Graph Embeddings

Tim Semenov
timofei@sourced.tech
July 12, 2017

Table of contents

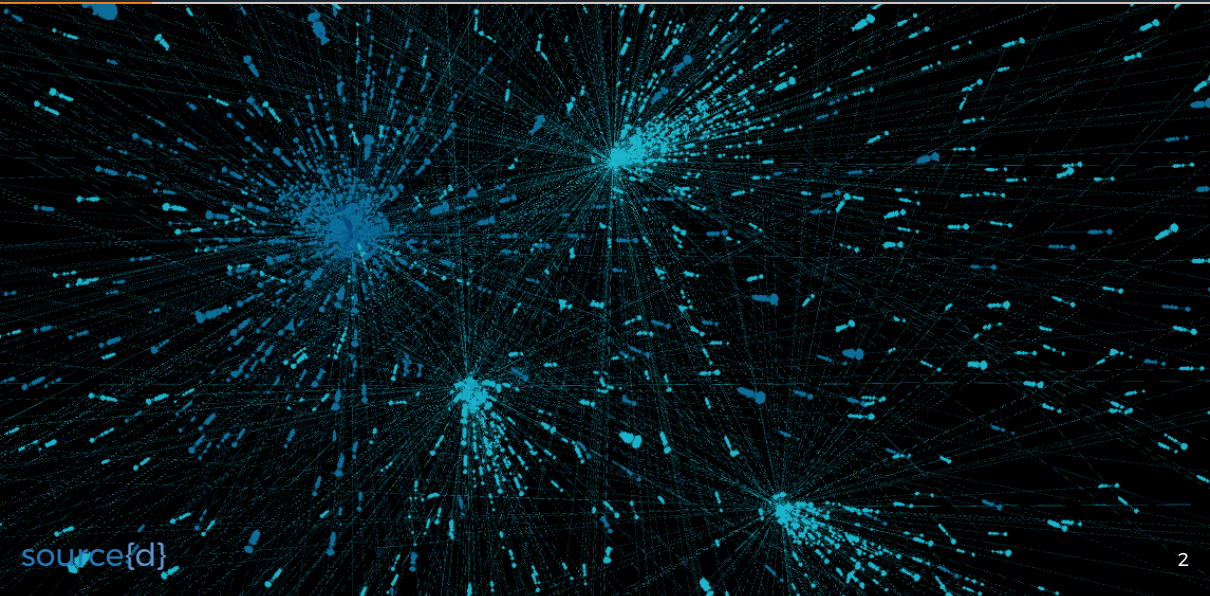
1. Introduction

2. Graph embeddings

3. Experiment results

4. Conclusion

Introduction



Notations

Adjacency matrix:	A
Degree matrix:	$D = diag(\sum_j A_{ij})$
Transition matrix:	$M = D^{-1}A$
Node neighborhood:	$N(i)$

Notations

Adjacency matrix:	A
Degree matrix:	$D = diag(\sum_j A_{ij})$
Transition matrix:	$M = D^{-1}A$
Node neighborhood:	$N(i)$
First-order proximity:	edges of adjacency matrix.
Second-order proximity:	similarity between nodes' neighborhoods.

Notations

Pointwise mutual information:

$$PMI(x, y) = \log \frac{P(x, y)}{P(x)P(y)}$$

Positive PMI:

$$PPMI(x, y) = \max(PMI(x, y), 0)$$

Shifted PPMI:

$$SPPMI_k(x, y) = \max(PMI(x, y) - \log k, 0)$$

Graph embedding methods

Taxonomy [2]:

1. Factorization based
2. Random Walk based
3. Deep Learning based
4. Other

Factorization

1. HOPE [4]
   - $\min ||S - U^S U^{t^T}||_F^2$
   - $U = [U^S, U^t]$ – the embedding matrix.
   - S – proximity matrix:
     - $S = A^2$ – common neighbors.
     - $S = ADA$ – Adamic-Adar.
     - …
2. GraRep [1]
   - $X_k = SPPMI_\beta(A^k),\ k = 1, \ldots, K$
   - $[U_k, \Sigma_k, V_k^T] = SVD(X_k)$
   - $W_k = U_{k,d}(\Sigma_{k,d})^{\frac{1}{2}}$

Random Walk: node2vec

Optimization problem [3]:

$$\max \sum_{v_i \in V} \log P(N(i)|v_i)$$

Conditional independence:

$$P(N(i)|v_i) = \prod_{j \in N(i)} P(v_j|v_i), \quad P(v_j|v_i) = \frac{\exp(u_j^T u_i)}{\sum_{k=1}^{|V|} \exp(u_k^T u_i)}$$

Random Walk: node2vec

Optimization problem [3]:

$$\max \sum_{v_i \in V} \log P(N(i)|v_i)$$

Conditional independence:

$$P(N(i)|v_i) = \prod_{j \in N(i)} P(v_j|v_i), \quad P(v_j|v_i) = \frac{\exp(u_j^T u_i)}{\sum_{k=1}^{|V|} \exp(u_k^T u_i)}$$

Simplified optimization problem:

$$\max \sum_{v_i \in V} \Big[-\log Z_{v_i} + \sum_{j \in N(i)} u_j^T u_i \Big], \quad Z_{v_i} = \sum_{v_i \in V} \exp(u_j^T u_i)$$

Random Walk: node2vec

Let c_i denote the i-th node in the random walk:

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

Assume the walk just traversed edge (t, v):

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

Other: LINE

Joint probability between vertices [5]:

$$P(v_i, v_j) = \frac{1}{1 + \exp(-u_i^T u_j)}, \quad \hat{P}(v_i, v_j) = \frac{w_{ij}}{W}$$

Conditional distribution of the contexts:

$$P(v_j|v_i) = \frac{\exp(u_j'^T u_i)}{\sum_{k=1}^{|V|} \exp(u_k'^T u_i)}, \quad \hat{P}(v_j, v_i) = \frac{w_{ij}}{d_i}$$

Other: LINE

Joint probability between vertices [5]:

$$P(v_i, v_j) = \frac{1}{1 + \exp(-u_i^T u_j)}, \quad \hat{P}(v_i, v_j) = \frac{w_{ij}}{W}$$

Conditional distribution of the contexts:

$$P(v_j|v_i) = \frac{\exp(u_j'^T u_i)}{\sum_{k=1}^{|V|} \exp(u_k'^T u_i)}, \quad \hat{P}(v_j, v_i) = \frac{w_{ij}}{d_i}$$

First-order proximity loss:

$$O_1 = -\sum_{(i,j)\in E} w_{ij} \log p_1(v_i, v_j)$$

Second-order proximity loss:

$$O_2 = -\sum_{(i,j)\in E} w_{ij} \log p_2(v_j|v_i)$$

source{d}

Other: LINE

Use negative sampling to optimize O_2:

$$\log \sigma(u_j'^T u_i) + \sum_{i=1}^{K} E_{v_n \sim P_n(v)}[\log \sigma(-u_n'^T u_i)]$$

O_1 has a trivial minima, so we modify it to utilize negative sampling:

$$\log \sigma(u_j^T u_i) + \sum_{i=1}^{K} E_{v_n \sim P_n(v)}[\log \sigma(-u_n'^T u_i)]$$

Other: LINE

Use negative sampling to optimize O_2:

$$\log \sigma(u_j'^T u_i) + \sum_{i=1}^{K} E_{v_n \sim P_n(v)}[\log \sigma(-u_n'^T u_i)]$$

O_1 has a trivial minima, so we modify it to utilize negative sampling:

$$\log \sigma(u_j^T u_i) + \sum_{i=1}^{K} E_{v_n \sim P_n(v)}[\log \sigma(-u_n'^T u_i)]$$

In case of low degree we add weight to second-order neighbors:

$$w_{ij} = \sum_{k \in N(i)} w_{ik} \frac{w_{kj}}{d_k}$$

Deep Learning: SDNE

Encoder-decoder model [6]:

$$y_i^{(k)} = \sigma(W^{(k)} y_i^{(k-1)} + b^{(k)}),\ k = 1, \ldots, K,\ y_i^{(0)} = x_i$$

$$\hat{y}_i^{(k)} = \sigma(\hat{W}^{(k)} \hat{y}_i^{(k-1)} + \hat{b}^{(k)}),\ k = 1, \ldots, K,\ \hat{y}_i^{(0)} = y_i^{(K)},\ \hat{y}_i^{(K)} = \hat{x}_i$$

Deep Learning: SDNE

Encoder-decoder model [6]:

$$y_i^{(k)} = \sigma(W^{(k)} y_i^{(k-1)} + b^{(k)}),\ k = 1, \ldots, K,\ y_i^{(0)} = x_i$$

$$\hat{y}_i^{(k)} = \sigma(\hat{W}^{(k)} \hat{y}_i^{(k-1)} + \hat{b}^{(k)}),\ k = 1, \ldots, K,\ \hat{y}_i^{(0)} = y_i^{(K)},\ \hat{y}_i^{(K)} = \hat{x}_i$$

Loss functions:

$$L_1 = \sum_{i,j=1}^{n} a_{i,j} ||y_i^{(k)} - y_j^{(k)}||_2^2$$

Deep Learning: SDNE

Encoder-decoder model [6]:

$$y_i^{(k)} = \sigma(W^{(k)} y_i^{(k-1)} + b^{(k)}),\ k = 1, \ldots, K,\ y_i^{(0)} = x_i$$

$$\hat{y}_i^{(k)} = \sigma(\hat{W}^{(k)} \hat{y}_i^{(k-1)} + \hat{b}^{(k)}),\ k = 1, \ldots, K,\ \hat{y}_i^{(0)} = y_i^{(K)},\ \hat{y}_i^{(K)} = \hat{x}_i$$

Loss functions:

$$L_1 = \sum_{i,j=1}^{n} a_{i,j} ||y_i^{(k)} - y_j^{(k)}||_2^2$$

$$L_2 = \sum_{i=1}^{n} ||(\hat{x}_i - x_i) \odot b_i||_2^2, \quad b_{i,j} = \begin{cases} 1 & \text{if } a_{i,j} = 0 \\ \beta > 1 & \end{cases}$$

Deep Learning: SDNE

Encoder-decoder model [6]:

$$y_i^{(k)} = \sigma(W^{(k)} y_i^{(k-1)} + b^{(k)}),\ k = 1, \ldots, K,\ y_i^{(0)} = x_i$$

$$\hat{y}_i^{(k)} = \sigma(\hat{W}^{(k)} \hat{y}_i^{(k-1)} + \hat{b}^{(k)}),\ k = 1, \ldots, K,\ \hat{y}_i^{(0)} = y_i^{(K)},\ \hat{y}_i^{(K)} = \hat{x}_i$$

Loss functions:

$$L_1 = \sum_{i,j=1}^{n} a_{i,j} ||y_i^{(k)} - y_j^{(k)}||_2^2$$

$$L_2 = \sum_{i=1}^{n} ||(\hat{x}_i - x_i) \odot b_i||_2^2, \quad b_{i,j} = \begin{cases} 1 & \text{if } a_{i,j} = 0 \\ \beta > 1 & \end{cases}$$

$$L = L_2 + \alpha L_1 + \nu L_{reg}, \quad L_{reg} = \frac{1}{2} \sum_{k=1}^{K} (||W^{(k)}||_F^2 + ||\hat{W}^{(k)}||_F^2)$$

Datasets

- Blogcatalog, Flickr and Youtube
  Social networks of online users. Each user is labelled by at least one category.
- Arxiv GR-QC
  Paper collaboration network which covers papers in the field of General Relativity and Quantum Cosmology from arXiv.
- 20-Newsgroup
  Tf-idf vectors of each word are used to represent documents. The documents are connected based on their cosine similarity.

Datasets

Dataset	|V|	|E|
Blogcatalog	10312	667966
Flickr	80513	11799764
Youtube	1138499	5980886
Arxiv GR-QC	5242	28980
20-Newsgroup	1720	Full-connected

Experiment: reconstruction task

Arxiv GR-QC [6]

	SDNE	GraRep	LINE	DeepWalk
MAP	0.836	0.05	0.69	0.58

Blogcatalog [6]

	SDNE	GraRep	LINE	DeepWalk
MAP	0.63	0.42	0.58	0.28

Experiment: link prediction

Arxiv GR-QC [6]

Algorithm	P@2	P@10	P@100	P@200	P@300
SDNE	1	1	1	1	1
LINE	1	1	1	1	0.99
DeepWalk	1	0.8	0.6	0.555	0.443
GraRep	1	0.2	0.04	0.035	0.033

Summary

Summary

- Use matrix factorization when you can.

Summary

- Use matrix factorization when you can.
- Use random walk when matrix size is too large.

Summary

- Use matrix factorization when you can.
- Use random walk when matrix size is too large.
- Use deeper models on top of rough embeddings.

Summary

- Use matrix factorization when you can.
- Use random walk when matrix size is too large.
- Use deeper models on top of rough embeddings.

Follow us on github:

https://github.com/src-d/role2vec

References i

[1] S. Cao, W. Lu, and Q. Xu.
Grarep: Learning graph representations with global structural information.
In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, pages 891–900, New York, NY, USA, 2015. ACM.

[2] P. Goyal and E. Ferrara.
Graph embedding techniques, applications, and performance: A survey.
arXiv preprint arXiv:1705.02801, 2017.

[3] A. Grover and J. Leskovec.
node2vec: Scalable feature learning for networks.
In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864. ACM, 2016.

References ii

[4] M. Ou, P. Cui, J. Pei, Z. Zhang, and W. Zhu.
Asymmetric transitivity preserving graph embedding.
In *KDD*, pages 1105–1114, 2016.

[5] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei.
LINE: Large-scale Information Network Embedding.
ArXiv e-prints, Mar. 2015.

[6] D. Wang, P. Cui, and W. Zhu.
Structural deep network embedding.
In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1225–1234. ACM, 2016.