Network Embedding Summary

- GraRep: Learning Graph Representations with Global Structural Information.
- Discriminative Deep Random Walk for Network Classification
- CANE: Context-Aware Network Embedding for Relation Modeling
- Label Informed Attributed Network Embedding

张峻伟

GraRep: Learning Graph Representations with Global Structural Information.

CIKM 2015.

Shaosheng Cao, Wei Lu, Qiongkai Xu

工作：

1、提出了skip-gram模型的显式损失函数，将不同的K步关系保存在不同的子空间中，揭示了与图形有关的全局结构信息

2、使用矩阵分解优化每个模型，通过组合从不同模型学习的不同表示，为每个顶点构建全局表示。

3、整合了不同非线性的组合的损失函数

GraRep：一种用于学习加权图的顶点表示的新模型.

Deep Walk：不清楚他们的学习过程中所涉及的图上定义的确切损失函数是什么

LINE：定义了1-step和2-step关系信息的损失函数，但是扩展性差

Table 1: Overall Algorithm

GraRep Algorithm
Input Adjacency matrix S on graph Maximum transition step K Log shifted factor β Dimension of representation vector d

1. Get k-step transition probability matrix A^k
Compute $A = D^{-1}S$
Calculate $A^1, A^2, \ldots, A^K$, respectively
2. Get each k-step representations
For $k = 1$ to K
    2.1 Get positive log probability matrix
    calculate $\Gamma_1^k, \Gamma_2^k, \ldots, \Gamma_N^k$ ($\Gamma_j^k = \sum_p A_{p,j}^k$) respectively
    calculate $\{X_{i,j}^k\}$
$$X_{i,j}^k = \log\left(\frac{A_{i,j}^k}{\Gamma_j^k}\right) - \log(\beta)$$
    assign negative entries of X^k to 0
    2.2 Construct the representation vector W^k
    $[U^k, \Sigma^k, (V^k)^T] = SVD(X^k)$
    $W^k = U_d^k (\Sigma_d^k)^{\frac{1}{2}}$
End for
3. Concatenate all the k-step representations
$W = [W^1, W^2, \ldots, W^K]$

Output Matrix of the graph representation W

Discriminative Deep Random Walk for Network Classification

用于网络分类的判别式深度随机游走

ACL 2016

Juzheng Li, Jun Zhu, Bo Zhang

Dept. of Comp. Sci. & Tech., State Key Lab of Intell. Tech. & Sys. Tsinghua University,

判别式深度随机游走（DDRW）

- 嵌入目标

$$\min_{\boldsymbol{\theta}} \ \mathcal{L}_r(\boldsymbol{\theta}, \boldsymbol{\alpha}), \tag{1}$$

$\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_{|V|})$ 表示在潜在空间中嵌入的向量

$\boldsymbol{\alpha}$ 表示原图中的拓扑结构 $\mathcal{L}_r$ 表示嵌入表示和原拓扑结构之间的差距

然后使用随机游走和Word2Vec方法获得目标函数

$$\mathcal{L}_r(\boldsymbol{\theta}, \boldsymbol{\alpha}) = -\sum_{i=1}^{\tau} \frac{1}{s} \sum_{t=1}^{s} \sum_{-R \le j \le R, j \neq 0} \log p(\boldsymbol{\omega}_{i,t+j} | \boldsymbol{\omega}_{i,j}). \tag{2}$$

$$p(\boldsymbol{\omega}_O | \boldsymbol{\omega}_I) = \frac{\exp(\boldsymbol{\theta}_{\boldsymbol{\omega}_O}^T \hat{\boldsymbol{\theta}}_{\boldsymbol{\omega}_I})}{\sum_{i=1}^{|V|} \exp(\boldsymbol{\theta}_i^T \hat{\boldsymbol{\theta}}_{\boldsymbol{\omega}_I})}, \tag{3}$$

- 分类目标

$$\min_{\boldsymbol{\theta},\boldsymbol{\beta}} \mathcal{L}_c(\boldsymbol{\theta},\boldsymbol{\beta},\boldsymbol{y}). \tag{4}$$

$\boldsymbol{y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_{|V|})$ 表示标签，$\boldsymbol{\beta}$ 表示随后地分类器

我们选择被称为L2正则化和L2损失支持向量分类（SVC）的分类器

$$\begin{aligned}&\mathcal{L}_c(\boldsymbol{\theta},\boldsymbol{\beta},\boldsymbol{y})\\&=C\sum_{i=1}^{|V|}(\sigma(1-\boldsymbol{y}_i\boldsymbol{\beta}^T\boldsymbol{\theta}_i))^2+\frac{1}{2}\boldsymbol{\beta}^T\boldsymbol{\beta},\end{aligned} \tag{5}$$

其中，C为正则化参数，如果x>0，σ(x)=x，否则σ(x)=0

【1】Rong-En Fan 2008. LIBLINEAR.A library for large linear classification. *Journal of Machine Learning Research*

- 联合学习

我们方法的主要目标是对给定的网络中未标签的顶点分类。我们在中间嵌入，即潜在的表示网络结构的帮助下达到这个目标。我们同时优化第3.1节和第3.2节中的两个目标。特别的定义目标函数：

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\alpha}, \boldsymbol{y}) = \eta \mathcal{L}_r(\boldsymbol{\theta}, \boldsymbol{\alpha}) + \mathcal{L}_c(\boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{y})$$

其中，η 是平衡两个目标函数权重的重要参数。我们解决这个联合优化问题：

$$\min_{\boldsymbol{\theta}, \boldsymbol{\beta}} \ \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\alpha}, \boldsymbol{y}). \qquad (6)$$

CANE: Context-Aware Network Embedding for Relation Modeling

MACL 2017

Cunchao Tu
Tsinghua University

Han Liu
Northeastern University

Zhiyuan Liu
Tsinghua University

Maosong Sun
Tsinghua University

- 提出的框架

With above definitions, CANE aims to maximize the overall objective of edges as follows:

$$\mathcal{L} = \sum_{e \in E} L(e). \tag{1}$$

$$L(e) = L_s(e) + L_t(e), \tag{2}$$

where $L_s(e)$ denotes the structure-based objective and $L_t(e)$ represents the text-based objective.

Thus, the structure-based objective aims to measure the log-likelihood of a directed edge using the structure-based embeddings as

$$L_s(e) = w_{u,v} \log p(\mathbf{v}^s|\mathbf{u}^s). \tag{3}$$

Following LINE (Tang et al., 2015), we define the conditional probability of v generated by u in Eq. (3) as

$$p(\mathbf{v}^s|\mathbf{u}^s) = \frac{\exp(\mathbf{u}^s \cdot \mathbf{v}^s)}{\sum_{z \in V} \exp(\mathbf{u}^s \cdot \mathbf{z}^s)}. \tag{4}$$

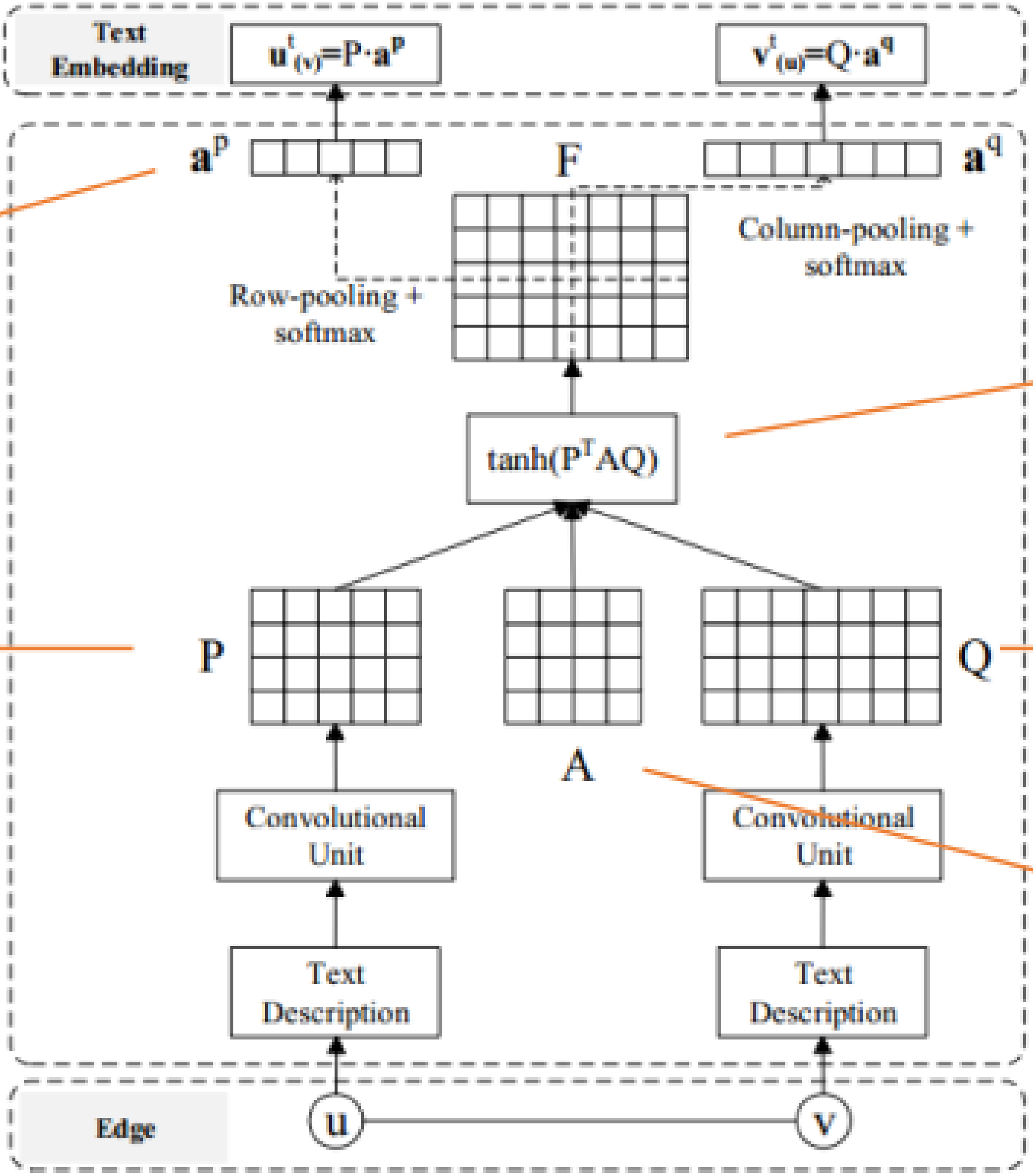
The text-based objective $L_t(e)$ can be defined with various measurements. To be compatible with $L_s(e)$, we define $L_t(e)$ as follows:

$$L_t(e) = \alpha \cdot L_{tt}(e) + \beta \cdot L_{ts}(e) + \gamma \cdot L_{st}(e), \quad (5)$$

where α, β and γ control the weights of various parts, and

$$\begin{aligned} L_{tt}(e) &= w_{u,v} \log p(\mathbf{v}^t|\mathbf{u}^t), \\ L_{ts}(e) &= w_{u,v} \log p(\mathbf{v}^t|\mathbf{u}^s), \\ L_{st}(e) &= w_{u,v} \log p(\mathbf{v}^s|\mathbf{u}^t). \end{aligned} \quad (6)$$

● 文本嵌入的方法（CNN）



$$\mathbf{u}_{(v)} = \mathbf{u}^s \oplus \mathbf{u}^t_{(v)} \text{ and } \mathbf{v}_{(u)} = \mathbf{v}^s \oplus \mathbf{v}^t_{(u)}$$

$$a_i^p = \frac{\exp(g_i^p)}{\sum_{j\in[1,m]} \exp(g_j^p)}.$$

correlation matrix **F**

$\mathbf{P} \in \mathbb{R}^{d\times \boxed{m}}$

$\mathbf{Q} \in \mathbb{R}^{d\times \boxed{n}}$

the lengths of two corresponding text sequences

Attentive matrix

Label Informed Attributed Network Embedding

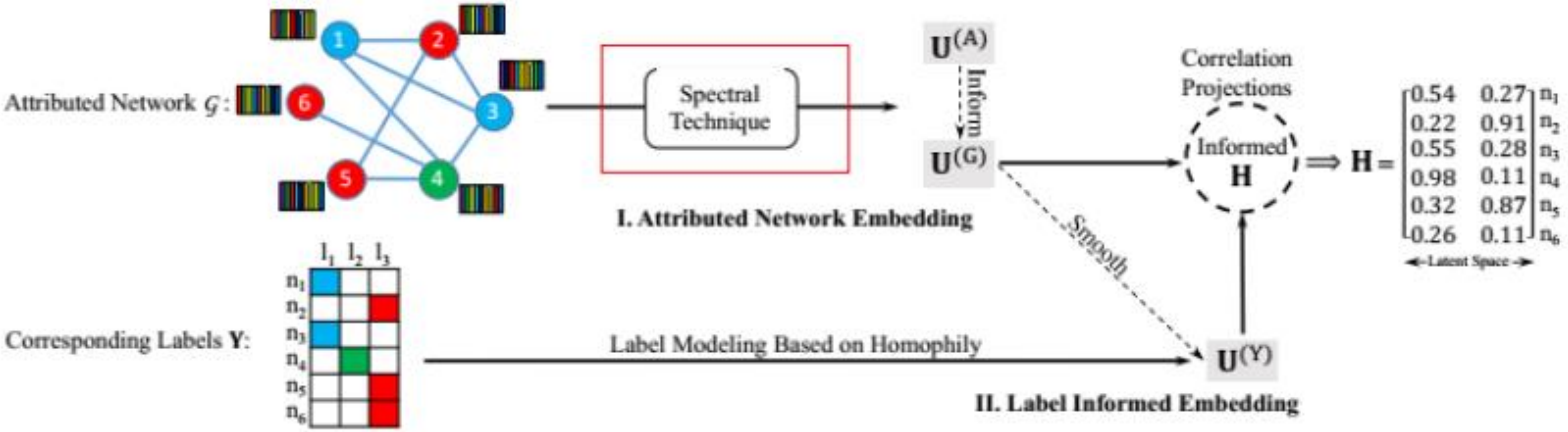
WSDM 2017.

Xiao Huang
Texas A&M University
College Station

Jundong Li
Arizona State University

Xia Hu
Texas A&M University
College Station

LANE


Attributed Network $\mathcal{G}$:
1
2
3
4
5
6
Spectral Technique
I. Attributed Network Embedding
$\mathbf{U}^{(A)}$
Inform
$\mathbf{U}^{(G)}$
Correlation Projections
Informed H
$\Longrightarrow \mathbf{H} =$
0.54 0.27 n_1
0.22 0.91 n_2
0.55 0.28 n_3
0.98 0.11 n_4
0.32 0.87 n_5
0.26 0.11 n_6
Latent Space
Smooth
Corresponding Labels $\mathbf{Y}$:
l_1 l_2 l_3
n_1 n_2 n_3 n_4 n_5 n_6
Label Modeling Based on Homophily
$\mathbf{U}^{(Y)}$
II. Label Informed Embedding

Algorithm 1: Label informed Attributed Network Embedding

Input: d, ϵ, $\mathcal{G}$, $\mathbf{Y}$.

Output: Embedding representation $\mathbf{H}$.

Construct the affinity matrices $\mathbf{S}^{(G)}$ and $\mathbf{S}^{(A)}$;
Compute Laplacian matrices $\mathcal{L}^{(G)}$, $\mathcal{L}^{(A)}$ and $\mathcal{L}^{(Y)}$;
Initialize $t = 1$, $\mathbf{U}^{(A)} = \mathbf{0}$, $\mathbf{U}^{(Y)} = \mathbf{0}$ and $\mathbf{H} = \mathbf{0}$;
repeat
    Update $\mathbf{U}^{(G)}$ by solving Eq. (13);
    Update $\mathbf{U}^{(A)}$ by solving Eq. (14);
    Update $\mathbf{U}^{(Y)}$ by solving Eq. (15);
    Update $\mathbf{H}$ by solving Eq. (16);
    $t = t + 1$;
until $\mathcal{J}_t - \mathcal{J}_{t-1} \leq \epsilon$;
return $\mathbf{H}$.

$$\mathcal{L}^{(G)} = \mathbf{D}^{(G)^{-\frac{1}{2}}}\mathbf{S}^{(G)}\mathbf{D}^{(G)^{-\frac{1}{2}}}$$

$$(\mathcal{L}^{(G)} + \alpha_1 \mathbf{U}^{(A)}\mathbf{U}^{(A)^T} + \alpha_2 \mathbf{U}^{(Y)}\mathbf{U}^{(Y)^T} + \mathbf{H}\mathbf{H}^T)\mathbf{U}^{(G)} = \boldsymbol{\lambda}_1 \mathbf{U}^{(G)}, \tag{13}$$

$$(\alpha_1 \mathcal{L}^{(A)} + \alpha_1 \mathbf{U}^{(G)}\mathbf{U}^{(G)^T} + \mathbf{H}\mathbf{H}^T)\mathbf{U}^{(A)} = \boldsymbol{\lambda}_2 \mathbf{U}^{(A)}, \tag{14}$$

$$(\alpha_2 \mathcal{L}^{(YY)} + \alpha_2 \mathbf{U}^{(G)}\mathbf{U}^{(G)^T} + \mathbf{H}\mathbf{H}^T)\mathbf{U}^{(Y)} = \boldsymbol{\lambda}_3 \mathbf{U}^{(Y)}, \tag{15}$$

$$(\mathbf{U}^{(G)}\mathbf{U}^{(G)^T} + \mathbf{U}^{(A)}\mathbf{U}^{(A)^T} + \mathbf{U}^{(Y)}\mathbf{U}^{(Y)^T})\mathbf{H} = \boldsymbol{\lambda}_4 \mathbf{H}. \tag{16}$$

Thank you