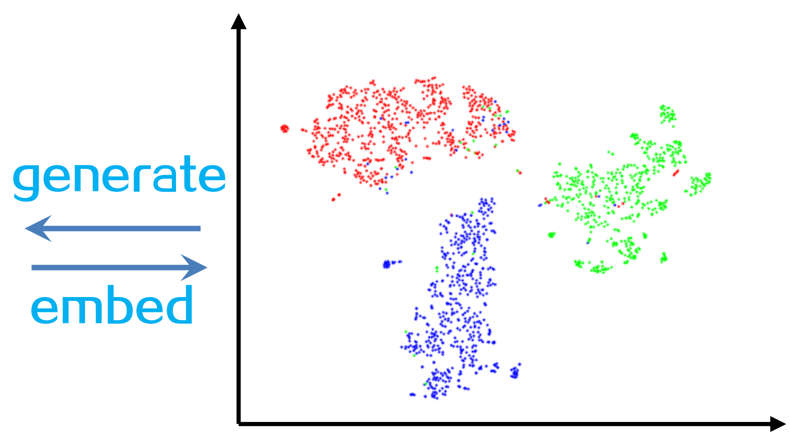
- 网络表示学习用来做什么？　　网络嵌入 or 图嵌入
  - 网络表示学习算法负责从网络数据中学习得到网络中每个节点的向量表示，之后这些节点表示就可以作为节点的特征应用于后续的网络应用任务，如节点分类、链接预测等。

G = (V, E)



G = (V)

Vector Space



Can apply classical ML methods

背景简介

- 有哪些网络表示学习方法？ SDNE(Structure Deep Network Embedding)

网络表示学习

- 基于矩阵特征向量
  - 局部线性表示
  - Laplace Eigenmaps
  - 有向图表示
- 基于简单神经网络
  - DeepWalk
  - LINE
- 基于矩阵分解
  - GraRep
- 基于深层神经网络
  - SDNE
- 基于社区发现
  - BIGCLAM
- 保存特殊性质的网络表示
  - HOPE
  - CNRL

基本概念

基本概念

- 基本概念
  - 图
  - 一阶相似度
  - 二阶相似度
  - 网络嵌入（Network Embedding）
  - 深度编码器
  - 拉普拉斯特征映射(Laplacian Eigenmaps)

基本概念

- 图
  - 将网络记为图$G(V,E)$，其中$V=\{v_1,v_2,\cdots,v_n\}$是节点集合，$E=\{e_{ij}\}_{i,j=1}^{n}$是边的集合，e_{ij}表示节点v_i和v_j之间的边。

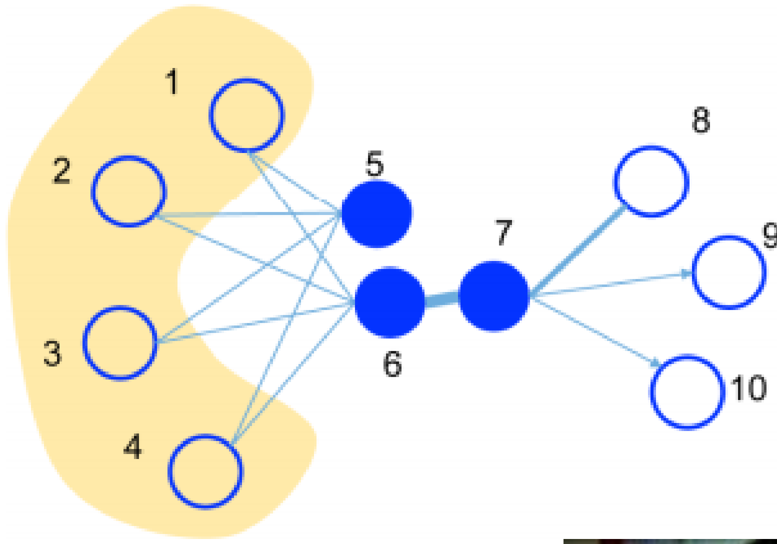
基本概念

- 一阶相似度
  - 图G的邻接矩阵定义为$S \in \mathbb{R}^{n \times n}$，包含与每个边相关的非负权重：$s_{i,j} \geq 0$。如果$v_i$和$v_j$没有相互连接，则$s_{i,j} = 0$。
  - 邻接矩阵S中的一行 $S_i = \left\{s_{i,1}, s_{i,2}, \cdots, s_{i,|V|}\right\}$ 表示v_i与其他顶点之间的一阶相似度。如果权重$s_{i,j} > 0$，则v_i和v_j之间存在正的一阶相似度，权重越高，两个节点越相似。如果节点之间没有连接，一阶相似性为0 。

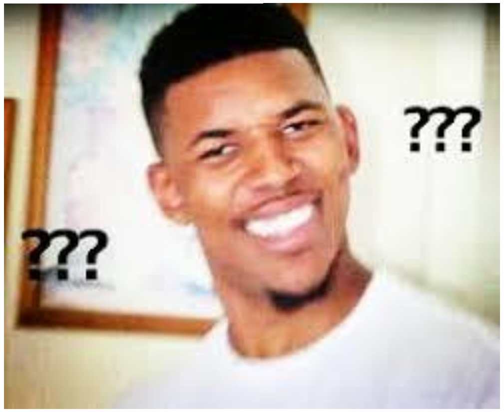
基本概念

- 二阶相似度
  - 二阶相似度描述了一对节点的邻域结构的接近程度。
  - 节点 v_i 和 v_j 之间的二阶相似度定义为：S_i 和 S_j 之间的相似性。

二阶相似性



$$s_{6,7} > 0$$
$$s_{5,6} = 0$$

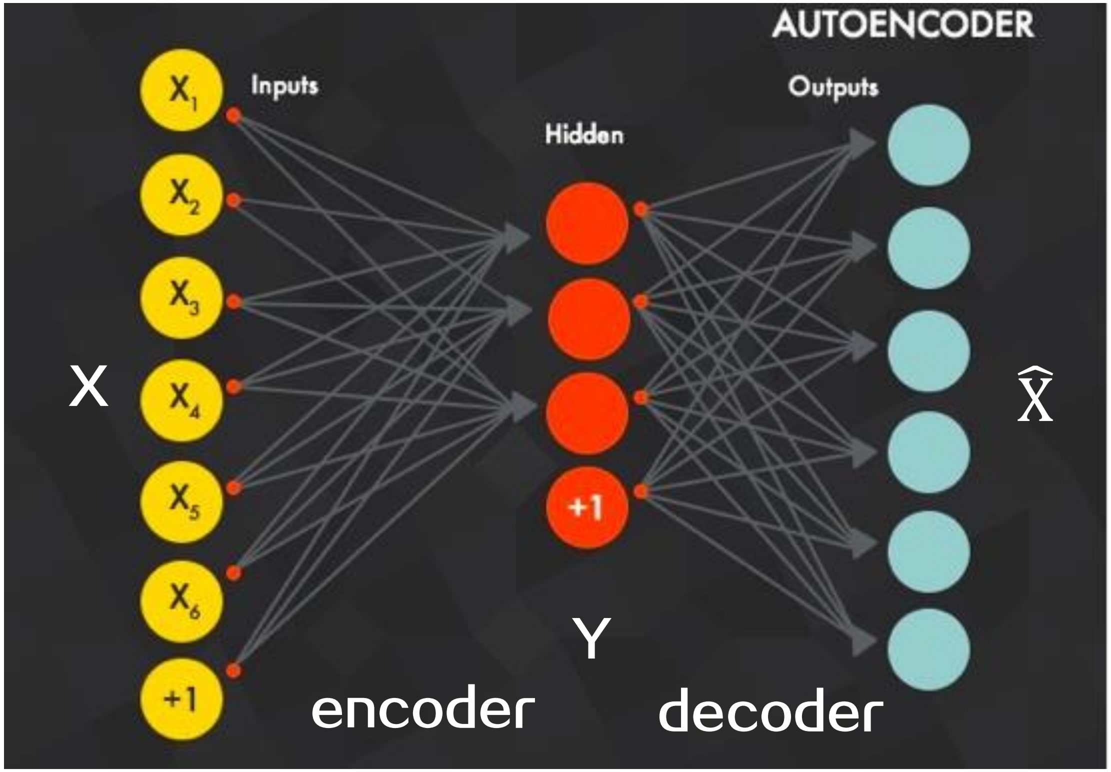


基本概念

- **网络嵌入**
  - 给定一个表示为$G=(V,E)$的图，网络嵌入旨在学习映射函数$f: v_i \longmapsto y_i \in \mathbb{R}^d$，其中$d \ll |V|$。函数$f$的目的是使$y_i$和$y_j$之间的相似性 保留了 v_i和v_j 之间定义的相似度度量（这里SDNE算法只保留一阶和二阶相似度）。
  - 这些低维的向量表示使得快速高效的算法设计成为可能，而不必再去考虑原本的网络结构。

基本概念

- 自编码器——用于数据降维
  - 无监督机器学习算法
  - 将 Input 压缩为 Hidden 的部分称为 encoder，将 Hidden 还原为 Input 的部分称为 decoder



基本概念

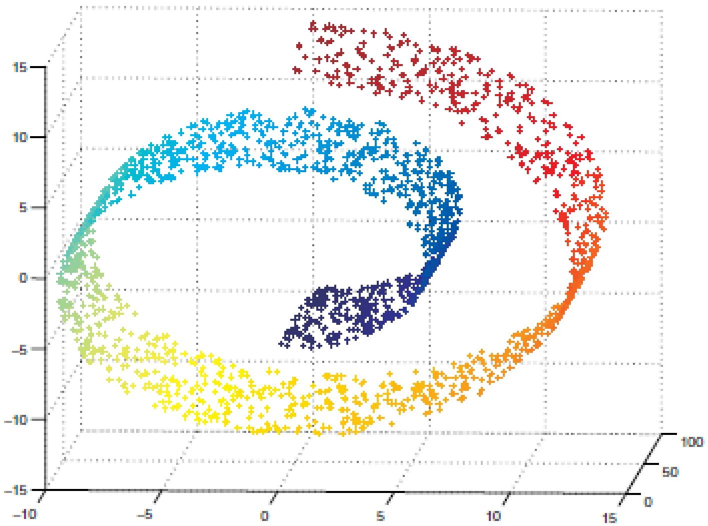
- 拉普拉斯特征映射(Laplacian Eigenmaps)——机器学习降维算法
  - 通过平滑项的方式，使得在原始空间中两个相似的节点，在低维的向量空间中有相近的表示。
  - 最小化的目标函数如下：

$$L_{1st} = \frac{1}{2}\sum_{i,j=1}^{n} \|Y_i - Y_j\|_2^2 W_{i,j} = tr(Y^T L Y)$$

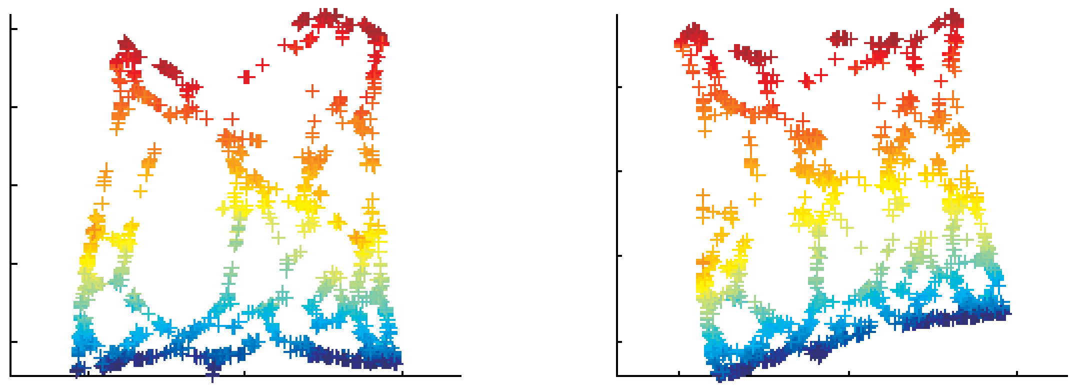
  - 其中$L = D - W$是图G的Laplacian矩阵, $D \in \mathbb{R}^{n\times n}$是一个对角矩阵，$D_{i,i} = \sum_j W_{i,j}$
  - 该优化问题可以转化为 Laplace 矩阵的特征向量计算问题

M. Belkin, P. Niyogi, Laplacian eigenmaps and spectral techniques for embedding and clustering, in: NIPS, 14, 2001, pp. 585 - 591.

- 机器学习降维算法——拉普拉斯特征映射(Laplacian Eigenmaps)



2000 Random data points on the swiss roll



N = 5　t = 5.0　　N = 10　t = 5.0

降维到2D

M. Belkin, P. Niyogi, Laplacian eigenmaps and spectral techniques for embedding and clustering, in: NIPS, 14, 2001, pp. 585 - 591.

算法原理

算法原理

P	如何对网络进行嵌入
C	基于深度模型
D	网络结构高度非线性； 网络表示需要保留其局部和全局的结构信息； 网络的稀疏性
L	KDD 2016

背景简介

- **提出问题**

- **解决办法**
  - 针对网络结构非线性，构建了多层非线性函数深度学习模型
  - 针对全局和局部结构保留以及稀疏性问题，同时利用一阶相似性和二阶相似性学习网络的局部结构信息和全局结构信息。

算法原理

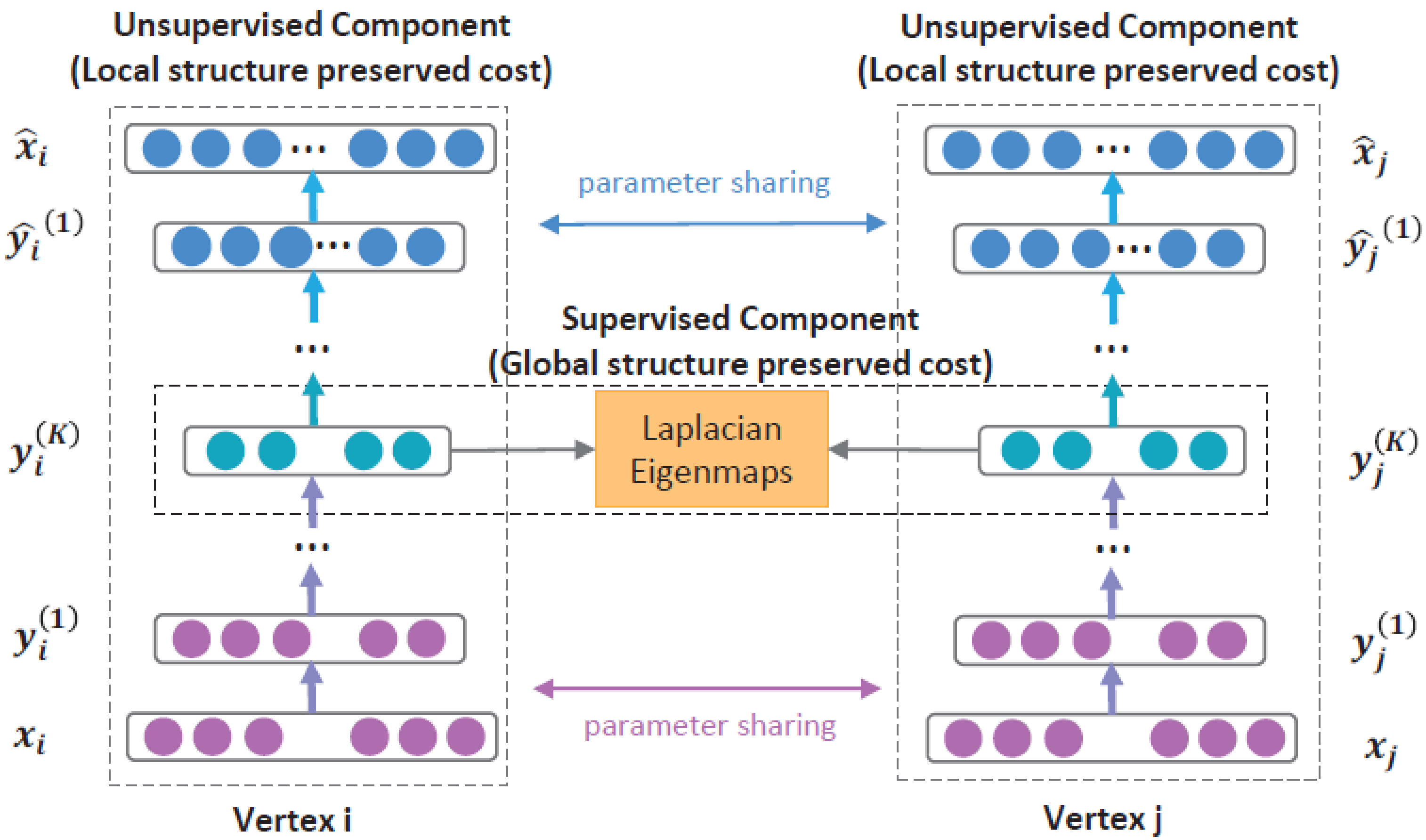
T	基于将网络中的节点映射成为低维向量表示
I	输入网络G(V;E)的邻接矩阵S；预训练模型参数

P	For {
	1. 对每个节点的邻接矩阵进行编码和解码重构
	2. 在每个节点最后一个编码层，加上了和其他节点的约束条件
	3. 最小化损失函数
	3. 更新模型
	}

O	网络向量表示Y

算法原理

- **算法原理图**


Unsupervised Component
(Local structure preserved cost)
Unsupervised Component
(Local structure preserved cost)
$\hat{x}_i$
$\hat{y}_i^{(1)}$
$y_i^{(K)}$
$y_i^{(1)}$
x_i
$\hat{x}_j$
$\hat{y}_j^{(1)}$
$y_j^{(K)}$
$y_j^{(1)}$
x_j
parameter sharing
Supervised Component
(Global structure preserved cost)
Laplacian
Eigenmaps
parameter sharing
Vertex i
Vertex j

算法原理

- **损失函数**

$$L_{mix} = L_{2nd} + \alpha L_{1st} + \nu L_{reg}$$

$$= \left\|(\hat{X} - X) \odot B\right\|_F^2 + \alpha \sum_{i,j=1}^{n} s_{i,j}\left\|y_i - y_j\right\|_2^2 + \nu L_{reg}$$

  - **其中L_{reg}是一个L2范数正则项，用于防止过拟合，其定义如下：**

$$L_{reg} = \frac{1}{2}\sum_{k=1}^{K}\left(\left\|W^{(k)}\right\|_F^2 + \left\|\widehat{W}^{(k)}\right\|_F^2\right)$$

算法原理

- 无监督部分如何提取网络二阶相似度的信息
  - 自编码器对输入进行编码，其中每一层的隐含表示计算为：

$$y_i^{(1)} = \sigma W^1 x_i + b^{(1)}$$

$$y_i^{(k)} = \sigma W^{(k)} y_i^{(k-1)} + b^{(k)}, k = 2, \cdots, K$$

  - 再用解码器对隐含层表示$y_i^{(k)}$进行解码重构得到$\hat{x}_i$
  - 原始自编码器的损失函数定义为：

$$L = \sum_{i=1}^{n} \|\hat{x}_i - x_j\|_2^2$$

算法原理

- 无监督部分如何提取网络二阶相似度的信息
  - 根据网络稀疏性，对非0值重构的惩罚重于对0值重构的惩罚，改进损失函数：

$$L_{2nd} = \sum_{i=1}^{n} \left\| (\hat{x}_i - x_j) \odot b_i \right\|_2^2 = \left\| (\hat{X} - X) \odot B \right\|_F^2$$

  - 其中, ⊙表示哈达马乘积（对应元素相乘）。当$s_{i,j} = 0$时, $b_{i,j} = 1$;当$s_{i,j} > 0$时, $b_{i,j}$为大于1的超参数。
  - 通过自编码器,如果两个节点具有相近的邻接点结构,则在嵌入后的向量表示空间中距离越近。

算法原理

- 有监督部分如何提取网络一阶相似度的信息
  - 拉普拉斯特征映射（Laplacian Eigenmaps）

$$L_{1st} = \sum_{i,j=1}^{n} s_{i,j} \left\| y_i - y_j \right\|_2^2 = 2tr(Y^T L Y)$$

  - 其中$L = D - S$, $D \in \mathbb{R}^{n \times n}$是一个对角矩阵，$D_{i,i} = \sum_j s_{i,j}$
  - 当两个相似的节点在嵌入空间中被映射到相距很远的位置时候，增加惩罚项

算法原理

- **损失函数**

$$L_{mix} = L_{2nd} + \alpha L_{1st} + \nu L_{reg}$$

$$= \left\|(\hat{X} - X) \odot B\right\|_F^2 + \alpha \sum_{i,j=1}^{n} s_{i,j}\left\|y_i - y_j\right\|_2^2 + \nu L_{reg}$$

 – **其中L_{reg}是一个L2范数正则项，用于防止过度拟合，其定义如下：**

$$L_{reg} = \frac{1}{2}\sum_{k=1}^{K}\left(\left\|W^{(k)}\right\|_F^2 + \left\|\widehat{W}^{(k)}\right\|_F^2\right)$$

算法原理

- ## 算法基本步骤和流程图

Algorithm 1 Training Algorithm for the semi-supervised deep model of *SDNE*

Input: the network $G = (V, E)$ with adjacency matrix S, the parameters α and ν

Output: Network representations Y and updated Parameters: θ

1: Pretrain the model through deep belief network to obtain the initialized parameters $\theta = \{\theta^{(1)}, ..., \theta^{(K)}\}$

2: $X = S$

3: **repeat**

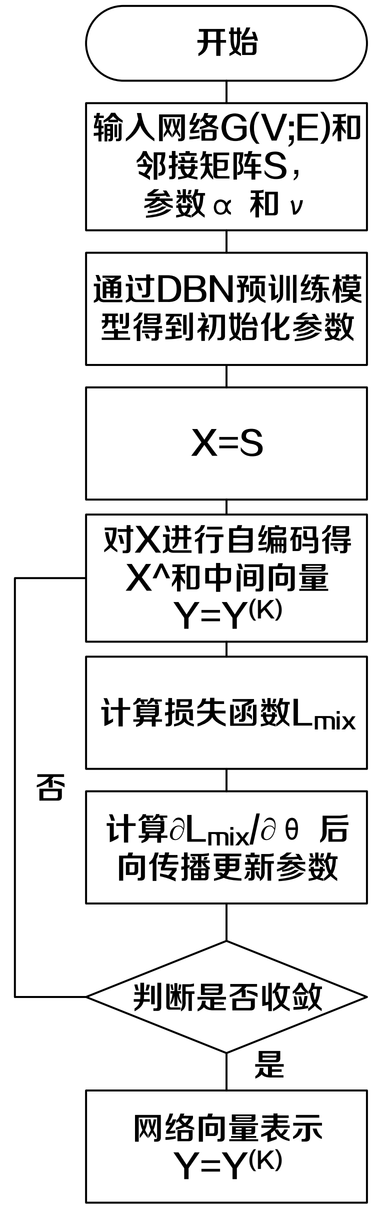
4: Based on X and θ, apply Eq. 1 to obtain $\hat{X}$ and $Y = Y^K$.

5: $\mathcal{L}_{mix}(X; \theta) = \|(\hat{X} - X) \odot B\|_F^2 + 2\alpha tr(Y^T L Y) + \nu \mathcal{L}_{reg}$.

6: Based on Eq. 6, use $\partial L_{mix} / \partial \theta$ to back-propagate through the entire network to get updated parameters θ.

7: **until** converge

8: Obtain the network representations $Y = Y^{(K)}$



算法原理

- 数据集：BlogCatalog，Flickr，YouTube，ARXIV GR-QC，20-NEWSGROUP
- 对比方法：DeepWalk，LINE，GraRep，Laplacian Eigenmaps
  - Multi-label Classification

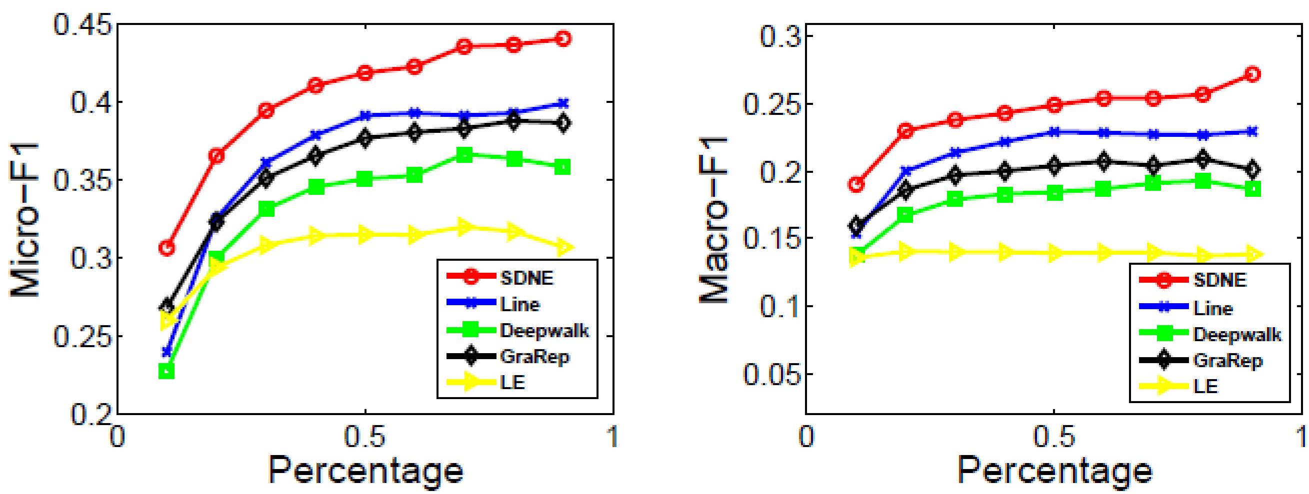


Figure 4: Micro-F1 and Macro-F1 on BLOGCATALOG.

优劣分析

- 优势
  - 之前的模型都为浅层模型, SDNE为深层网络嵌入模型
- 劣势
  - 当网络中添加新节点时，如果未观察到新节点与现有节点的连接，则无法给出新节点的向量表示

应用总结

- 算法的应用领域
  - 节点分类
  - 链接预测
  - 社区发现
  - 推荐系统
- 未来的发展
  - 动态网络嵌入

参考文献

[1] D. Wang, P. Cui, W. Zhu, Structural deep network embedding, in: Proceedings of the 22nd International Conference on Knowledge Discovery and Data Mining, ACM, 2016, pp. 1225 – 1234.

[2] M. Belkin, P. Niyogi, Laplacian eigenmaps and spectral techniques for embed- ding and clustering, in: NIPS, 14, 2001, pp. 585 – 591.

[3] 涂存超, 杨成, 刘知远,等. 网络表示学习综述[J]. 中国科学:信息科学, 2017(8).

[4]《Structural Deep Network Embedding》阅读笔记 https://zhuanlan.zhihu.com/p/24769965

谢谢！

大成若缺，其用不弊。大盈若冲，其用不穷。大直若屈。大巧若拙。大辩若讷。静胜躁，寒胜热。清静为天下正。