An integrated platform to evaluate graph embedding

Mr. Janvier RULINDA, Mr. Jean de Dieu TUGIRIMANA, Mr. Antoine NZARAMBA, Mr. Felix Oduor Aila, Mr. Gilbert Kipkirui Langat

Abstract

Graphs, such as social networks, word co-occurrence networks, and communication networks, occur naturally in various real-world applications. Analyzing these networks yields insight into the structure of society, language, and different patterns of communications. Many approaches have been proposed to perform the analysis. Recently, many researchers are interested on methods which use the representation of graph nodes in vector. In this research, we provide a comprehensive and structured analysis of an integrated platform to evaluate graph embedding techniques. We will first explain our platform and then evaluate many techniques of graph embedding on a few common datasets and compare their performance against one another. We have implemented graph embedding techniques on an integrated platform and further evaluated the performance of each technique.

Key terms: Integrated platform, Graph embedding technique, Graph embedding application evaluation metrics.

1. Introduction

Graph analysis has been attracting increaseing attention in the recent years due to the ubiquity of networks in the real world. Graphs have been used to denote information in various areas including biology (Protein- Protein interaction), social sciences (friendship networks) [1] and linguistics (word co-occurrence networks) [2]. Modeling the interaction between entities as graphs has enabled researchers to understand the various network systems in a systematic manner [3]. For example, social networks have been used for applications like friendship or content recommendation, as well as for advertisement [4]. Graph analytics tasks can be broadly abstracted into the following categories: (a) link prediction [4], (b) clustering [5] and visualization [6]. Link prediction refers to the task of predicting missing links or links that are likely to occur in the future. Clustering is a task of grouping a set of nodes in such a way that nodes in the same group are more similar to each other than to those in other groups; visualization helps to understand the structure of network. Graph embedding is the conversion of graph data into vector space in which graph structure information and graph properties are preserved, which can be used as input to machine learning models for a prediction task, clustering task and

- *Mr. Janvier RULINDA: currently pursuing master's degree program in Computer Science in Central South University, China, PH-(+86)13875939502. E-mail: janrul99@yahoo.fr*
- *Mr. Jean de Dieu Tugirimana: currently pursuing master's degree program in Computer Science in Central South University, China*
- *Mr. Antoine NZARAMBA: currently pursuing PhD program in Computer Science in Central South University, China*
- *Mr. Felix Oduor Aila: currently pursuing master's degree program in Computer Science in Central South University, China*
- *Mr. Gilbert Kipkirui Langat: currently pursuing PhD program in Computer Science in Central South University, China*

visualization task.

Our contribution can be summarized as follows:

(1) We propose a robust and expanded platform for graph embedding which integrate 11 graph embedding techniques.

(2) We evaluate the specifics with regards to performance of each model (graph embedding technique) and compare them on different datasets. We use graph embedding application evaluation metrics such as link predictions, clustering and visualization.

2. Definitions and preliminaries

We represent the set {1… n} by [n] in the rest of the paper. We start by formally defining several preliminaries which have been defined similar to Wang et al. [15].

Definition 1. (Graph**)** A graph G (V, E) is a collection of $V = \{v_1, \ldots, v_n\}$ vertices (nodes) and $E = \{e_{ij}\}_{i,j=1}^{n}$ edges. The adjacency matrix S of graph G contains non-negative weights associated with each edge: $s_{ij} \geq 0$. If v_i and v_j are not connected to each other, then $s_{ij} = 0$. For undirected weighted graphs, $s_{ij} = s_{ji}$ $\forall i, j \in [n]$.

The edge weight s_{ij} is generally treated as a measure of similarity between the nodes V_i and V_j. The higher the edge weight, the more similar the two nodes are expected to be.

Definition 2. (First-order proximity) Edge weights S_{ij} are also called **first-order proximities** between nodes V_i and V_j, since they are the first and foremost measures of similarity between two nodes.

We can similarly define higher-order proximities between nodes. For instance,

Definition 3. (Second-order proximity) The second-order proximity between a pair of nodes describes the proximity of the pair's neighborhood structure. Let $S_i = [S_{i1}, \ldots, S_{in}]$ denote the first-order proximity between V_i and other nodes. Then, second-order proximity between V_i and V_j is determined by the similarity of S_i and S_j.

Second-order proximity compares the neighborhood of two nodes and treats them as similar if they have a similar neighborhood. It is possible to define higher-order proximities using other metrics, e.g. katz index, Rooted PageRank, Common Neighbors, Adamic Adar, etc. (for detailed definitions, omitted here in the interested of space, see Ou at al.[16]).

Definition 4. (Graph embedding) Given a graph G = (V,E), a graph embedding is a mapping $f: V_i \to y_i \in R^d$ $\forall i, \in [n]$ such that d << |v| and the function *f* preserves some proximity measure defined on graph G. An embedding therefore maps each node to a low-dimensional feature vector and tries to preserve the connection strengths between vertices. For instance, an embedding preserving first-order proximity might be obtained by minimizing $\sum_{i,j} S_{ij} \| y_i - y_j \|_2^2$. Let two node pairs ($v_i$, v_j) and (v_i, v_k) be associated with connections strength such that $S_{ij} > S_{ik.}$ in this case, Vi and Vj will be mapped to points in the embedding space that will be closer each other than the mapping of v_i and $v_{k.}$

3. Integrated platform and discussion of graph embedding techniques

Many years ago, there has been a lot of research in the field of graph embedding with focus on designing new embedding technique. For us we have built an 11 model integrated platform implementation of graph embedding with a single unified interface. These graph embedding techniques are;

3.1. DeepWalk

DeepWalk [17] learns social representations of graph's vertices, by modeling a stream of short random walks. Social representations are latent features of the vertices that capture neighborhood similarity and community membership. These latent representations encode social relations in a continuous vector space with a relatively small number of dimensions.

3.2. Node2vec

Node2vec [18] is a semi-supervised algorithm for scalable feature learning in networks. Their approach returns feature representations that maximize the likelihood of preserving network neighborhoods of nodes in a D-dimensional feature space. They used a second order random walk approach to generate network neighborhoods for nodes.

3.3. GraRep

GraRep [19] learns graph representations of graph: each vertex of graph is represented with a low-dimensional vector in which meaningful semantic, relation and structural information convoyed by the graph can be accurately captured. They used the skip-gram model to capture the k-step (k =1; 2; 3 ;..) relationship between each vertex and its k-step neighbors in the graph with different values of k.

3.4. LINE

LINE [20] this model solves the problem of embedding very large information networks into low-dimensional vector spaces. This model is able to scale very large, arbitrary types of networks: undirected, directed and/or weighted. It has a carefully designed objective function that preserves both the first-order and second-order proximities.

3.5. SDNE

Wang et al. [15] proposed to use deep autoencoders to preserve the first and second order network proximities. They achieve this by jointly optimizing the two proximities. The approach uses highly non-linear functions to obtain the embedding. The model consists of two parts: unsupervised and supervised. The former consists of an autoencoder aiming at finding an embedding for a node which can reconstruct its neighborhood.

3.6. OhmNet

OhmNet[21], a hierarchy-aware unsupervised node feature learning approach for multi-layer networks. They build a multi-layer network, where each layer represents molecular interactions in a different human tissue. OhmNet then automatically learns a mapping of proteins, represented as nodes, to a neural embedding based low-dimension space of features. In the first phase OhmNet applies node2vec's algorithm to construct network neighborhoods for each node in every layer. In second phase, OhmNet uses an iterative approach in which features associated with each object in the hierarchy are iteratively updates by fixing the rest of the features. The two phases are executed sequentially.

3.7. GEMSEC

GEMSEC [22] is a graph embedding procedure which learns embedding nodes and clusters the nodes at the same time. Sequence based graph embedding procedures create similar representations for nodes which have similar sampled neighborhoods. Nodes with similar neighborhoods are expected to be in the same community. This means that clustering the representation can reveal the community structure.

3.8. M-NMF

Modularized Nonnegative Matrix Factorization (M-NMF) [23] is a graph embedding model which preserves both the microscopic structure (pairwise node similarity) and mesoscopic structure (community) for network embedding. For microscopic structure, they incorporate first- and second-order proximities of nodes to learn the representations using factorization, for mesoscopic structure, the communities are detected by a modularity constraint term. Then these two terms are connected by exploiting the consensus relationship between the representations of nodes and community structure of network with an auxiliary community representation matrix, and thus they can be jointly optimized.

3.9. Struc2vec

Struc2vec [24] is flexible framework for learning latent representations for the structural identity of nodes. It uses a hierarchy to measure node similarity at different scales and constructs a multilayer graph to encode structural similarities and generate structural context for nodes. Struc2vec assess structural similarity between nodes independently of node and edge attributes as well as their position in the network.

3.10. Diff2vec

Diff2vec[25] is graph embedding which extract a subgraph of the neighborhood of a node using diffusion like process, and call it diffusion graph. On this subgraph, it computes an Euler tour to use as a sequence. By covering all adjacencies in the graph, the Euler tour contains a more complete view of the local neighborhood than random walks. The sequences generated by diff2vec are then used to train a neural network with one hidden layer containing d neurons for d-dimensional embedding.

3.11. Prune

Proximity and Ranking preserving unsupervised Network Embedding (Prune)[26], an unsupervised Siamese neural network structure to learn node embeddings from not only community-aware proximity but also global node ranking. It is taking the embedding vectors of the adjacent nodes of a link as the training input; the shared hidden layers of this model non-linearly map node embedings to optimize a carefully designed objective function. During training, the objective function, for global node ranking and community aware proximity, propagate gradients back to update embedding vectors.

4. Applications

As graph representations, embedding can be used in a variety of tasks. These applications can be broadly classified as: visualization (4.1), clustering (4.2) and link prediction (4.3).

4.1. Visualization

Application of visualizing graphs can be dated back to 1736 when Euler used it to solve "Konigsberger Bruckenproblem"[27]. In the recent years, graph visualization has found applications in software engineering [28], electrical circuits [29], biology and sociology [1]. Battista et al. [30] and Eades et al.[31] this research a range of methods used to draw graphs and define aesthetic criteria for this purpose. Herman et al.[30] generalize this and view it from an information visualization perspective. They study and compare various traditional layouts used to draw graphs including tree, 3D and hyperbolic-based layouts.

As embedding represents a graph in a vector space, dimensionality reduction techniques like Principal Component Analysis (PCA) [32] and t-distributed stochastic neighbor embedding (t-SNE) [6] can be applied on it to visualize the graph. The authors of DeepWalk [17] illustrated the goodness of their embedding approach by visualizing the Zachary's karate Club network. The authors of Line [20] visualized the DBLP co-authorship network, and showed that Line is able to cluster together authors in the same field. The authors of SDNE [15] applied it on 20-newsgroup document similarity network to obtain clusters documents based on topics.

4.2. clustering

Graph clustering can be of two types: (a) structure based, and (b) attribute based clustering. The former can be further divided into two categories, namely community based, and structurally equivalent clustering. Structure based methods [5,14,33], aim to find dense subgraphs with high number of intra-cluster edges, and low number of inter-cluster edges. Structural equivalence clustering [34], on the contrary, is designed to identify nodes with similar roles (like bridges and outliers). Attribute based methods [13] utilize node labels, in addition to observed links, to cluster nodes.

White et al. [35] used k-means on the embedding to cluster the nodes and visualize the clusters obtained on Wordnet and NCAA datasets verifying that the clusters obtained have intuitive interpretation. Recent methods on embedding haven't explicitly evaluated their models on this task and thus it is a promising field of research in the graph embedding community.

4.3. Link prediction

Networks are constructed from the observed interactions between entities, which may be incomplete or inaccurate. The challenge often lies in identifying spurious interactions and predicting missing information. Link prediction refers to the task of predicting either missing interaction or links that may appear in the future in an evolving network. Link prediction is pervasive in biological network analysis, where verifying the existence of links between nodes requires costly experimental tests. Limiting the experiments to links ordered by presence likelihood has been shown to be very cost effective. In social networks, link prediction is used to predict probable friendship, which can be used for recommendation and lead to a more satisfactory user experience. Liben-Nowell et al. [4], Lu et al.[36] and Hasan et al.[37] survey the recent progress in this field and categorize the algorithms into (a) similarity based (local and global) [7, 8, 38], (b) maximum likelihood based [9, 10] and (c) probabilistic methods [11, 12, 39].

Embeddings capture inherent dynamics of the network either explicitly or implicitly thus enabling application to link prediction. Wang et al. [15] and Ou et al. [16] predict links from the learned node representations on publicly available collaboration and social networks. In additional, Grover et al. [18] apply it to biology networks. They show that on these data set links predicted using embedding are more accurate than traditional similarity based link prediction methods described above.

5. Experiment Setup

We employed the following 3 real datasets to evaluate graph embedding.

KARATE: Zachary’s karate network is a well-known social network of a university karate club. It has been widely studied in social network analysis. The network has 34 nodes and 78 edges.

FACEBOOK is a social network of friendship. It has 224 nodes and 6385 edges.

WIKIPEDIA is a word co-occurrence network. It has 2405 nodes, 17981 edges and 19 labels.

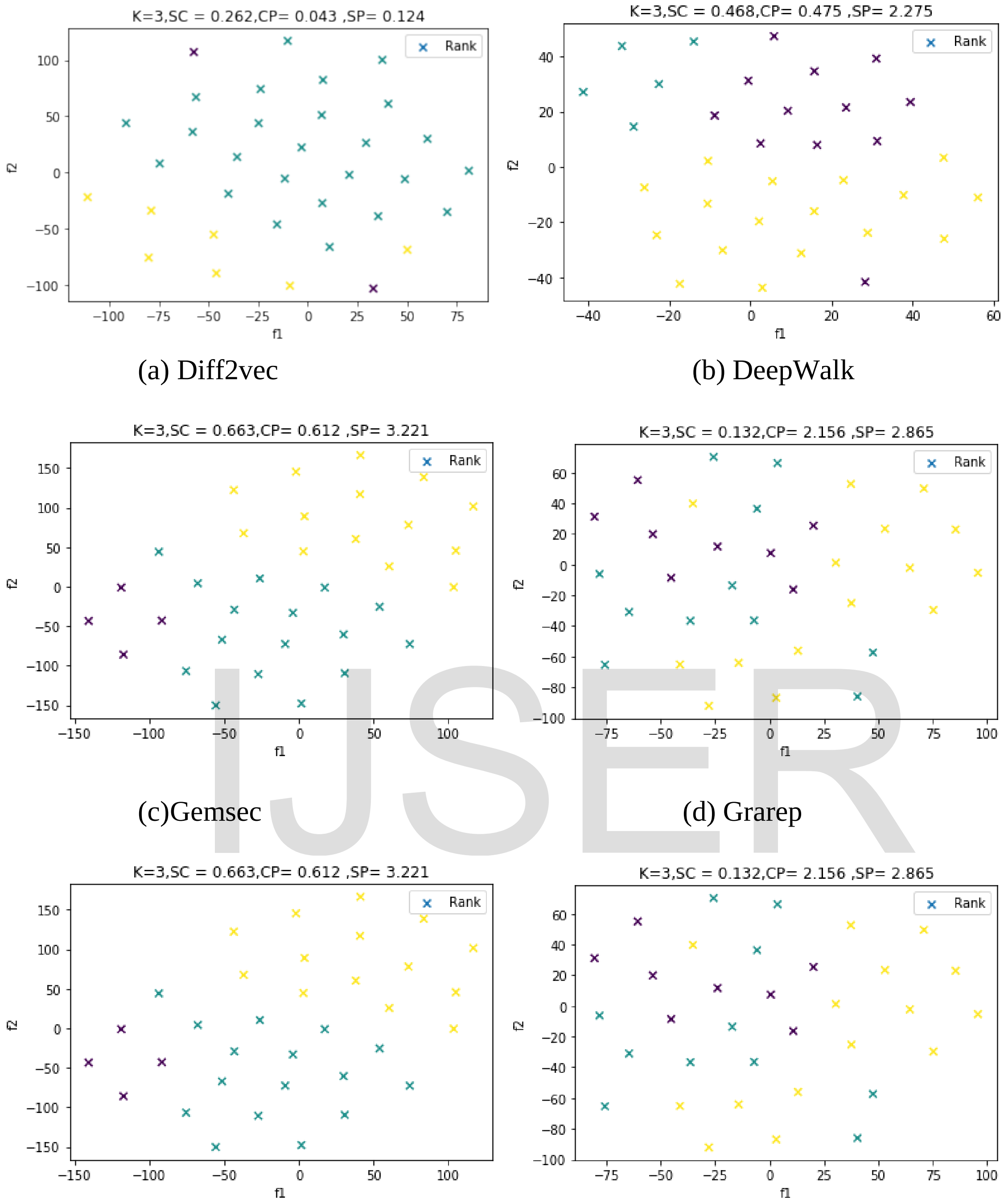
6. Experiments and analysis

In this section, we evaluate and compare embedding methods on the task of visualization, link prediction and clustering. The experiments were performed on a MacOs Sierre system with cora i5, 8Gb RAM and processor 2.8 GHZ.

6.1. Visualization

Since embedding is a low dimensional vector representation of nodes in the graph, it allows us to visualize the nodes to understand the network topology. As different embedding methods preserve different structures in the network, their ability and interpretation of node visualization differ.

Visualization of karate graph is shown in figure 1. As we know the underlying community structure, we used k= 3 number of cluster, the clusters are displayed in different colors. We observe that the embeddings generated by **GEMSEC** and **Struc2vec** are better than other because the clusters are well separated.



K=3,SC = 0.262,CP= 0.043 ,SP= 0.124

(a) Diff2vec

K=3,SC = 0.468,CP= 0.475 ,SP= 2.275

(b) DeepWalk

K=3,SC = 0.663,CP= 0.612 ,SP= 3.221

(c)Gemsec

K=3,SC = 0.132,CP= 2.156 ,SP= 2.865

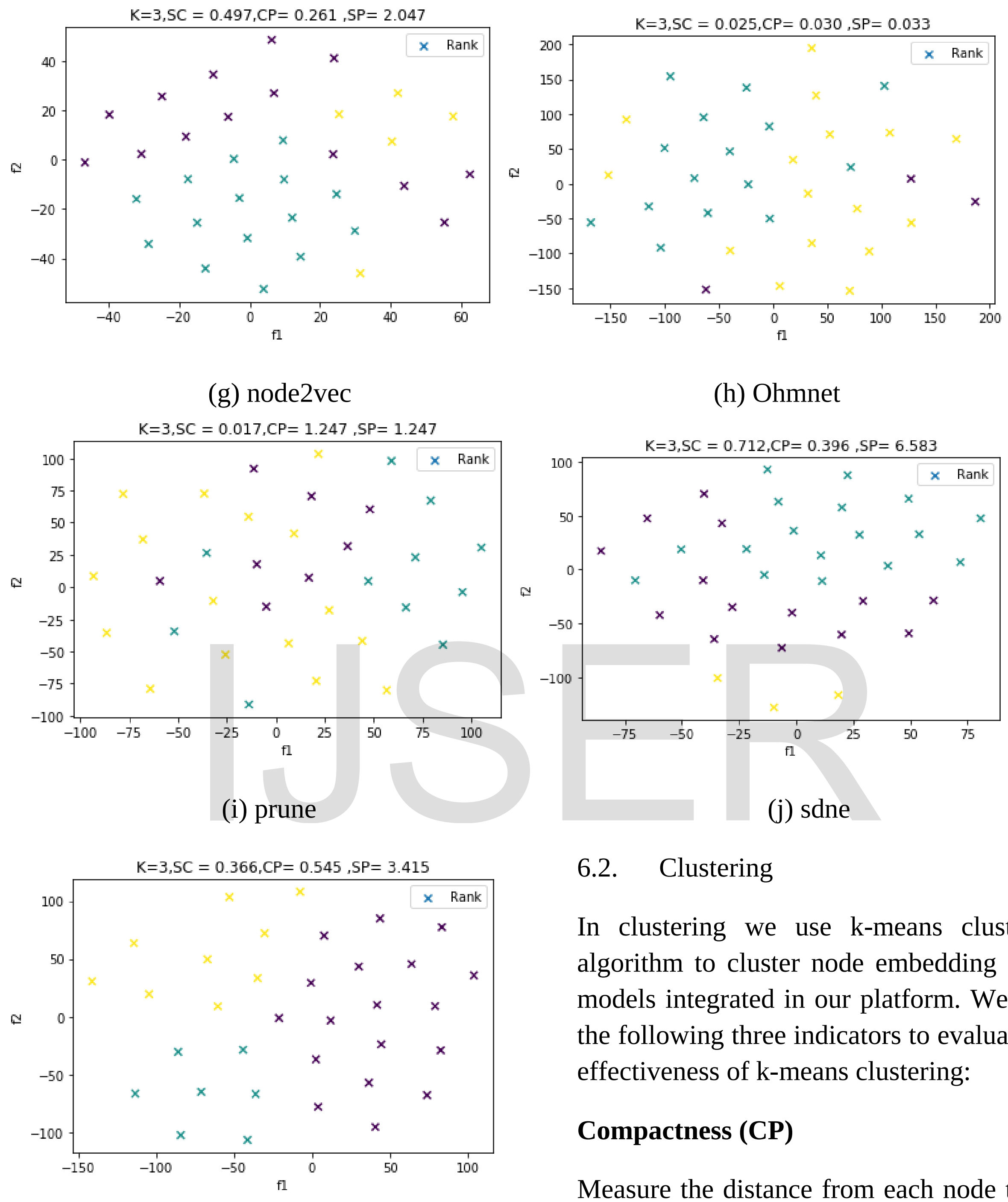
(d) Grarep

K=3,SC = 0.663,CP= 0.612 ,SP= 3.221

(e) Line

K=3,SC = 0.132,CP= 2.156 ,SP= 2.865

(f) mnmf



(g) node2vec

(h) Ohmnet

(i) prune

(j) sdne

(k) struc2vec

Figure 1: visualization of karate club using K-means clustering (original dimensional graph embedding is 128). Each point corresponds to a node in the graph. Color of a node denotes its cluster.

6.2. Clustering

In clustering we use k-means clustering algorithm to cluster node embedding of all models integrated in our platform. We have the following three indicators to evaluate the effectiveness of k-means clustering:

Compactness (CP)

Measure the distance from each node to the center point of the corresponding cluster. The low value of compactness means better clustering effect.

Separation (SP)

Measure the average distance between two or more cluster center points. High value of separation means better clustering effect.

Silhouette coefficient (SC)

Silhouette is the ratio of compactness and separation. Silhouette coefficient is better when Silhouette coefficient is close to 1. A good clustering effect needs to meet the following conditions:

- Silhouette coefficient close to 1, small value of compactness and large value of separation.

Methods	CP	SC	SP
DeepWalk	0.475	0.468	2.275
Line	0.696	0.025	2.075
Grarep	2.156	0.132	2.865
Node2vec	0.261	0.497	2.047
SDNE	0.396	**0.712**	**6.583**
OhmNet	**0.03**	0.025	0.033
MNMF	0.753	0.264	1.438
Diff2vec	0.043	0.262	0.124
Struc2vec	0.545	0.366	3.415
Gemsec	0.612	0.663	3.221
Prune	1.247	0.017	1.247

Table 1.K-means clustering of karate dataset

Methods	CP	SC	SP
DeepWalk	1.986	0.495	**8.838**
Line	0.9	0.014	0.657
Grarep	0.425	0.14	3.019
Node2vec	0.834	0.471	7.231
SDNE	3.747	0.206	8.025
OhmNet	**0.042**	**0.608**	0.661
MNMF	0.83	0.134	1.323
Diff2vec	1.372	0.251	3.813
Struc2vec	1.621	0.303	5.99
Gemsec	0.213	0.5	4.987
Prune	0.566	0.02	0.371

Table 2.k_means clustering of facebook

Methods	CP	SC	SP
DeepWalk	4.238	**0.292**	**12.922**
Line	0.779	0.099	2.025
Grarep	1.764	0.056	1.694
Node2vec	3.296	0.051	2.785
SDNE	5.254	0.035	3.923
OhmNet	1.051	0.174	1.937
MNMF	0.629	0.039	1.053
Diff2vec	2.649	0.066	3.133
Struc2vec	0.913	0.181	6.66
Gemsec	5.742	0.082	7.645
Prune	**0.132**	0.047	0.173

Table 3.k-means clustering of Wikipedia

- Table 1 (k-means clustering of karate data) we can see that the compactness value of OhmNet embedding is better than other models because it has small values of compactness. The separation values of SDNE embedding is highest than other models it means it has better clustering effect, again the silhouette values of SDNE is better because it has SC value is close to 1. On karate dataset, **SDNE** is better than other models, because it fulfill two indicators in three to evaluate effect clustering.
- Table 2 (k-means clustering of facebook), the compactness value of OhmNet is small than other methods and silhouette coefficient is close to 1. That means **OhmNe**t is better than other methods on facebook data.
- The table 3 shows the result of different models on Wikipedia data after k-means clustering, **DeepWalk** is better than other models because it has highest value of silhouette

coefficient which is close to 1 and the highest value of separation.

6.3. Link prediction

We use the link prediction task to verify the accuracy of integrated graph embedding models for predicting the existence of edges in the graph. In the link prediction task, we extract 90% of each dataset out of the dataset as a training set, we leave 10% of each dataset as positive sample test set and we randomly generate a set of 10% nonexistent edges as negative sample test set for each dataset. We take vector representation of all nodes and use logistic regression methods to predict the existence of edges in the test set. Link prediction accuracy of **M-nmf** and **diff2vec** models are good than others on karate club dataset, link prediction accuracy of **prune** model is good than other on facebook dataset and link prediction accuracy of **GraRep** model is better than other models on Wikipedia data. The prediction accuracy rate of 11 graph embedding models on different dataset is shown bellow.

Methods	karate	facebook	wikipedia
DeepWalk	0.2857	0.4815	0.8772
Node2vec	0.2857	0.5026	0.8662
OhmNet	0.2857	0.492	0.7944
Line	0.4285	0.4973	0.8407
Gemsec	0.4285	0.4838	0.5222
Grarep	0.5714	0.4938	**0.9258**
Sdne	0.5714	0.4991	0.5286
Prune	0.5714	**0.5343**	0.5303
Struc2vec	0.5714	0.4666	0.6855
M-nmf	**0.7142**	0.5131	0.5385
Diff2vec	**0.7142**	0.485	0.898

Table 4. Accuracy rate of 11 graph embedding models on different datasets.

7. Conclusions

In this research we integrated graph embedding models and evaluated their performance on common datasets. We didn't find a single graph embedding model that is best to all datasets by using evaluation metrics like link prediction, clustering and visualization.

This research has made the following contributions; first, we have developed a robust and expanded platform for graph embedding. This will give users of our platform to choose the model that works best for them depending on the dataset they have. We have evaluated the specifics with regards to performance of each model and compared them on different datasets. There are several methods that are yet to be integrated so this is just the first step towards the integration process and we are calling upon researcher to expand further the integration scope and as well as work on data standardization model and this will go a long way in solving the challenge of time and space complexity.

References

[1] L. C. Freeman, visualizing social networks, Journal of social structure 1 (1) (2000) 4.

[2] R. F. I Cancho, R. V. Sol’e, The small world of human language, Proceedings of the Royal Society of London B:Biological Sciences 268 (1482) (2001) 2261-2265.

[3] J. Leskovec, J. Kleinberg, C. Faloutsos, Graph evolution: Densification and shrinking diameters, ACM Transactions on Knowledge Discovery from Data (TKDD) 1 (1) (2007) 2.

[4] D. Liben-Nowell, J. Kleinberg, The link-prediction problem for social networks, journal of the Association for Information Science and Technology 58 (7) (2007) 1019–1031.

[5] C. H. Ding, X. He, H. Zha, M. Gu, H. D. Simon, A min-max cut algorithm for graph partitioning and data clustering, in: International Conference on Data Mining, IEEE, 2001, pp. 107–114.

[6] L. v. d. Maaten, G. Hinton, Visualizing data using t-sne, Journal of Machine Learning Research 9 (2008) 2579–2605.

P. Jaccard, Etude comparative de la distribution florale dans une portion des Alpes et du Jura, Impr. Corbaz, 1901.

[8] L. A. Adamic, E. Adar, Friends and neighbors on the web, Social networks 25 (3)(2003) 211–230.

[9] A. Clauset, C. Moore, M. E. Newman, Hierarchical structure and the prediction of missing links in networks, Nature 453 (7191) (2008) 98–101.

[10] H. C. White, S. A. Boorman, R. L. Breiger, Social structure from multiple networks. i. block models of roles and positions, American journal of sociology 81 (4) (1976) 730–780.

[11] N. Friedman, L. Getoor, D. Koller, A. Pfe_er, Learning probabilistic relational models, in: IJCAI, 1999, pp. 1300–1309.

[12] D. Heckerman, C. Meek, D. Koller, Probabilistic entity-relationship models, prms, and plate models, Introduction to statistical relational learning (2007) 201–238.

[13] Y. Zhou, H. Cheng, J. X. Yu, Graph clustering based on structural/attribute similarities, Proceedings of the VLDB Endowment 2 (1) (2009) 718–729.

[14] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Transactions on pattern analysis and machine intelligence 22 (8)(2000) 888–905.

[15] D. Wang, P. Cui, W. Zhu, Structural deep network embedding, in: Proceedings of the 22nd International Conference on Knowledge Discovery and Data Mining, ACM, 2016, pp. 1225–1234.

[16] M. Ou, P. Cui, J. Pei, Z. Zhang, W. Zhu, Asymmetric transitivity preserving graph embedding, in: Proc. of ACM SIGKDD, 2016, pp. 1105–1114.

[17] B. Perozzi, R. Al-Rfou, S. Skiena, Deepwalk: Online learning of social representations, in: Proceedings 20th international conference on Knowledge discovery and data mining, 2014, pp. 701–710.

[18] A. Grover, J. Leskovec, node2vec: Scalable feature learning for networks, in: Proceedings of the 22nd International Conference on Knowledge Discovery and Data Mining, ACM, 2016, pp. 855–864.

[19] S. Cao, W. Lu, Q. Xu, Grarep: Learning graph representations with global structural information, in: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, ACM, 2015, pp. 891–900.

[20] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, Q. Mei, Line: Largescale information network embedding, in: Proceedings 24th International Conference on World Wide Web, 2015, pp. 1067–1077.

[21] M. Zitnik, J. Leskovec, Predicting multicellular function through multi-layer tissue networks, arXiv: 1707.04638v1 [cs.LG] 14 Jul 2017.

[22] B. Rozemberczki, R. Davies, R. Sarkar, C. Sutton, GEMSEC: Graph Embedding with Self Clustering, arXiv : 1802.03997v1 [cs.SI] 12 Feb 2018.

[23] X. Wang, P. Cui, J. Wang, J. Pei, W. Zhu, S. Yang, Community Preserving Network Embedding, in: Proceedings

of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17), pp. 203-209.

[24] L. F. R. Ribeiro, P. H. P. Saverese, D. R. Figueiredo, struc2vec: Learning Node Representations from Structural Identity, in: KDD'17, August 13–17, 2017, Halifax, NS, Canada, 385-394

[25] B. Rozemberczki, R. Sarkar, Fast Sequence Based Embedding with Diffusion Graphs

[26] Y. An Lai, C. Chi Hsu, W. Hao Chen, PRUNE: Preserving Proximity and Global Ranking for Network Embedding, in: 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

[27] D. Jungnickel, T. Schade, Graphs, networks and algorithms, Springer, 2005.

[28] E. R. Gansner, S. C. North, An open graph visualization system and its applications to software engineering, Software Practice and Experience 30 (11) (2000) 1203–1233.

[29] G. Di Battista, P. Eades, R. Tamassia, I. G. Tollis, Algorithms for drawing graphs: an annotated bibliography, Computational Geometry 4 (5) (1994) 235–282.

[30] I. Herman, G. Melanc¸on, M. S. Marshall, Graph visualization and navigation in information visualization: A survey, IEEE Trans on visualization and computer graphics 6 (1) (2000) 24–43.

[31] P. Eades, L. Xuemin, How to draw a directed graph, in: Visual Languages, 1989. IEEE Workshop on, IEEE, 1989, pp. 13–17.

[32] K. Pearson, Liii. on lines and planes of closest fit to systems of points in space, The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 2 (11) (1901) 559–572.

[33] M. E. Newman, M. Girvan, Finding and evaluating community structure in networks, physical review E 69 (2) (2004) 026113.

[34] X. Xu, N. Yuruk, Z. Feng, T. A. Schweiger, Scan: a structural clustering algorithm for networks, in: Proceedings 13th international conference on Knowledge discovery and data mining, 2007, pp. 824–833.

[35] S. White, P. Smyth, A spectral clustering approach to finding communities in graphs, in: Proceedings of the 2005 SIAM international conference on data mining, SIAM, 2005, pp. 274–285.

[36] L. L¨u, T. Zhou, Link prediction in complex networks: A survey, Physica A: Statistical Mechanics and its Applications 390 (6) (2011) 1150–1170.

[37] M. Al Hasan, M. J. Zaki, A survey of link prediction in social networks, in: Social network data analytics, 2011, pp. 243–275.

[38] L. Katz, A new status index derived from sociometric analysis, Psychometrika 18 (1)(1953) 39–43.

[39] K. Yu, W. Chu, S. Yu, V. Tresp, Z. Xu, Stochastic relational models for discriminative link prediction, in: NIPS, 2006, pp. 1553–1560.