LINE: Large-scale Information Network Embedding

- Published in WWW'15
- Link: LINE (https://arxiv.org/abs/1503.03578)

作者

> Jian, Tang
> Meng, Qu
> Mingzhe, Wang
> Ming, Zhang
> Jun, Yan
> Qiaozhu, Mei

研究目的

- 針對大型網絡的 embedding 處理 (over millions of nodes and billions of edges) 且在運算時也保持高效率

研究方式

資料結構

- non-negative weighted graph

演算法

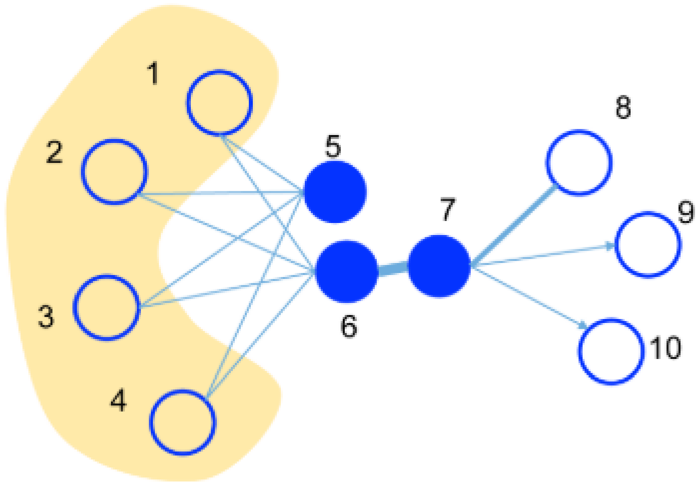

Figure 1: A toy example of information network. Edges can be undirected, directed, and/or weighted. Vertex 6 and 7 should be placed closely in the low-dimensional space as they are connected through a strong tie. Vertex 5 and 6 should also be placed closely as they share similar neighbors.

- An objective function that preserves both the local and global network structures
    - first-order proximity: 保存局部性的結構 (through simple connection incidence), v_i and v_j 間的一階相似 (first-order proximity, distribution) 為:
$$p_1(v_i, v_j) = \frac{1}{1 + \exp(-\vec{u_i}^T \cdot \vec{u_j})}$$
where $\vec{u_i} \in R^d$ 是 vertex v_i 的低維表徵
        - the empirical probability is $\hat{p}_1(i, j) = \frac{w_{ij}}{W}$ where $W = \sum_{(i,j)\in E} w_{ij}$
        - edge is w_{ij} or 0 if no edge
        - objective function
$$O_1 = d(\hat{p}_1(\cdot, \cdot), p_1(\cdot, \cdot))$$
        - 最後選擇 minimize KL-divergence (只能用在 undirected graph 上)
$$O_1 = -\sum_{(i,j)\in E} w_{ij} \log p_1(v_i, v_j)$$
    - second-order proximity: 保存全局性的結構 (through the shared neighborhood structures of the vertices), 將 vertex 想像成是文本，根據 vertex 前後的鄰居組合來找出相似的 vertex, 對 directed edge (i, j) 而言, v_i 和 v_j 的二階相似 (second-order proxiconditional distribution) 為
$$p_2(v_j|v_i) = \frac{\exp(\vec{u_j}'^T \cdot \vec{u_i})}{\sum_{k=1}^{|V|} \exp(\vec{u_k}'^T \cdot \vec{u_i})}$$
where $|V|$ is the number of vertices
        - the empirical distribution is $\hat{p}_2(\cdot|v_i)$
        - objective function
$$O_2 = \sum_{i\in V} \lambda_i d(\hat{p}_2(\cdot|v_i), p_2(\cdot|v_i))$$

其中 λ_i 是為了解決 vertex 在圖中可能有不同的重要程度，可透過 degree or PageRank 等方式解決

- 最後一樣選擇 minimize KL-divergence

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

- let $p_u = (w_{u,1}, \ldots w_{u,|V|})$ denotes the first-proximity of u with all the other vertices, u 和 v 的二階相似可透過 p_u 和 p_v 所衡量
- 本文中一、二階相似是分開訓練再合併起來使用
- 利用 negative sampling 來提高運算效率 (Ref-1 (https://python5566.wordpress.com/2018/03/17/nlp-%E7%AD%86%E8%A8%98-negative-sampling/))
  - word2vec 中使用
  - 對高頻單詞進行抽樣
  - 取出 one-hot=False 的詞，不全部更新，在每個訓練樣本只會更新一小部分的模型權重，進而降低計算量

- 提高演算法效率:
  - 傳統上是使用 stochastic gradient descent 的方式，但是現實中的圖是 weighted 的，具有高 variance 的情形下不適用 -> hard to define learning rate and might cause gradient explosion
  - 本文提出 edge-sampling algorithm: 1) sample the edges based on the probabilities proportional to their weights 2) 將選中的 edge 看作是 binary edges
  - 根據邊的權重進行採樣 -> alias method (Ref-2 (https://blog.csdn.net/haolexiao/article/details/65157026))

研究貢獻

- LINE: 可以針對各種圖進行 embedding (undirected, directed, and/or weighted)
  - objective funtion
  - edge-sampling algorithm
- 跟 graph factorization (GF) 技術相比，更能保有全局性的資訊 (GF 更多的注重 first-order proximity)
- 跟 DeepWalk (DW) 相比，DW 沒有針對 second-order proximity 進行進一步的處理，較接近 Depth-first search (本文是接近 Breadth-first search)

參考

- LINE: Large-scale Information Network Embedding - https://arxiv.org/abs/1503.03578 (https://arxiv.org/abs/1503.03578)
- Negative sampling - https://python5566.wordpress.com/2018/03/17/nlp-筆記-negative-sampling/ (https://python5566.wordpress.com/2018/03/17/nlp-%E7%AD%86%E8%A8%98-

negative-sampling/)

- Alias method - https://blog.csdn.net/haolexiao/article/details/65157026 (https://blog.csdn.net/haolexiao/article/details/65157026)