
Article:
Lozano, Miguel Angel, Escolano, Francisco, Curado, Manuel et al. (1 more author) (2021) Network Embedding from the Line Graph:Random Walkers and Boosted Classification. Pattern Recognition Letters. pp. 36-42. ISSN 0167-8655

https://doi.org/10.1016/j.patrec.2020.12.018

Network Embedding from the Line Graph: Random Walkers and Boosted Classification

Miguel Angel Lozano[a,*], Francisco Escolano[a], Manuel Curado[b], Edwin R. Hancock[c]

[a]*Department of Computer Science and AI, University of Alicante*
[b]*Polytechnic School, Catholic University of Murcia*
[c]*Department of Computer Science, University of York*

Abstract

In this paper, we propose to embed edges instead of nodes using state-of-the-art neural/factorization methods (DeepWalk, node2vec, NetMF). These methods produce latent representations based on co-ocurrence statistics by simulating fixed-length random walks and then taking bags-of-vectors as the input to the Skip Gram Learning with Negative Sampling (SGNS). We commence by expressing commute times embedding as matrix factorization, and thus relating this embedding to those of DeepWalk and node2vec. Recent results showing formal links between all these methods via the spectrum of graph Laplacian, are then extended to understand the results obtained by SGNS when we embed edges instead of nodes. Since embedding edges is equivalent to embedding nodes in the line graph, we proceed to combine both existing formal characterizations of the line graphs and empirical evidence in order to explain why this embedding dramatically outperforms its nodal counterpart in multi-label classification tasks.

Keywords: Network embedding, SGNS, Line graph, Spectral Theory.

1. Introduction

The recent success of neural graph embeddings such as LINE (Tang et al., 2015), DeepWalk (Perozzi et al., 2014a) and node2vec (Grover and Leskovec, 2016a) has opened a new path for analyzing networks. Despite these embeddings outperform spectral ones in tasks such as link prediction and multi-label node classification, Spectral Graph Theory (Chung, 1997) is still key tool for understanding and characterizing neural embeddings (Qiu et al., 2018a).

In this paper, we contribute with empirical evidence showing that neural embeddings (Section 2) can boost their performance in multi-label classification by embedding edges instead of nodes. We conjecture that this fact is due to the spectral properties of *line graphs*, whose nodes are the edges of the original graphs (Section 3). However, since general line graphs have not been fully characterized yet, we can only correlate our empirical findings (Section 4) with some of the well known properties of line graphs and the spectral characterization of neural embeddings. More precisely, we conjecture that the spectrum of the normalized Laplacian of a graph majorizes that of the normalized Laplacian of its line graph. This conjecture is solved for regular graphs and it is coherent with our empirical observations. In addition, this fact explain the boosted performance on classification in conjunction with the lack of scale-freedom and better clustering coefficients of line graphs.

*Corresponding author: Tel.: +34-965-903900; fax: +34-965-903902;

Email address: malozano@ua.es (Miguel Angel Lozano)

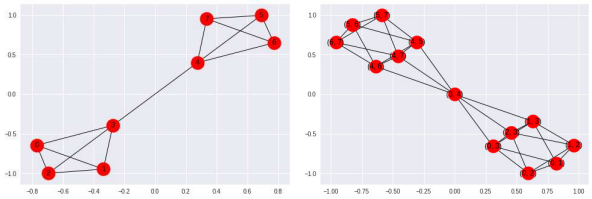

Figure 1: Barbell graph linking two cliques (left) and its line graph (right)

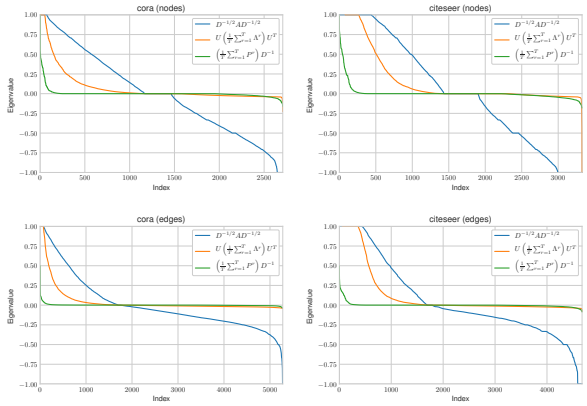

Figure 2: Eigenvalues of the original (top) and line graph (bottom), for Cora (left) and CiteSeer (right) databases

2. Classic vs Neural Embeddings

2.1. Classic Embeddings

Let $G = (V, E, \mathbf{A})$ be a graph/network with $n = |V|$ nodes, $m = |E|$ edges, where $E \subseteq V \times V$, and adjacency matrix $\mathbf{A}$. Then, *node embedding* consists of finding a mapping $f : V \to \mathbb{R}^d$ (with $d \ll n$) so that the resulting $d-$dimensional vectors capture the structural properties of each vertex. As a result, we have $\|f(i)-f(j)\|^2 \to 0$ if nodes i and j are structurally similar within the graph G. Traditionally, nodal structural similarity was associated with the reachability of node j from node i (and vice versa) through random walks (Lovász, 1996). This characterization leads us to define both *hitting times* H_{ij} (expected steps taken by a random walk to reach j from i) and *commute times* $CT_{ij} = H_{ij} + H_{ji}$ (which also includes the expected steps needed to return to i from j). Since random walks are encoded by transition matrices of the form $\mathbf{P} = \mathbf{D}^{-1}\mathbf{A}$, where $\mathbf{D} = diag(d_1, \ldots, d_n)$ is the diagonal matrix with the degrees of the nodes, the spectral analysis of $\mathbf{P}$ is a natural way of understanding both hitting and commute times. More precisely, let $\mu_1 = 1 \geq \mu_2 \geq \ldots \geq \mu_n \geq -1$ be the spectrum of the transition matrix. It is well known that hitting times and commute times are highly conditioned by the *spectral gap* $\mu = 1 - \max\{\mu_2, |\mu_n|\}$. When several communities are encoded by a connected graph G, then H_{ij} and CT_{ij} are only meaningful when $\mu \to 0$ (small bottlenecks between communities); otherwise, these quantities rely on the local densities (degrees) of the nodes i and j, and one cannot discriminate whether two nodes belong to the same community or not (von Luxburg et al., 2014). Consequently, the applicability of node embeddings based on commute times to clustering is quite limited (see representative examples of image segmentation and tracking in Qiu and Hancock (2007)). In this regard, recent research is focused on simultaneously minimizing the spectral gap and shrinking (whenever possible) inter-community commute distances via graph densification (Curado et al., 2019) before embedding the nodes.

Therefore, once G is processed (or rewired) commute times embedding leads to learn two matrices $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{n\times d}$, whose rows are denoted by $\mathbf{x}_i$ and $\mathbf{y}_j$ respectively and $\mathbf{x}_i$ is the embedding of the node i. Following Qiu and Hancock (2007), the commute times embedding matrix $\mathbf{X}$ results from factorizing

$$vol(G)\mathcal{G} = \mathbf{X}\mathbf{Y}^T \,, \tag{1}$$

where $vol(G) = \sum_{i=1}^{n} d_i$ is the volume of the graph and $\mathcal{G}$ is its Green's function, i.e. the pseudo-inverse of the normalized graph Laplacian $\mathcal{L} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$, whose spectrum is $\lambda_1 = 1-\mu_1 = 0, \lambda_2 = 1-\mu_2, \ldots, \lambda_n = 1-\mu_n \leq 2$, i.e. if μ_i is an eigenvalue of $\mathbf{P}$ then $\lambda_i = 1 - \mu_i$ is an eigenvalue of $\mathcal{L}$.

2.2. Neural Embeddigs

Neural embeddings such as LINE (Tang et al., 2015), DeepWalk (Perozzi et al., 2014a) and node2vec (Grover and Leskovec, 2016a), exploit random walks in a different way. Namely, they simulate a fixed number N of random walks with fixed length L emanating from the nodes of G and then capture co-ocurrence statistics of pairs of nodes. Each path consists of a sequence of visited nodes $w_1, w_2, \ldots, w_i, \ldots, w_L$. The first node w_1

of each path, assimilated to a word in a textual corpus (skip-gram model), is sampled from a prior distribution $P(w)$. The next nodes in the random walk are obtained according to the transition matrix $\mathbf{P}$. The context of w_i is given by the nodes/words surrounding it in a $T-$sized window $w_{i-T}, \ldots, w_{i-1}, w_{i+1}, \ldots, w_{i+T}$. Then, the node-context pairs (w, c) are given by (w_{i-r}, w_i) and (w_i, w_{i+r}) for $r = 1, \ldots, T$. All these pairs are added to the multiset $\mathcal{D}$ used for learning with *negative sampling*. Negative sampling implies not only to consider likely node-context pairs (w, c) but also b unlikely ones (w, c'): the negative samples c', are nodes that can be drawn from the steady-state probability distribution of the random walk, i.e. $P_N(i) = d_i/vol(G)$. This process is called Skip Gram Learning with Negative Sampling (SGNS) and leads to the following factorization (Levy and Goldberg, 2014):

$$\mathbf{M} = \mathbf{X}\mathbf{Y}^T, \text{ with } \mathbf{M}_{ij} = \log\left(\frac{\#(w_i, c_j)|\mathcal{D}|}{\#(w_i)\#(c_j)}\right) - \log b \ , \quad (2)$$

where: $\#(w_i, c_i)$ is the number of times the corresponding node-context pair is observed,$\#(w_i)$ is the number of times the node i is observed and similarly for node $\#(c_j)$; finally log(.) is the element-wise logarithm and b is the number of negative samples.

2.3. LINE and DeepWalk vs node2vec Factorizations

These strategies differ in the way they sample (and thus vectorize) the graph for SGNS. LINE and DeepWalk rely on first-order random walks whereas node2vec is driven by second-order random walks.

2.3.1. LINE & DeepWalk.

LINE's factorization is a direct result from the cost function associated with SGNS.

This method aims to learn two representation matrices $\mathbf{X}$ and $\mathbf{Y}$, whose rows are denoted by $\mathbf{x}_i$ and $\mathbf{y}_j$, respectively. In particular, the latent representations of both the word/node $\mathbf{x}_i$ and the context $\mathbf{y}_j$ are assumed to be correlated with the existence of an edge between nodes i and j, i.e. $\mathbf{A}_{ij} \log g(\mathbf{x}_i^T \mathbf{y}_j)$ is maximized, where $g(.)$ is the sigmoid function. Following Qiu et al. (2018a), this leads to

$$\begin{aligned} \mathbf{x}_i^T \mathbf{y}_j = \log\left(\frac{vol(G)\mathbf{A}_{ij}}{d_i d_j}\right) - \log b \ \Rightarrow \\ \Rightarrow \log\left(vol(G)\mathbf{D}^{-1}\mathbf{A}\mathbf{D}^{-1}\right) - \log b = \mathbf{X}\mathbf{Y}^T \ . \end{aligned} \quad (3)$$

DeepWalk, on the other hand, leads to a more complex factorization. Assuming that the first node of each random walk is drawn from the steady state distribution, we have that, when $L \to \infty$,

$$\frac{\#(w_i, c_j)|\mathcal{D}|}{\#(w_i)\#(c_j)} \xrightarrow{p} \frac{vol(G)}{2T}\left(\frac{1}{d_j}\sum_{r=1}^{T}\mathbf{P}_{ij}^r + \frac{1}{d_i}\sum_{r=1}^{T}\mathbf{P}_{ji}^r\right) \quad (4)$$

where $\xrightarrow{p}$ denotes *convergence in probability*. This yields

$$\log\left(\frac{vol(G)}{T}\left(\sum_{r=1}^{T}\mathbf{P}^r\right)\mathbf{D}^{-1}\right) - \log b = \mathbf{X}\mathbf{Y}^T \ , \quad (5)$$

which is equivalent to LINE for $T = 1$.

2.3.2. node2vec.

The underlying idea of this embedding is to add more flexibility to the random walk. This is done by defining two parameters p and q that control, respectively, the likelihood of inmediately revisit a node in the walk and making the walk very local. To that end, node2vec needs to evaluate the probability of the next nodes given the preceeding one in the walk, i.e. we have a 2nd-order random walk. This walk is characterized by the hypermatrix $\underline{\mathbf{P}}$, where $\underline{\mathbf{P}}_{i(jk)}$ denotes the probability of reaching i from j given that the node preceeding j is k. Thus, the 2nd order random walk can be reduced to a 1st order one on the edges of the graph (Benson et al., 2017) as it is done in the implementation of node2vec. The stationary distribution $\mathbf{X}_{ik}$ for this type of random walks satisfies $\sum_k \underline{\mathbf{P}}_{i(jk)}\mathbf{X}_{ik} = \mathbf{X}_{ij}$. Qiu et al. (Qiu et al., 2018a) have found that

$$\frac{\#(w_i, c_j)|\mathcal{D}|}{\#(w_i)\#(c_j)} \xrightarrow{p} \frac{\frac{1}{2T}\sum_{r=1}^{T}\left(\sum_u \mathbf{X}_{w_i u}\underline{\mathbf{P}}^r_{c_j(w_i u)} + \sum_u \mathbf{X}_{c_j u}\underline{\mathbf{P}}^r_{w_i(c_j u)}\right)}{\left(\sum_u \mathbf{X}_{w_i u}\right)\left(\sum_u \mathbf{X}_{c_j u}\right)} \quad (6)$$

and, despite the matricial expression for the factorization is more elusive, the final factorization differs significantly from those of DeepWalk and LINE.

3. Node vs Edges Embedding

3.1. The Line Graph

In this paper, we are mainly concerned with the impact of embedding the edges of G instead of its nodes. This

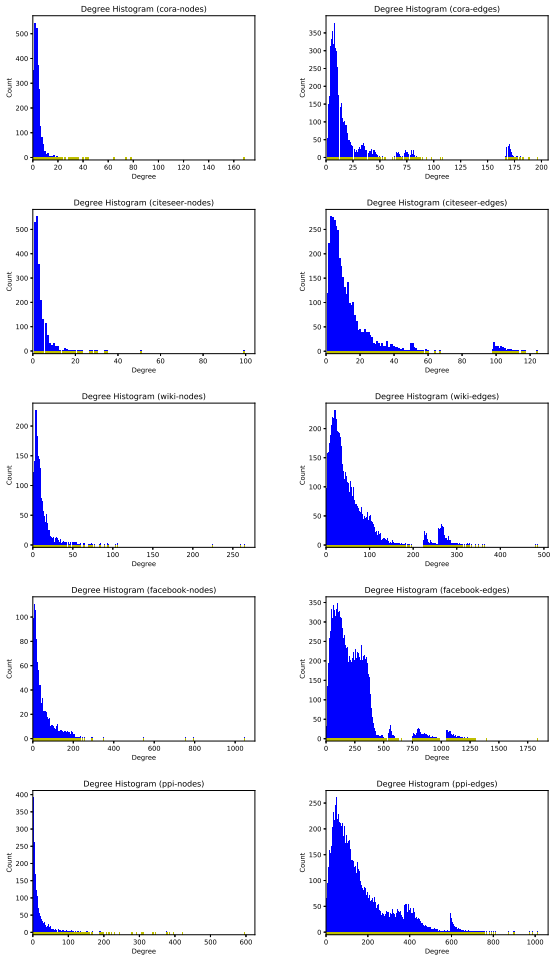

Figure 3: Degree histogram of the original (left) and line (right) graph.

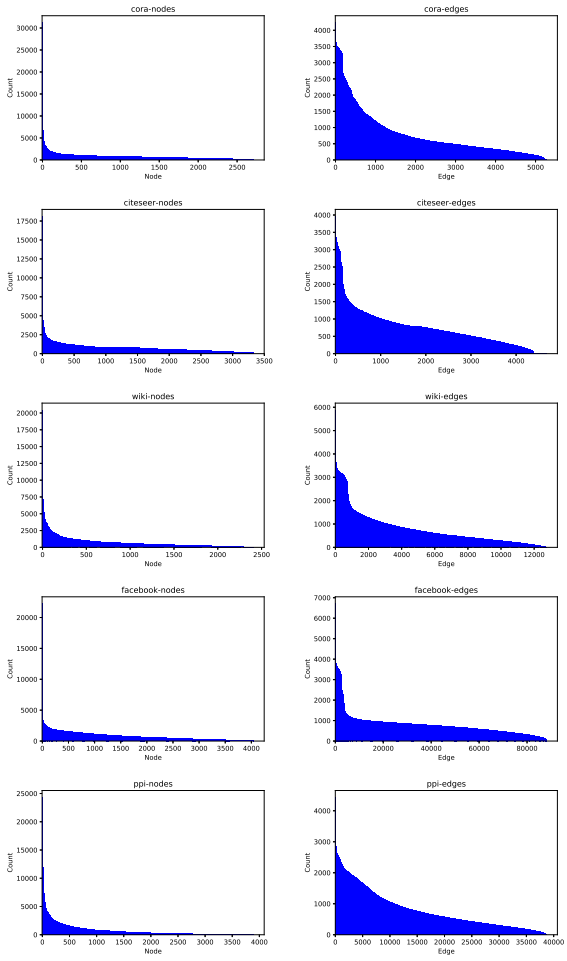

Figure 4: Walk histogram of the original (left) and line (right) graph.

means that a word w_i in the previous expressions is not yet associated with a node of G but with a node of its *line graph* ℓG. The nodes of ℓG are the edges of G and there is an edge in the line graph if two edges in G share a node. More formally, given the $n \times m$ incidence matrix $\mathbf{B}$, where n and m are the number of nodes and edges, respectively, of G, and $\mathbf{B}_{i\alpha}$ is 1 if the link α is related to node i and 0 otherwise, we have that the elements of the $m \times m$ adjacency matrix C of ℓG are $C_{\alpha\beta} = \sum_{i=1}^{n} \mathbf{B}_{i\alpha}\mathbf{B}_{i\beta}(1 - \delta_{\alpha\beta})$. The term $(1 - \delta_{\alpha\beta})$ is introduced to avoid self-loops in the line graph (when $\alpha = \beta$ it is set to 0).

3.2. *Spectral Analysis*

Some interesting properties of line graphs ℓG vs G:

- *Boosted edge density*. A single node i in G leads to a clique of $d_i(d_i - 1)/2$ edges in ℓG (see Fig. 1). Despite this gives a high prominence to notable nodes of G, it flexibilizes community detection (Evans and Lambiotte, 2009). In addition, the steady state distribution of a random walk in ℓG is $P_N(\alpha_{(i,j)}) = d_\alpha / vol(\ell G)$ where $d_\alpha = d_i + d_j - 2$ and $vol(\ell G) = \sum_{\alpha,\beta} C_{\alpha\beta} = \sum_{i=1}^{n} d_i(d_i - 1)$.

- *Redundant spectrum for $m > n$.* Let $\lambda_1(\ell G) \geq \lambda_2(\ell G) \geq \ldots \geq \lambda_m(\ell G)$ be the spectrum of $\mathbf{C}$. Then, for $m > n$, $\lambda_{n+1}(\ell G) = \ldots = \lambda_m(\ell G) = -2$. As a result, $\lambda_i(\mathbf{L}(\ell G)) \geq 4$, for the largest $m - n$ eigenvalues of $\mathbf{L}(\ell G)$, the unnormalized Laplacian matrix of ℓG (Yan, 2002). This may reduce significantly the medium-large eigenvalues of $\mathcal{L}(\ell G)$ with respect to those of $\mathcal{L}(G)$, that is increase those of $\mathbf{I} - \mathcal{L}(\ell G)$ wrt those of $\mathbf{I} - \mathcal{L}(G)$ (see Fig. 2).

- *Majorization of the spectrum of the Laplacian of ℓG.* For $k-$regular G, we have that ℓG is $2(k-1)-$regular (see Ramane et al. (2005)). This is consistent with Biggs (1974) (Thm.8 in Chapter 3) where $\lambda_i(\ell G) = \lambda_i(G) + 2 - k$ for $i \leq n$. This implies that $\lambda_i(\mathbf{L}(\ell G)) = 2(k-1) - \lambda_i(G) \geq 0$ for $i \leq n$. As a result, we have $\lambda_2(\mathbf{L}(\ell G)) = 2(k-1) - 2 + \lambda_2(\mathbf{L}(G))$, i.e. $\lambda_2(\mathbf{L}(\ell G)) > \lambda_2(\mathbf{L}(G))$ for $k > 2$. Since the normalized Laplacian of a $k-$regular graph satisfies $\mathcal{L}(G) = \frac{1}{k}\mathbf{L}(G)$, and similarly we have $\mathcal{L}(\ell G) = \frac{1}{2(k-1)}\mathbf{L}(\ell G)$, it is straightforward to obtain $\lambda_2(\mathcal{L}(\ell G)) < \lambda_2(\mathcal{L}(G))$ for $k > 2$. Therefore, the spectral gap of G majorizes that of ℓG in regular graphs and this leads to larger mixing times in ℓG wrt G.

  We conjecture that the above majorization is also valid for non-regular graphs, because it is consistent with our observations. In Fig. 2 we show that the largest part of the spectrum of $\mathbf{I} - \mathcal{L}$ in the line graph majorizes that of G. Since the spectrum driving DeepWalk is (approximately) of the form $\frac{1}{T}\sum_{r=1}^{T}\mu_i^r$, this leads (in general) to small spectral gaps for the line graphs, and thus large mixing times (green lines show the real spectra driving random walks in DeepWalk; in all cases, $T = 10$). Large mixing times tend to reduce the redundancy of the embeddings, since the random walks typically sample far from the stationary distribution (e.g. they surf far from notable nodes) (see Mohaisen et al. (2010)). We will see the impact of this fact in the Experimental Section.

  Taking the regular case as a departure point, solving the majorization conjecture (specially its impact in mixing times) in forthcoming work requires a careful look of other families of graphs. For instance, Benjamini et al. (2014) proved that the mixing times of random graphs is $\Theta(\log^2 n)$, where n is the number of nodes, and this leads to $\Theta(\log^2 m)$ mixing times in line graphs. In addition, solving this question also requires to analyze the not scale freedom of line graphs obtained from scale-free ones. We will explore this point in the following section.

4. Experiments and Discussion

4.1. Datasets (Networks)

In our experiments, we have used the following datasets:

- **CiteSeer for Document Classification** (Sen et al., 2008). Citation network containing 3312 scientific publications with 4676 links between them. Each publication is classified in one of 6 categories.

- **Cora** (Sen et al., 2008). Citation network containing 2708 scientific publications with 5278 links between them. Each publication is classified in one of 7 categories.

- **Wiki** [1]. Contains a network of 2405 web pages with 17981 links between them. Each page is classified in one of 19 categories.

- **Facebook** social circles (McAuley and Leskovec, 2012). Consists of 4039 nodes (users) and 88234 links between them, organized in 10 categories (groups of users).

- **Wikipedia Part-of-Speech (POS)** (Mahoney, 2011). Co-ocurrence of words appearing in the first million of the bytes of the dumping of Wikipedia. The categories correspond to the labels of Part-of-Speech (POS) inferred by the Stanford POS-Tagger. Contains 4777 nodes, and 92517 undirected links. Each node may have several labels. We have 40 labels (categories).

- **Protein-Protein Interactions (PPI)** [2] (Breitkreutz et al., 2008). We use a subgraph of the PPIs associated with the Homo Sapiens. The network has 3890

[1] https://github.com/thunlp/MMDW/

[2] https://downloads.thebiogrid.org/BioGRID

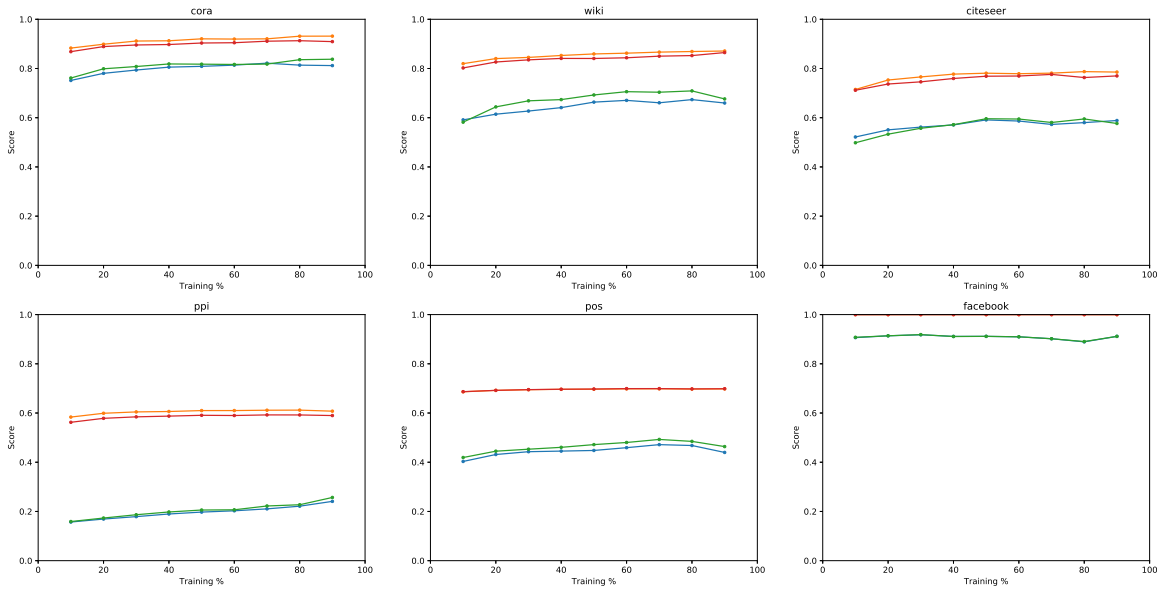

Figure 5: Evolution of the performance as a function of the fraction of known labels in the training set (Micro-F1 score)

Table 1: Properties of the datasets

	Nodes	Edges	Line graph edges	Gap	Con. comps.	Labels	Multi-label
wiki	2405	12761	355644	0.000000	45	19	no
cora	2708	5278	52301	0.000000	78	7	no
citeseer	3327	4676	27174	0.000000	438	6	no
ppi	3890	38739	3018220	0.000000	35	50	yes
pos	4777	92517	49568882	0.576132	1	40	yes
facebook	4039	88234	9314849	0.000837	1	10	yes

nodes and 76584 links. Each node may have several labels corresponding to the 50 possible categories.

Facebook, PPI and POS have been retrieved from SNAP [3] (Leskovec and Krevl, 2014). CiteSeer and Cora have been retrieved from LINQS [4].

See Table 1 for details of these datasets. All the networks are considered as undirected graphs. Nodes in ℓG corresponding to edges in G that connect nodes of different classes, will be assigned to both classes. These nodes of ℓG will be considered *inter-class nodes* (border nodes). On the other hand, edges connecting nodes of the same class will be considered *intra-class nodes* in ℓG. Thus, originally single-labelled networks are transformed into multi-label networks when their line graph is computed. In the case of originally multi-label graphs, it is generalized as follows: the set of labels of a node $\alpha_{(i,j)}$ in ℓG is obtained as the union of the sets of labels of i and j in G. In the next table we show the amount of inter-class and intra-class nodes in originally single-label datasets:

[3] `https://snap.stanford.edu/node2vec/`
[4] `https://linqs.soe.ucsc.edu/data`

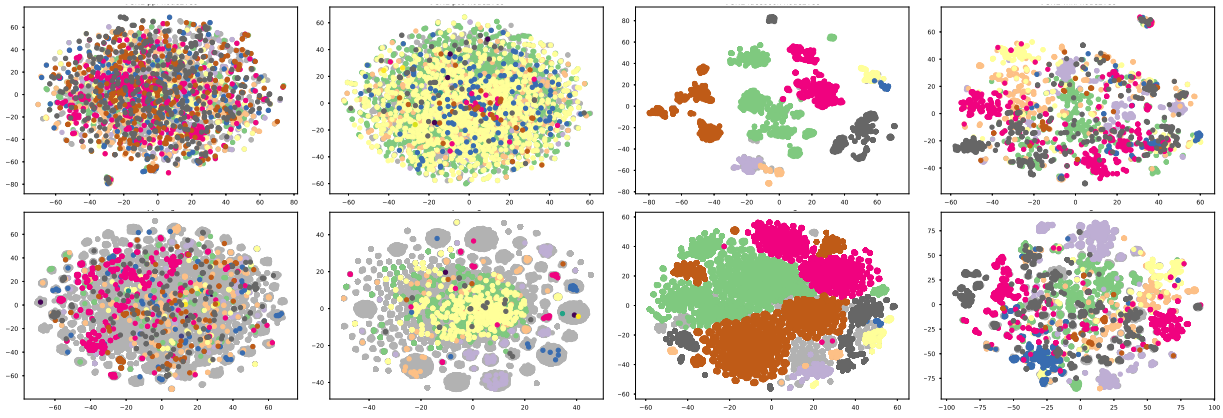

Figure 6: t-SNE embeddings. Original graph (top) and line graph (bottom), for PPI (first column), POS (second column), Facebook (third column) and Wiki (fourth column) databases.

Table 2: Micro-F1

	node2vec		DeepWalk		NetMF		CTE	LLE
	Nodes	Edges	Nodes	Edges	Nodes	Edges		
citeseer	0.591071	0.768626	0.595833	0.780930	0.595833	**0.782386**	-	-
cora	0.808715	0.903557	0.817578	**0.920538**	0.824963	0.900414	-	-
wiki	0.663342	0.840709	0.692436	**0.859129**	0.684954	0.852148	-	-
pos	0.447845	**0.697572**	0.471620	0.696640	0.449926	-	0.375037	0.393165
ppi	0.197435	0.590731	0.205786	0.609937	0.236206	**0.610787**	-	-
facebook	0.911516	**0.999900**	0.911516	**0.999900**	0.910182	-	0.239217	0.231214

	Inter-class nodes	Intra-class nodes
wiki	4526 (35%)	8235 (65%)
cora	1003 (19%)	4275 (81%)
citeseer	1190 (25%)	3486 (75%)

4.2. *Random walk analysis*

The above spectral analysis leads to explain the behaviour of random walkers surfing the line graphs instead of the original graphs. The larger density of ℓG wrt to that of G, and the spectral majorization (leading to a larger mixing time) suggest that the walkers explore *more efficiently* ℓG than G (i.e. they explore more in-depth the line graphs than the original graphs). This is partially due to the fact that line graphs usually loose the power-law property of the original graphs when they are scale-free. Wang et al. (2016) showed that if one assume that the degree of nodes are independent, then line graphs of scale-free graphs follow a power law of slope $\alpha_{\ell G} = 2$ smaller than that of the original scale-free graphs (with $\alpha_G = 3$). However, as the preferential attachment property of scale-free graphs rules out degree independence, scale-freedom is not preserved in line graphs. In Fig. 3 there are represented the histograms of node degrees for each dataset, both considering the original graph (left) and its line graph (right). It can be observed that the line graphs tend to spread the histogram, specially in the case of denser graphs (wiki, ppi, facebook).

A second fact that explains the efficiency of random walkers on line graphs is *assortativity* (preferential connection to nodes with the same properties). Wang et al. (2016) derivated a formula for the assortativity of a line graph ℓG as a function of that of its original graph G. There is clearly a non-linear dependence between both assortativities, and more interestingly the assortativities of line graphs are larger (and usually positive) than those of

Table 3: Macro-F1

	node2vec		DeepWalk		NetMF		CTE	LLE
	Nodes	Edges	Nodes	Edges	Nodes	Edges		
citeseer	0.544490	0.730930	0.545774	0.745995	0.555131	**0.746387**	-	-
cora	0.798782	0.898863	0.804928	**0.917838**	0.817348	0.895754	-	-
wiki	0.528603	0.764205	0.597948	**0.787771**	0.536270	0.759482	-	-
pos	0.084183	0.773035	0.094148	**0.774001**	0.081939	-	0.041637	0.033617
ppi	0.168237	0.566912	0.178405	0.587934	0.206204	**0.594749**	-	-
facebook	0.821899	0.999377	0.822243	**0.999407**	0.818303	-	0.108742	0.045155

the original graphs with negative assortativities. Since the assortativity is related to the clustering coefficient, it turns out that line graphs are (usually) better clustered than the original graphs.

The confluence of both the non-power law behaviour and higher assortativity of line graphs makes random walks to surf line graphs in a different (more efficient way). There is a smaller number of *key nodes* (with low probability of scape from them) and the exploration becomes more entropic. As a result, the embedding become less redundant (more informative). For instance, in Fig. 4 we show the walking histogram of these graphs, i.e. how many times each node if traversed by a random walk. In this case it is much more clear how line graph helps to distribute the walks among the nodes.

4.3. Classification experiments

The classification experiments aim to compare the effectiveness of node (original graph) and edge (line graph) embeddings to discriminate between different communities in the network.

These experiments are performed by training a logistic regression classifier with the embedding vectors corresponding to 50% of the nodes, and tested with the remaining vectors, using the OpenNE framework [5]. We compare the classification results obtained with different embedding methods: node2vec (Grover and Leskovec, 2016b), DeepWalk (Perozzi et al., 2014b), NetMF [6] (Qiu et al., 2018b), LLE (*Locally Linear Embedding*, Roweis and Saul (2000)) and CTE (*Commute Time Embedding*, Qiu and Hancock (2007)). In case of node2vec, DeepWalk and NetMF, we compare the embedding obtained both from the original graph (node embedding) and from its line graph (edge embedding). The obtained results are presented in Tables 2 (Micro-F1 score) and 3 (Macro-F1 score). Some cells in the results tables have been left in blank due to computational limitations.

In this experiment, we observe that in all cases the edge space is more convenient in order to classify in different communities. In addition, when the number of edges is much higher than the number of nodes, the obtained gain is also higher (facebook, pos, ppi, and wiki). We can also observe that a high ratio of intra-class connectivity (cora) also favors to obtain better classification results. In these cases, density of intra-class connectivity is boosted by line graphs.

The default values for p and q in node2vec are $p = q = 1$. After optimizing p and q in the range $\{0.25, 0.5, 1, 2, 4\}$ the maximum improvement of node2vec wrt DeepWalk in the classification score is 0.014 (micro and macro). Regarding spectral embeddings, CTE and LLE have been only tested in networks with a single connected component (pos and facebook). In particular, commute times (CTE) have a poor performance in multi-label classification because their factorization relies on the Green's function and this means that only the inverse of each eigenvalue is considered. However, DeepWalk is controlled by a polynomial associated with each eigenvalue.

In Fig. 5, we show the performance of classification (Micro-F1 score) with different percentages of training data (ranging from 10% to 90%). The line graph versions of node2vec and DeepWalk clearly outperform their nodal counterparts. The similarity in terms of performance of

[5] `https://github.com/thunlp/OpenNE`

[6] NetMF implementation has been retrieved from `https://github.com/xptree/NetMF/`

node2vec and DeepWalk is due to the fact that the 2nd order random walk of node2vec is not applied at the level of edges (it is unfeasible for large networks). Finally, in Fig. 6 we show the t-SNE embeddings for POS, PPI, Facebook and Wiki datasets. Edge embeddings clearly produce denser communities.

5. Conclusions

In this paper, we have contributed with empirical evidence showing that embedding edges clearly outperforms node-based embeddings in neural SGNS strategies. We conjecture that this is due to the larger mixing times of random walks in line graphs. We solve the conjecture for regular graphs and contribute with characterizing the behaviour of random walks surfing on line graphs. In particular, we explain the fact that line graphs generate more informative (more entropic, less redundant) embeddings by analyzing some key combinatorial properties: density, not scale free, positive assortativity and large clustering coefficient.

Future work includes a detailed check of this conjecture as well as more efficient (in time and space) strategies for designing walkers on the line graphs. In addition, we are studying the directed case (digraph) where the non-zero spectrum of the transition matrix of the original graph is preserved in the line digraph. In this case, given that the edges have a source and a destination node, an edge can be mapped directly to a single node (e.g., its destination node in citation networks).

Acknowledgments

M.A. Lozano, M. Curado and F. Escolano are funded by the project RTI2018-096223-B-I00 of the Spanish Government.

References

Benjamini, I., Kozma, G., Wormald, N.C., 2014. The mixing time of the giant component of a random graph. Random Struct. Algorithms 45, 383–407. URL: https://doi.org/10.1002/rsa.20539, doi:10.1002/rsa.20539.

Benson, A.R., Gleich, D.F., Lim, L., 2017. The spacey random walk: A stochastic process for higher-order data. SIAM Review 59, 321–345. doi:10.1137/16M1074023.

Biggs, N., 1974. Algebraic Graph Theory. Cambridge Mathematical Library. 2 ed., Cambridge University Press. doi:10.1017/CBO9780511608704.

Breitkreutz, B., Stark, C., Reguly, T., Boucher, L., Breitkreutz, A., Livstone, M.S., Oughtred, R., Lackner, D.H., Bähler, J., Wood, V., Dolinski, K., Tyers, M., 2008. The biogrid interaction database: 2008 update. Nucleic Acids Research 36, 637–640. doi:10.1093/nar/gkm1001.

Chung, F.R.K., 1997. Spectral Graph Theory. Conference Board of the Mathematical Sciences (CBMS), number 92, American Mathematical Society.

Curado, M., Escolano, F., Lozano, M.A., Hancock, E.R., 2019. Dirichlet densifiers for improved commute times estimation. Pattern Recognit. 91, 56–68. doi:10.1016/j.patcog.2019.02.012.

Evans, T.S., Lambiotte, R., 2009. Line graphs, link partitions, and overlapping communities. Phys. Rev. E 80, 016105. doi:10.1103/PhysRevE.80.016105.

Grover, A., Leskovec, J., 2016a. node2vec: Scalable feature learning for networks, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016, pp. 855–864. doi:10.1145/2939672.2939754.

Grover, A., Leskovec, J., 2016b. node2vec: Scalable feature learning for networks, in: Proceedings of KDD, pp. 855–864.

Leskovec, J., Krevl, A., 2014. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data.

Levy, O., Goldberg, Y., 2014. Neural word embedding as implicit matrix factorization, in: Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada, pp. 2177–2185.

Lovász, L., 1996. Random walks on graphs: A survey, in: Miklós, D., Sós, V.T., Szőnyi, T. (Eds.), Combinatorics, Paul Erdős is Eighty. János Bolyai Mathematical Society, Budapest. volume 2, pp. 353–398.

von Luxburg, U., Radl, A., Hein, M., 2014. Hitting and commute times in large random neighborhood graphs. Journal of Machine Learning Research 15, 1751–1798. URL: http://dl.acm.org/citation.cfm?id=2638591.

Mahoney, M., 2011. Large text compression benchmark. URL: http://www.mattmahoney.net/dc/textdata.

McAuley, J.J., Leskovec, J., 2012. Learning to discover social circles in ego networks, in: Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States., pp. 548–556.

Mohaisen, A., Yun, A., Kim, Y., 2010. Measuring the mixing time of social graphs, in: Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement, Association for Computing Machinery, New York, NY, USA. pp. 383–389. doi:10.1145/1879141.1879191.

Perozzi, B., Al-Rfou, R., Skiena, S., 2014a. Deepwalk: online learning of social representations, in: The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014, pp. 701–710. doi:10.1145/2623330.2623732.

Perozzi, B., Al-Rfou, R., Skiena, S., 2014b. Deepwalk: Online learning of social representations, in: Proceedings of KDD, pp. 701–710.

Qiu, H., Hancock, E.R., 2007. Clustering and embedding using commute times. IEEE Trans. Pattern Anal. Mach. Intell. 29, 1873–1890. doi:10.1109/TPAMI.2007.1103.

Qiu, H., Hancock, E.R., 2007. Clustering and embedding using commute times. IEEE Transactions on Pattern Analysis and Machine Intelligence 29, 1873–1890. doi:10.1109/TPAMI.2007.1103.

Qiu, J., Dong, Y., Ma, H., Li, J., Wang, K., Tang, J., 2018a. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec, in: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, ACM, New York, NY, USA. pp. 459–467. doi:10.1145/3159652.3159706.

Qiu, J., Dong, Y., Ma, H., Li, J., Wang, K., Tang, J., 2018b. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec, in: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, ACM. pp. 459–467.

Ramane, H., Walikar, H., Rao, S., Acharya, B., Hampiholi, P., Jog, S., Gutman, I., 2005. Spectra and energies of iterated line graphs of regular graphs. Appl. Math. Lett. 18, 679–682. doi:10.1016/j.aml.2004.04.012.

Roweis, S.T., Saul, L.K., 2000. Nonlinear dimensionality reduction by locally linear embedding. Science 290, 2323–2326. doi:10.1126/science.290.5500.2323.

Sen, P., Namata, G., Bilgic, M., Getoor, L., Gallagher, B., Eliassi-Rad, T., 2008. Collective classification in network data. AI Magazine 29, 93–106. URL: http://www.aaai.org/ojs/index.php/aimagazine/article/view/2157.

Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q., 2015. LINE: large-scale information network embedding, in: Proceedings of the 24th International Conference on World Wide Web, WWW 2015, Florence, Italy, May 18-22, 2015, pp. 1067–1077. doi:10.1145/2736277.2741093.

Wang, X., Trajanovski, S., Kooij, R., Mieghem, P., 2016. Degree distribution and assortativity in line graphs of complex networks. Physica A: Statistical Mechanics and its Applications 445, 343–356. doi:10.1016/j.physa.2015.10.109.

Yan, C., 2002. Properties of spectra of graphs and line graphs. Applied Mathematics-A Journal of Chinese Universities 17, 371–376. doi:10.1007/s11766-002-0017-7.