# A SCALABLE UNSUPERVISED FRAMEWORK FOR COMPARING GRAPH EMBEDDINGS

*Bogumił Kamiński, Paweł Prałat and François Théberge*

SIAM Workshop on Network Science 2020
July 9–10 · Toronto

## Summary

There are many algorithms to embed graphs into vector space, and most of them have parameters to set. We provide a framework to efficiently compare the quality of embeddings in a purely unsupervised way.

## Introduction

A graph embedding is a mapping of the vertices of a graph into $k$-dimensional vectors. Good embeddings should capture the graph topology and vertex-to-vertex relationship. Several graph embedding algorithms are available and for each algorithm, parameters need to be set such as the dimension of the embedding space. As a result, selecting the best embedding is a challenging task. We propose an unsupervised framework to compare the quality of different embeddings for a given graph. The framework relies on two main ingredients: (i) a good, stable graph clustering algorithm; we use the ECG algorithms detailed in [4], and (ii) a generalization of the Chung-Lu model for graphs which incorporates the geometry provided by the graph embedding.

## Geometric Chung-Lu Model

In the Chung-Lu model [1], given some degree distribution $\mathbf{w} = (w_1, \ldots, w_n)$ over $n$ vertices $v_1, \ldots, v_n$, edge probabilities of a generated graph are defined such that the expected degrees for the vertices agree with this distribution. In our proposed *Geometric Chung-Lu* model (GCL), we also consider an embedding of the vertices of $G$ in some $k$-dimensional space $\mathcal{E} : V \to \mathbb{R}^k$. In particular, for each pair of vertices, $v_i$, $v_j$, we know the distance between them: $\text{dist}(\mathcal{E}(v_i), \mathcal{E}(v_j))$. We consider $0 \le d_{i,j} \le 1$, a normalized version of those distances. The probability that $v_i$ and $v_j$ are adjacent is proportional to $s(d_{i,j})$, a decreasing function $s$. For some choice of $\alpha \in [0, \infty)$, we define $s(d_{i,j}) := (1 - d_{i,j})^\alpha$ for all $d_{i,j}$'s. This choice gives us a good variety of functions to choose from. Choosing a large value for $\alpha$ makes it less probable to have long edges in embedded space. For a small value for $\alpha$, the distance in embedded space has less importance, and it is completely ignored when $\alpha = 0$.

The GCL model is the random graph $\mathcal{G}(\mathbf{w}, \mathcal{E}, \alpha)$ on the vertex set $V = \{v_1, \ldots, v_n\}$ in which each pair of vertices $v_i, v_j$, independently of other pairs, forms an edge with probability $p_{i,j}$, where $p_{i,j} = x_i x_j s(d_{i,j})$ for some learned weights $x_i \in \mathbb{R}_+$. The weights are such that the expected degree of $v_i$ is $w_i = \deg_G(v_i)$ for all $1 \le i \le n$. We show in [2] that there exists a unique selection of weights $x_i$, provided that the maximum degree of $G$ is less than the sum of degrees of all other vertices. Moreover, we show how to efficiently compute those weights numerically to any desired precision.

## The Framework

Given a graph $G = (V, E)$, its degree distribution $\mathbf{w}$ on $V$, and an embedding $\mathcal{E} : V \to \mathbb{R}^k$ of its vertices in $k$-dimensional space, we perform the five steps detailed below to obtain $\Delta_{\mathcal{E}}(G)$, a *divergence score* for the embedding. We can apply this algorithm to compare several embeddings $\mathcal{E}_1, \ldots, \mathcal{E}_m$, and select the best one via $\text{argmin}_i \Delta_{\mathcal{E}_i}(G)$.

**Step 1:** Run some stable *graph* clustering algorithm on $G$ to obtain a partition $\mathbf{C}$ of the vertex set $V$ into $\ell$ communities $C_1, \ldots, C_\ell$.

**Step 2:** For each $1 \le i \le \ell$, let $c_i$ be the proportion of edges of $G$ with both endpoints in $C_i$. Similarly, for each $1 \le i < j \le \ell$, let $c_{i,j}$ be the proportion of edges of $G$ with one endpoint in $C_i$ and the other one in $C_j$. Define:

$$\bar{\mathbf{c}} = (c_{1,2}, \ldots, c_{1,\ell}, c_{2,3}, \ldots, c_{\ell-1,\ell}), \quad \hat{\mathbf{c}} = (c_1, \ldots, c_\ell) \quad (1)$$

These *graph-based vectors* characterize the partition $\mathbf{C}$ from the perspective of $G$.

**Step 3:** Given $\alpha \in \mathbb{R}_+$ and vertex partition $\mathbf{C}$, consider $\mathcal{G}(\mathbf{w}, \mathcal{E}, \alpha)$, the GCL model. For each $1 \le i < j \le \ell$, we compute $b_{i,j}$, the expected proportion of edges of $\mathcal{G}(\mathbf{w}, \mathcal{E}, \alpha)$ with one endpoint in $C_i$ and the other one in $C_j$. Similarly, for each $1 \le i \le \ell$, let $b_i$ be the expected proportion of edges within $C_i$. We get:

$$\begin{aligned}\bar{\mathbf{b}}_{\mathcal{E}}(\alpha) &= (b_{1,2},\ldots,b_{1,\ell},b_{2,3},\ldots,b_{\ell-1,\ell}),\\ \hat{\mathbf{b}}_{\mathcal{E}}(\alpha) &= (b_1,\ldots,b_\ell).\end{aligned} \quad (2)$$

These *GCL-based vectors* characterizes partition $\mathbf{C}$ from the perspective of the embedding $\mathcal{E}$.

**Step 4:** We use the Jensen-Shannon divergenceto measure the dissimilarity $\Delta_\alpha$ between the vectors obtained in (1) and (2).

**Step 5:** Run steps 3 and 4 for several choices of $\alpha$. Let $\hat{\alpha} = \text{argmin}_\alpha \Delta_\alpha$. We define the *divergence score* for embedding $\mathcal{E}$ on $G$ as: $\Delta_{\mathcal{E}}(G) = \Delta_{\hat{\alpha}}$.

### Illustration

We illustrate our framework on the well-known Zachary's Karate Club graph [5]. We generated over 600 embeddings in dimension 2 to 128, using several different algorithms. In Figure 1, we display the best and worst embeddings according to our framework. Projection in 2 dimensions is obtained with UMAP[1]. Different colors and shapes for the vertices correspond to the two known communities. We clearly see that the best embedding does a much better job at keeping the vertices within each community close. Results over several other real and artificial graphs can be found in [2], all with conclusions similar to Figure 1.

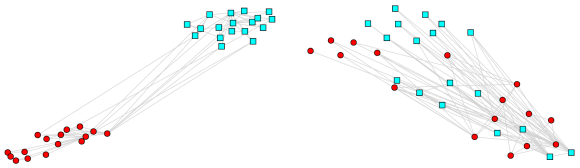

Figure 1: The Karate Club Graph. We show the best (left) and worst embeddings according to our framework given over 600 different choices.

### Scaling to Large Graphs

In order to scale up to very large graphs, we need to address the fact that in Step 3, we require the computation of $\Theta(n^2)$ distances in the embedded space, which can be prohibitive. We do so by grouping vertices from the same part of $\mathbf{C}$ that are close to each other in the embedded space. Once such refinement of partition $\mathbf{C}$ is generated, we simply replace each group by the corresponding auxiliary vertex that is placed in the center of mass of the group it is associated with. We call such auxiliary vertices landmarks, which we can then use to approximate the vectors in (2). The only small difference we need to account for is that this process will introduce loop edges for the landmarks. For such edges, we use the average distance between nodes in the group and the landmark in embedded space. Since we aim for a fast algorithm, the total number of landmarks should be close to $O(n^{1/2})$.

In Figure 2, we illustrate this process by looking at the ABCD benchmark graph [3][2] with 100,000 vertices. On the x-axis, we vary the number of landmarks while on the y-axis, we compare the divergence for some given embedding, as well as the running time. We see that very good results are obtained with as little as hundreds of landmarks, which run in a few seconds, compared to several minutes with 10,000 landmarks.

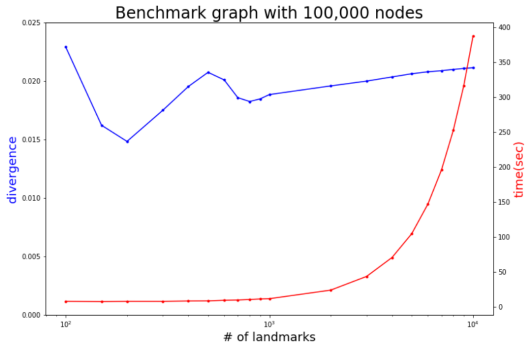


Figure 2: An illustration of the scalable landmark-based version of our framework on a benchmark graph with 100,000 vertices.

### References

[1] F. Chung and L. Lu. *Complex Graphs and Networks*. American Mathematical Society, 2006.

[2] B. Kamiński, P. Prałat, and F. Théberge. An unsupervised framework for comparing graph embeddings. *Journal of Complex Networks*, 2020.

[3] B. Kamiński, P. Prałat, and F. Théberge. Artificial benchmark for community detection (abcd) — fast random graph model with community structure. *arXiv:2002.00843*, preprint.

[4] V. Poulin and F. Théberge. Ensemble clustering for graphs: Comparison and applications. *Applied Network Science vol. 4, no. 51*, 2019.

[5] W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research 33*, 1977.

[1] `github.com/lmcinnes/umap`

[2] `github.com/bkamins/ABCDGraphGenerator.jl`