A response to "Link prediction using low-dimensional node embeddings: The measurement problem"

Jean, Pierre Both[1,*] and Jianshu Zhao[2]

[1]Université Paris-Saclay, CEA, List, Palaiseau, France. (Retired)
[2]Center for Bioinformatics and Computational Genomics, Georgia Institute of Technology, Atlanta, Georgia, USA
[*]Corresponding author : jeanpierre.both@gmail.com

April 2024

1 Introduction

Menand and Seshadhri (2024) recently developed a new metric to evaluate link prediction task in graph representation learning, which is called $VCMPR@k$. They claimed that the widely used metric AUC (area under curve) is biased because low-dimensional vectors cannot capture sparse ground truth using dot product similarities, a standard practice in graph representation learning. It turns out that $VCMPR@k$ shows much lower values, less than 0.3 for many graphs while AUC are all 0.9 above. Although their theoretical framework and mathematical derivation are appealing and elegant, we argue that their method suffers from practical problems with respect to real-world network or graph structures.

2 Low value bias of VCMPR@k

Here we identify from a practical perspective factors biasing $vcmpr@k$ to low values. First we recall that many real-world graphs, nodes degrees follow a power law distribution. So even if the mean degree is in orders of tens, a large fraction of nodes may have degree much lower. For example, the Blog catalog graph has mean degree 64.7 but 50% of nodes have degree less than 21 and 25% less than 8. In the high degree region 1% of nodes have degrees above 700. (Quantiles are given at Degrees). A node with no incident edge deleted contributes to 0 in the $VCMPR@k$ formula $VCMPR_i(k) = \frac{t_i(k)}{\min(d_i,k)}$. With an edge deletion probability p of 0.2 on average, only nodes of degree greater or equal than 5 can contribute significantly. Precisely the binomial distribution implies that nodes of degree less than 4 will contribute 0 with probability at least 0.41.
For the Blog catalog graph, we can expect around 5% of nodes contributes to 0 although there is no deleted edge to find (simulation shows it is 10%). For the Amazon graph we can expect 50% of nodes to have zero contribution (simulation shows it is 40%). In the same way: if the number of deleted edges is less than $min(d_i, k)$, which is expected for $p \cdot d_i \leq k$, the score of a node will be less than 1 even if all edges are retrieved. If a large proportion of nodes satisfy $d_i \leq k$ we should expect $vcmpr$ to be around p. (For example in the Amazon graph 90% of nodes have degree less than 10, so we could expect a result at the order of 0.9*0.2+1.0 *0.1).

We can try to estimate an upper bound of the expectation of $vcmpr$ supposing a uniform sampling of nodes as in the paper. We note r_i the number of deleted edge incidents to node i. On the set of nodes $\{i|d_i > k\}$ we have $\nu_i = \frac{t_i}{k}$ and on the set $\{i|d_i \leq k\}$ we have $\nu_i = \frac{t_i}{d_i} \leq \frac{r_i}{d_i}$. On

the average (depending on the degree) we have $r_i \leq p \cdot d_i$. We introduce an integer l and split the set $\{i|d_i > k\}$ into $\{i|l.k \geq d_i > k\}$ and $\{i|d_i > l \cdot k\}$. Now we have the following conditional expectations:

- on $\{i|d_i \leq k\}$ we have $\nu_i \leq \frac{p.d_i}{d_i} = p$
- on $\{i|l.k > d_i > k\}$ we have $\nu_i = \frac{t_i}{k} \leq \frac{p.d_i}{k} \leq \frac{l.p.k}{k} = l \cdot p$ with probability $P(d_i \leq l \cdot k) - P(d_i \geq k)$
- on $\{i|d_i > k \cdot l\}$ we have $\nu_i \leq \frac{t_i}{k} \leq 1$ with probability $P(d_i > l \cdot k)$

We can thus expect $vcmpr \leq p \cdot P(d_i \leq k) + p \cdot l \cdot P(l.k \geq d_i > k) + P(d_i > l \cdot k)$. Plugging degrees quantiles of the Blog and Amazon graphs with $p = 0.2, k = 10$. For the Blog graph with $l = 3$ we get $vcmpr \leq 0.2 \cdot 0.3 + 0.2 \cdot 3 \cdot 0.3 + (1.0 - 0.575) = 0.665$. For the Amazon graph with $l = 1$ we get $vcmpr \leq 0.2 \cdot 0.925 + 0.2 \cdot 1 \cdot 0. + (1.0 - 0.875) \leq 0.32$. Alternatively a Julia script in Degrees/e_vcmpr.jl can compute expectations over quantiles degrees.

We proposed a modified version of *vcmpr*: $\widetilde{vcmpr} = \frac{t_i(k)}{\min(r_i, d_i, k)}$, for nodes with $r_i > 0$. For nodes with $r_i = 0$, they cannot contribute to $\widetilde{vcmpr}$ as there are no edges to retrieve. This version of *vcmpr* is larger than the orignal *vcmpr*. However, it is not yet comparable to AUC as it fixes only the problem of nodes with no edge deleted. precision. Therefore, we propose the following new metric.

3 A new metric: centric AUC

We let i be a node, d_i be its degree, r_i the number of deleted edges incident to i. So, after edge deletion, a node i has degree $d_i - r_i$ and we have $n - 1 - (d_i - r_i)$ potential edges to test Given a node i we sort the n-1 edges by decreasing prediction of existence of a true edge between i and j. We define c_j as the number of true edges seen up to j. When exploring the j-th node in the sorted array there are 2 possibilities:

- j corresponds to a true (train edge), we increment c_j by 1
- j corresponds to a deleted edge. As our array is sorted, the probability that this edge has a higher ranking than a random edge is just the ratio between the number of indexes greater than j that do not correspond to a true edge already found and the number of potential edges. It is simply the ratio of potential edges after j to the total number of potential edges $\frac{n-j-(d_i-r_i(-c_j))}{n-1-(d_i-r_i)}$

Averaging over j,this defines our centric AUC at i. Averaging over uniformly sampled nodes i to get a centric AUC. We tested this centric AUC with 2 embedding algorithms implemented in our software graphembed. The first is HOPE (Ou et al. 2016) and relies on a SVD, and the second, called NodeSketch (Yang et al. 2019), relies on sketching. Centric AUC should work equally well for other graph embedding algorithms.

3.1 Results

Results of some graph embeddings are given in Table 1 below. The Blog graph has a smaller centric AUC than the global AUC in each embedding but in a less pronounced way than that of *vcmpr*. The Amazon and Dblp graph has a similar centric AUC with global AUC in each embedding.

Table 1: Embedding results and AUC

Graph	Algo Parameters	$\widetilde{vcmpr}$	Centric AUC	Global AUC
A. Sketching results. Parameters given in a triplet: dimension, nbhops, decay				
Blog	1000,5,0.5	$0.054 \pm 3.5 \times 10^{-3}$	$0.681 \pm 5.9 \times 10^{-3}$	0.93
Amazon	1000,5,0.5	$0.54 \pm 1.3 \times 10^{-2}$	$0.961 \pm 4 \times 10^{-3}$	$0.978 \pm 3.2 \times 10^{-4}$
Dblp	1000,5,0.5	$0.58 \pm 1.3 \times 10^{-2}$	$0.918 \pm 6 \times 10^{-3}$	$0.901 \pm 6.6 \times 10^{-4}$
Dblp	400,4,0.4	$0.574 \pm 1.1 \times 10^{-2}$	$0.94 \pm 5.1 \times 10^{-3}$	$0.961 \pm 4.4 \times 10^{-4}$
B. Hope embedding. Parameters given in a couple: dimension, nb iterations				
Blog	400,5	$0.17 \pm 5.4 \times 10^{-3}$	$0.698 \pm 7.4 \times 10^{-3}$	0.952
Amazon	400,5	$5.7 \times 10^{-2} \pm 6.3 \times 10^{-2}$	$0.834 \pm 6{,}4 \cdot 10^{-3}$	$0.856 \pm 9.1 \times 10^{-4}$

4 Conclusion

The gap between $vcmpr@k$ and AUC seems more related to the gap between precision and AUC than to its centric aspect. It must also be emphasized that computing a centric quality index by sampling only a limited number of nodes (a few thousands) due to the sorting cost of distances in large graphs is statistically difficult.

References

N. Menand and C. Seshadhri. Link prediction using low-dimensional node embeddings: The measurement problem. *Proceedings of the National Academy of Sciences 121, e2312527121*, 2024.

M. Ou, P. Cui, J. Pei, Z. Zhanga, and W. Zhu. Asymmetric transitivity preserving graph embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1105–1114, 2016.

D. Yang, P. Rosso, B. Li, and P. Cudre-Mauroux. Nodesketch: Highly-efficient graph embeddings via recursive sketching. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1162–1172, 2019.