Alnaimy and Desouki *Journal of Big Data* (2022) 9:41
https://doi.org/10.1186/s40537-022-00595-2

Journal of Big Data

RESEARCH Open Access

Expanded graph embedding for joint network alignment and link prediction

MHD Samy Alnaimy[1*] and Mohammad Said Desouki[2]

*Correspondence:
samy.alnaimy@hiast.edu.sy;
samy.alnaimy@gmail.com
[1] Department of Informatics, Higher Institute for Applied Sciences and Technology, Damascus, Syria
Full list of author information is available at the end of the article

Abstract

Link prediction in social networks has been an active field of study in recent years fueled by the rapid growth of many social networks. Many link prediction methods are harmed by users' intention of avoiding being traced across networks. They may provide inaccurate information or overlook a great deal of information in multiple networks. This problem was overcome by developing methods for predicting links in a network based on known links in another network. Node alignment between the two networks significantly improves the efficiency of those methods. This research proposes a new embedding method to improve link prediction and node alignment results. The proposed embedding method is based on the Expanded Graph, which is our new novel network that has edges from both networks in addition to edges across the networks. Matrix factorization on the Finite Step Transition and Laplacian similarity matrices of the Expanded Graph has been used to obtain the embeddings for the nodes. Using the proposed embedding techniques, we jointly run network alignment and link prediction tasks iteratively to let them optimize each other's results. We performed extensive experiments on many datasets to examine the proposed method. We achieved significant improvements in link prediction precision, which was 50% better than the peer's method, and in recall, which was 500% better in some datasets. We also scale down the processing time of the solution to be more applicable to big social networks. We conclude that computed embedding in this type of problem is more suitable than learning the embedding since it shortens the processing time and gives better results.

Keywords: Social network analysis, Expanded graph, Network alignment, Link prediction, Cross-graph embedding, Finite step transition, Laplacian, Singular value decomposition

Introduction

Since social network analysis has been receiving massive attention in recent years, researchers have gone a step further and studied multiple social networks together to capture the impact between them. By studying social networks together, more features and information can be extracted and identified. The link prediction problem [1] is one of the most studied problems in this field. One can define the link prediction problem as estimating the probability of connecting two unconnected nodes in a network. There are many link prediction applications such as Recommendation Systems [2]. Methods of link prediction are divided into (1) similarity based [3–5], (2) artificial intelligent based

[6, 7], (3) correlation information based [8] or a mixture of them. Another well-studied problem with social networks is the network alignment problem [9]. Network alignment is a graph related problem listed under the general problem of Entity Matching [10]. It aims to align similar nodes between two separate graphs. Researchers in [11] classified network alignment features into (1) profile features, (2) content features, (3) network features. In this study, link prediction was carried out between nodes in one social network based on an analysis of existing link patterns between nodes in another social network. We achieved that by aligning the nodes between the two networks then predicting the missing links between the aligned nodes in the first network and vice versa. The link prediction uses the existing links between the correspondence nodes in the second network to predict links in the first network. During the network alignment task, only the network features were used while profile and content features were ignored. The main objectives of this research are to predict links in a social network based on known links in another network using only network features such that the prediction is not impacted by (1) the heterogeneity of the available attributes in each network, (2) the contrast between the information provided in each network due to the lack of updating changeable information, (3) the intention of social network users to provide incomplete or incorrect information in different social networks to avoid being tracked. The contributions in this study can be summarized by the following points:

- We provide a new cross-graph embedding method that catches the properties of two graphs.
- We use the proposed method in a framework that jointly applies network alignment and link prediction.
- We perform extensive experiments on public datasets and evaluate our method and peer's method [12] on many aspects to show how our method surpasses the existing method.

We have organized the remainder of the paper as follows; in "Related work" Section we surveyed the related work in the field of network alignment, link prediction and graph embedding. In "Notations" Section we provided the notations used in the solution. We introduced our method in "The proposed method" Section. In "Experiments" Section we summarized the experiments we applied to evaluate the proposed method. Finally, we outlined conclusions and future work in "Conclusion" Section.

Related work

Network alignment and link prediction are both well-studied fields. However, researchers usually study these problems in a separate manner. Three categories of features are involved in network alignment, some researchers studied them separately and others combined them in their research. Content features were studied separately by researchers in [13]. They question how important the published tweets are for node alignment. They extracted text features from the tweets and posts such as high-frequency words, part-of-speech tags and emoticons then trained a classifier to predict the alignment. Researchers in [14] projected the network structure features and attribute features into a heterogeneous information network then embedded the generated network vertices to

classify each vertices pair as an alignment or not using highway networks [15]. Researchers in [16] introduced how deep learning can be employed in matching the content features based on the users' posts and tweets. They provided a design space using character or word embedding and RNN or Attention summarization. Recently, researchers in [17] aligned nodes between two directed networks based on the structural and attributes features of the two networks. They embed the structure of the networks using embedding learning based on the nodes and their input edges and output edges. Then they obtained the attributes embedding using a multi-layer auto-encoder. Finally, they unified the two embeddings using an attention layer to produce one embedding to each node which holds the structure and attribute information of the node. Besides that, there are many works in link prediction. In [18], researchers provided a survey and a comparison between many link prediction methods, features and network types. They explored link prediction using local and global similarities, node or link attributes and much more. Researchers in [19] projected the graph snapshot as an entry in a time series. They applied graph embedding learning on this time series using the Long Short Term Memory network to predict the network links in the next time step. In [20], researchers treated the problem as supervised learning of missing edges. In the graph embedding field, researchers in [21] introduced the embedding throw analyzing the similarity matrix such as factorizing it with singular value decomposition SVD. In [22], the researchers did a biased random walk to extract the nodes and their neighbours as sequences using a sampling strategy then learned the embedding. Researchers in [23] wanted to take the advantage of node's attributes to feed the embedding, so they embed the nodes taking in mind their attributes to fully describe the nodes. Local attribute distributions of the node's neighbours in a fixed hop have been passed to a Skip-Gram model to embed the attributed nodes. In [24], researchers studied anonymized user identity linkage between two separate networks using unsupervised embedding on network features. They designed a cross-network embedding model to learn from both networks the Gaussian embeddings. Next, they calculated the distance between the embeddings using the 2^{th} Wasserstein distance to link the users between networks. Researchers in [12] studied two separate networks together to solve the problem of link prediction in a network based on another network. They used Skip-Gram embedding to embed both network's nodes then used greedy alignment to align networks nodes and finally predict links in each network around the aligned nodes. They repeated those steps so that link prediction and node alignment can support each other's results. The limitation of the literature studies in network alignment is the dependency on network features in addition to other features because when social network users provide inaccurate information in their profiles or social content, the usage of content and profile features is useless; therefore the method that we demand must use only network features. In addition, we want to embed the nodes without taking into account their attributes so that incorrect information in the attributes does not negatively impact the embedding. Accordingly, embedding using SVD decomposition is a suitable method if applied to the network nodes similarity matrix. [12] suffers from two problems: (1) using Skip-Gram embedding for unsupervised embedding learning increases processing time for large networks in particular, (2) predicting missing links by examining the existing links between aligned nodes has poor precision and recall. To address these issues, we devised a new embedding method that

depends on calculated embedding based on the networks similarity matrix to minimize embedding time and improve link prediction quality.

Notations

We will provide in this section the problem formal definition and the symbols used in the upcoming sections.

A graph (network) $G = (V, E)$ can be defined as a set of vertices V and a set of edges E. We assume that $E \subseteq V \times V$ and $N = |V|$ the number of vertices. In social networks, the vertices represent the users and the edges represent the relationship between those users (friendship, follow relationship, ...).

Node Embedding is defined as $emb : V \rightarrow R^d$ where embedding dimension $d \ll N$, and emb_u is the embedding of $u \in V$.

Given two networks $G_1 = (V_1, E_1)$,$G_2 = (V_2, E_2)$ we define the alignment from G_1 to G_2 as $ali : V_1 \rightarrow V_2$. So $ali(u_1) = u_2$ where u_1 and u_2 belong to V_1 and V_2 respectively.

S is the set of the alignment seeds that the system will consider as a ground truth. $S = \{(u_1, u_2) : u_1 \in V_1, u_2 \in V_2, ali(u_1) = u_2\}$.

k-hop neighbors of a node $u \in V$ are all nodes that are exactly k links away from node u and we will denote them as $N^k(u)$.

The proposed method

The overall proposed method is roughly summarized in Algorithm 1. All these steps will be explained in detail in the upcoming subsections. These steps are identical to the steps proposed in [12] while our study mainly contributed in the node embedding step to reduce the embedding time by introducing our novel Expanded Graph and embed its nodes using some equations. Also [12] depends on network and profile features where we depend only on network features in order to solve the problem of invalid or wrong profile data.

Algorithm 1: Overall overview of the proposed method

Data: Two graphs $G_1 = (V_1, E_1)$,$G_2 = (V_2, E_2)$ and the alignment seeds set S
Result: Predicted alignment set $S^{'}$, predicted edges $E_1^{'}$,$E_2^{'}$

1 Initialize obtained alignment set $S^{'} = S$ and obtained edges sets $E_1^{'} =$,$E_2^{'} =$;
2 **while** $|S^{'}| < |V_1|$ *and* $|S^{'}| < |V_2|$ **do**
3 Node Embedding: Construct an Expanded Graph $G = (V, E)$ from G_1 and G_2 then calculate the nodes embedding $emb_u : u \in V_1, emb_v : v \in V_2$ using the proposed method;
4 Network Alignment: Align top similar Nodes using greedy alignment on the embedding similarities (feed $S^{'}$ with more aligned nodes);
5 Link Prediction: Predict links based on the obtained alignments using supervised learning (feed $E_1^{'}$,$E_2^{'}$ with predicted links);
6 **end**

Node embedding

There are many embedding methods, some of them form the embedding from one graph features [21, 22, 25] and others form it from multiple graphs features [12]. We introduced a novel two graphs embedding that forms the embedding using the two graphs features in a form of one graph which we named Expanded Graph. After constructing the Expanded Graph, we will derive the similarity measures from its adjacency matrix

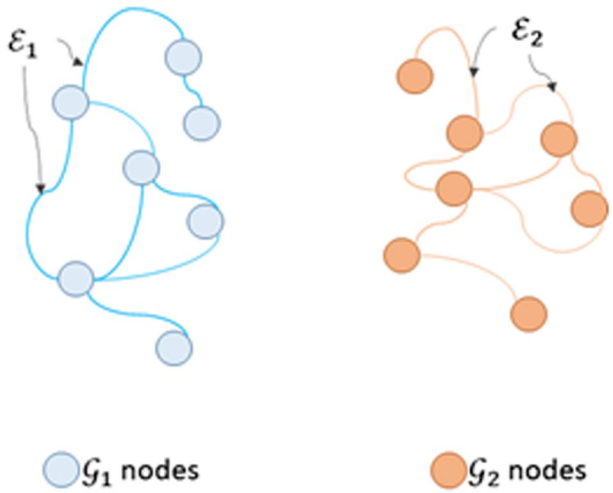

Fig. 1 Expanded graph—phase 1

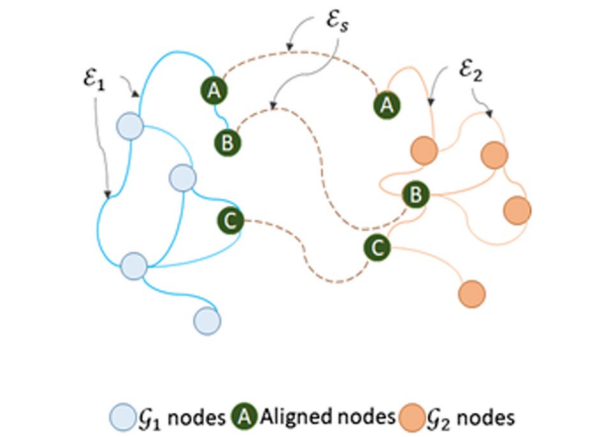

Fig. 2 Expanded graph—phase 2

then we will compute the nodes embedding by applying some equations on the similarity measure matrix.

Construct the expanded graph

Given two graphs $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$ and the alignment seeds set S, we constructed a new graph $G = (V, E)$. We named G as the Expanded Graph of G_1 and G_2. We assumed that $V = V_1 \cup V_2$. E can be initially represented as: $E = E_1 \cup E_2$. It basically consists of the existing edges in both graphs E_1 and E_2. Figure 1 shows the Expanded Graph in the phase 1. To embed across networks, we should add cross-network edges so we added them as E_s . Given seeds set S, E_s is a set of edges that we added between the aligned nodes in V_1 and their correspondence nodes in V_2. Figure 2 shows the Expanded Graph in the phase 2 so for now $E = E_1 \cup E_2 \cup E_s$. Also, we added E_{sim} where it is a set of edges that link each node in V_1 and the top h similar nodes in V_2 and vice versa.

So finally the Expanded Graph edges can be represented as $E = E_1 \cup E_2 \cup E_s \cup E_{sim}$. Figure 3 show the Expanded Graph in the final version. E_{sim} can be written as $E_{sim} = \{(u,v) : u \in V_1, v \in topNeb_{(h,G_1,G_2)}(u)\} \cup \{(u,v) : u \in V_2, v \in topNeb_{(h,G_2,G_1)}(u)\}$ where $topNeb_{(h,G,G')}(u)$ are the top h similar nodes in graph G' to the node u from graph G. We defined the similarity between any two nodes u,v from graphs G,G' respectively as the subgraph similarity $sim_{sub}(u,v)$ [12, 26] which is given by the Eq. 1.

$$sim_{sub}(u,v) = e^{-\alpha . f_k(u,v)} \tag{1}$$

where α is a hyperparameter that controls the distribution of the similarities and $f_k(u,v)$ is the distance between u and v in the k-hop which is given by Eq. 2.

$$f_k(u,v) = \sum_k \frac{dist(s_k(u), s_k(v))}{2} \tag{2}$$

$s_k(u)$ is the ordered degree sequence of $N^k(u)$ (u neighbors in the k-hop) and dist measure the distance between these sequences. For dist we used Eq. 3.

$$\begin{aligned} dist(s_k(u), s_k(v)) = & |min_{d \in s_k(u)} log(d+1) - min_{d \in s_k(v)} log(d+1)| \\ & + |max_{d \in s_k(u)} log(d+1) - max_{d \in s_k(v)} log(d+1)| \end{aligned} \tag{3}$$

Forming the similarity measure

We have proposed two similarity measures to use and later in "Node embedding" section we will show the evaluation of those measures and some other measures and how they affect the precision and recall of network alignment and link prediction.

Finite step transition matrix Finite Step Transition matrix FST [25] is equivalent to walking L steps sequentially and randomly as a random walker. FST is given by Eq. 4 Where P is the transition matrix which is given by Eq. 5. A is the adjacency matrix and

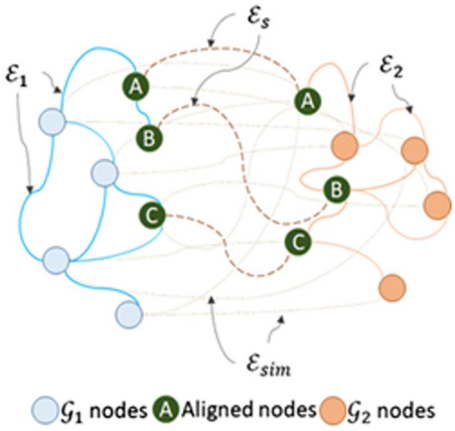

Fig. 3 Expanded graph—Final phase

D_{diag} is the diagonal degree matrix where all values are 0 except the diagonal as they are equal to the degree of the correspondence nodes.

$$FST^L = \sum_{l=1}^{L} P^l \tag{4}$$

$$P = D_{diag}^{-1} A \tag{5}$$

Laplacian matrix Laplacian Matrix LAP is given by Eq. 6 where A is the adjacency matrix and D is the degree matrix.

$$LAP = D - A \tag{6}$$

Calculating the embedding

We calculated the embedding using Singular Value Decomposition SVD of the similarity measure. SVD is one form of Matrix Factorization methods where the matrix M can be represented as Eq. 7.

$$M = U \Sigma V^* \tag{7}$$

SVD has been selected because it represents the given matrix by three matrixes where V represents a vector for each node where the values of this vector are ordered by the importance in representing the correspondence node according to the Eigen Values in Σ . Therefore, we used the truncated version of SVD where just an exact number of Eigen Values are selected. We set this truncation length to be equal to the number of embedding dimensions [27]. So V has been truncated to V_t then the embedding can be obtained by Eq. 8.

$$emb = M.V_t \tag{8}$$

Network alignment

After embedding each node in V_1 and V_2, we used the greedy alignment [28] to align the nodes between each graph. We calculated the similarity between each node embedding $emb_u : u \in V_1$ and $emb_v : v \in V_2$ using the cosine similarity given by Eq. 9, then we selected the top similar nodes and added them to the obtained aligned nodes set $S^{'}$.

$$sim_{emb}(u, v) = \frac{emb_u^t . emb_v}{|emb_u| . |emb_v|} \tag{9}$$

Link prediction

Link prediction in a graph is used to predict the probability of linking two nodes where there is no observed link between them. Given two graphs G_1 and G_2, we are

looking to predict this probability on the unobserved links between any two nodes in G_1 where $u, v : (u, v) \notin E_1$ meanwhile there is already an observed link between their alignments in G_2 $(ali(u), ali(v)) \in E_2$. To achieve that we used supervised learning as Logistic Regression LR. We trained the model with the existing edges in E_1 and E_2 then predict the unobserved edges in each graph as described before. As features, we feed the LR with the embedding of each node of the link and the Hadamard product (element-wise product) between these embeddings [12]. The result of the LR prediction will be in [0,1], we will only consider adding all predicted edges that their scores are above a threshold t to each graph G_1 and G_2.

So according to Algorithm 1, these three main steps will be repeated as iterations. In each iteration, we will embed all the nodes, align nodes between the networks to expand the alignment seeds and predict links in each network to expand the two graphs with more links. These expansions will still feed the network and enhance the precision and recall more and more in each iteration.

The proposed method variations

From the three main steps described before, we proposed three variation of the proposed method that share the network alignment and link prediction steps and differ in the node embedding step. Those variations are:

- EG-FST: It use the FST similarity measure.
- EG-LAP: It use the Laplacian similarity measure.
- EG-Mini: A customized version of EG-FST without recalculating the embedding in each iteration and without adding E_{sim} edges set. We added this method to evaluate the effect of these two factors in the overall process.

Experiments

We validated our method by running extensive experiments on many datasets to verify the network alignment precision and recall and link prediction precision and recall. We also evaluated the network size effect on time consumed by the proposed method variations. Finally as we used supervised learning, we evaluated how the training rate will affect the measures. All experiments ran on Google Colab on a device with 13GB memory, 2 physical 1-core CPUs (Intel(R) Xeon(R) CPU @ 2.20GHz) and 11441 MiB GPU (NVIDIA TESLA K80).

Datasets

We evaluated our method variations on three datasets Facebook/Twitter, Douban online/offline and DBLP/DBLP disturbed copy. All datasets statistics can be found in Table 1. Interoperability gives an idea of how the edges of the aligned nodes in the two networks overlap. Interoperability of two graphs G_1 and G_2 can be calculated with Eq. 10 where the intersection between two network edges is the intersection between the edges from the first network and the aligned edges from the second.

Table 1 Statistics for datasets used in experiments

Datasets	# Nodes	# Edges	Interoperability
Facebook	1043	4734	0.89
Twitter	1043	4860	
Douban online	1118	8164	0.76
Douban offline	1118	1511	
DBLP	2151	6306	0.94
DBLP disturbed copy	2151	5676	

$$interoperabilty(G_1, G_2) = \frac{2 * |E_1 \cap E_2|}{|E_1| + |E_2|} \tag{10}$$

All datasets consist of the nodes and edges of two networks.

- Facebook/Twitter: [29] has collected this dataset from about.me which is a platform that gathers different online social network accounts associated with the same user. Facebook/Twitter accounts have been collected and the graph map the users as nodes and friendship as edges.
- Douban online/offline: Douban is a Chinese online social network published by [30]. They took a subgraph from it (online) and constructed another graph based on real-world relations (offline). The alignment between the two networks is the match between the real world person and the Douban user.
- DBLP/DBLP disturbed copy: It is a co-authorship graph collected by [31] which consider the nodes as the authors and an edge links two authors if they have any common academic work. We used the same version used in [12] where researchers randomly generated another graph from it to align with the original while preserving the properties of the graph.

Evaluation protocols

For Network Alignment, we have evaluated the precision at the end of the iterations. Precision is how many aligned nodes have we aligned successfully on how many alignments we have made. Note that the network alignment recall in our proposed method will be the same of precision as in recall it will be how many aligned nodes we have aligned successfully on the overall aligned nodes. And since that we evaluate the network alignment through all the aligned nodes in the last iteration then the number of alignments we will make at the end of the iterations will be the same of the number of overall aligned nodes. Therefore, the network alignment precision and recall will be the same so we will only show the network alignment precision in the results.

For link prediction, we have also evaluated the precision and recall at the end of the iterations. The links that we should successfully predict is the set of links that do not exist in the first graph and exists between the aligned nodes in the second graph and vice versa. We have evaluated link prediction by comparing this set with the set of predicted edges in each network.

Datasets evaluation

We have evaluated the following methods on the three datasets and measured the network alignment precision and link prediction precision and link prediction recall. We have also evaluated how the training rate affects the measures. We define the training rate TR in our method as the rate of the seeds we start the proposed method with.

- CENALP: [12] proposed a cross-network skip-gram embedding with the same general algorithm for jointly align nodes and predict links. It is the latest and currently the most advanced method that do the two tasks jointly where all other methods just

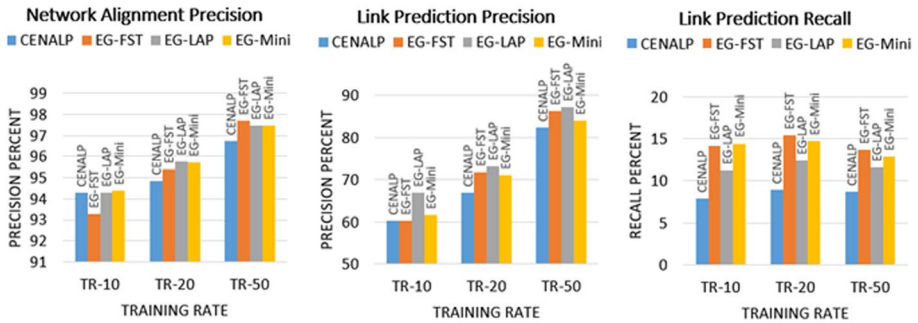

Fig. 4 Network alignment precision and link prediction precision and recall on Facebook/Twitter on CENALP, EG-FST, EG-LAP and EG-Mini

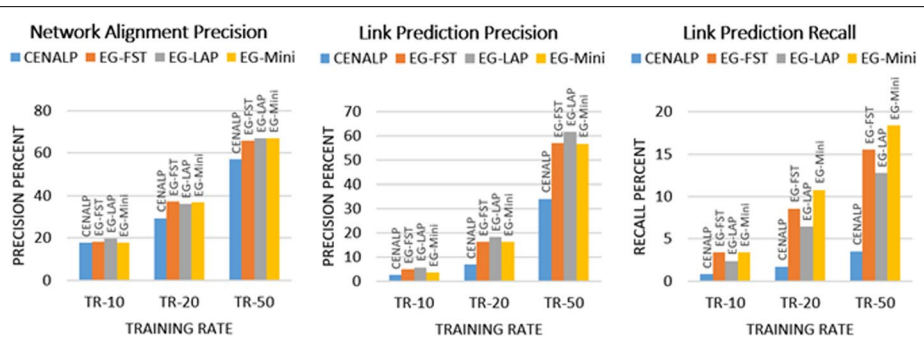

Fig. 5 Network alignment precision and link prediction precision and Recall on Douban on CENALP, EG-FST, EG-LAP and EG-Mini

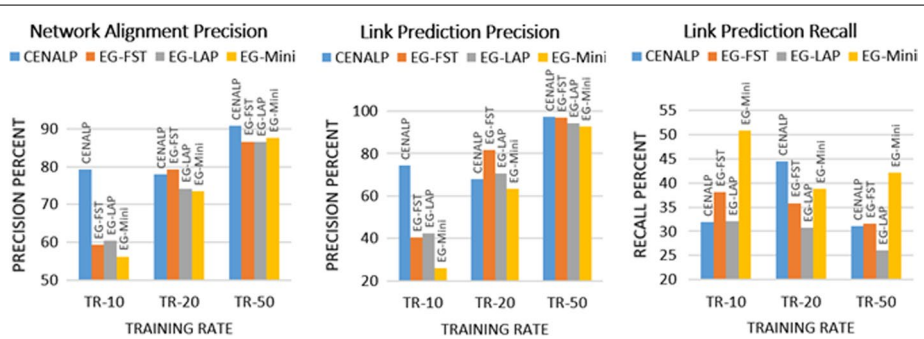

Fig. 6 Network alignment precision and link prediction precision and recall on DBLP on CENALP, EG-FST, EG-LAP and EG-Mini

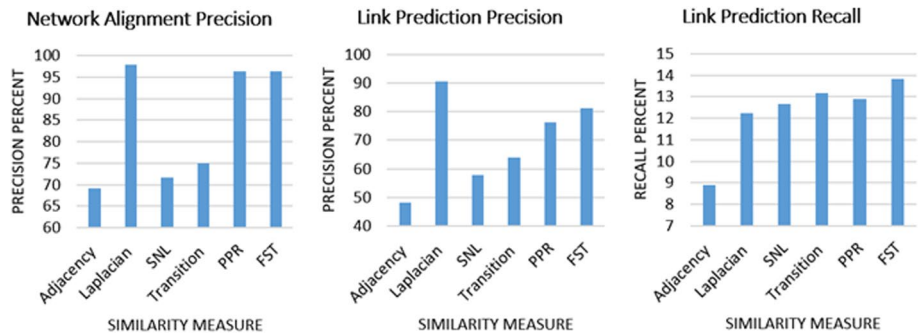

Fig. 7 Network alignment precision and link prediction precision and recall on Facebook/Twitter to compare similarity measures

do one task or do the two tasks separately. We used the parameters as discussed in the paper ($\alpha = 5, K = 3, q = 0.2, c = 0.5$)

- EG-FST: Our proposed method using the FST similarity measure. We set the parameters as ($\alpha = 5, K = 3, t = 0.5$)
- EG-LAP: Our proposed method using the Laplacian similarity measure. We set the same parameters as EG-FST.
- EG-Mini: A customized version of EG-FST without recalculating the embedding in each iteration and without adding E_{sim} edges set. We set the same parameters as EG-FST

As there is some random initializing in those methods, we have ran them 10 times and took the average value of each measurement. Figures 4, 5 and 6 show that all of our proposed methods outperform the CENALP in Facebook/Twitter and Douban but failed at DBLP. This case is intuitively reasonable since Facebook/Twitter and Douban are real-world people social network that friendship or follow relationship are the edges while in DBLP the edges are based on the co-authorship between papers. We can infer that as expected: as TR increases, we get more gain. Also, the results show that the big benefit comes in the Link Precision recall as the gain was 500% in Douban and 180% in Facebook/Twitter with comparison to CENALP. Douban shows that CENALP failed when the interoperability is somehow low but our methods did not get affected by this factor. In addition, EG-Mini shows that the factor of repeating the embedding throw iterations and adding E_{sim} edges set does not really affect the results with much differences.

Comparing multiple similarity measures

We have compared the similarity measures used in our proposed methods with another similarity measures. We have tested the following similarity measures:

- Adjacency Matrix: the default adjacency matrix which measure the distance between nodes as 1 if they are connected and 0 if they are not.
- Laplacian Matrix: as described in "The proposed method" Section

- Symmetric Normalized Laplacian Matrix SNL: calculated with Eq. 11 where D is the degree matrix and A is the adjacency matrix.

$$SNL = D^{-\frac{1}{2}}.A.D^{-\frac{1}{2}} \tag{11}$$

- Transition Matrix: as given in "The proposed method" Section
- Personalized PageRank Matrix PPR: Given by Eq. 12 where I is the identity matrix and P is the transition matrix and α is a hyperparameter (we used 0.1 as its value)

$$PPM = \alpha(I - (1-\alpha)P)^{-1} \tag{12}$$

- Finite Step Transition Matrix FST: as described in "The proposed method" Section.

Figure 7 shows the results of network alignment precision and link prediction precision and link prediction recall. As we can find, the best three methods in all measurements are Laplacian, Personalized PageRank and Finite Step Transition. We can infer that if we are looking for more link prediction precision we can use the Laplacian similarity measure, and for more link prediction recall we can use the Finite Step Transition similarity measure.

Network size effect on time

We tested the proposed method against other methods for network size scalability to show how the overall time will be affected. We ran the experiments on five datasets extracted from the DBLP dataset since it is large enough to form these sub-datasets from. Statistics of the five datasets can be found in Table 2. The methods we have compared are:

- CENALP/5 [12]: It is the CENALP method as described before, but since this method has a very far duration value range (because of the skip-gram training in

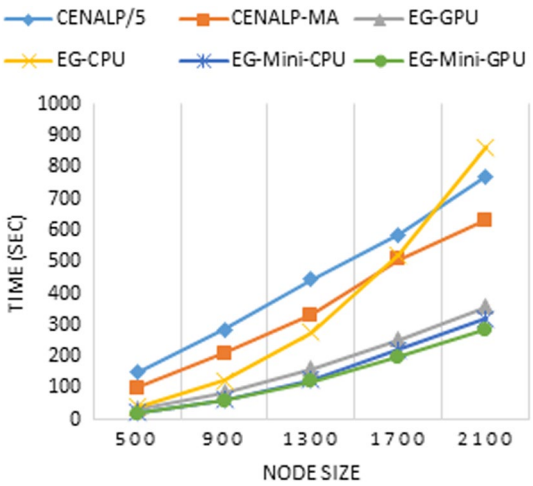

Fig. 8 Evaluate time vs. Network size

Table 2 Statistics of Datasets used in time evaluation

Datasets	# Nodes	# Edges
DBLP-500-1	500	1669
DBLP-500-2	500	1508
DBLP-900-1	900	3133
DBLP-900-2	900	2808
DBLP-1300-1	1300	4384
DBLP-1300-2	1300	3928
DBLP-1700-1	1700	5503
DBLP-1700-2	1700	4944
DBLP-2100-1	2100	6243
DBLP-2100-2	2100	5618

the embedding phase of each iteration), we have displayed its time measurement divided by 5 to scale its line down.

- CENALP-MA [12]: A variation of CENALP where they only apply the skip-gram embedding in the first iteration then used mean aggregate to recalculate the embedding in the upcoming iterations.
- EG-GPU: The EG-FST method implemented using GPU.
- EG-CPU: The EG-FST method implemented using CPU.
- EG-Mini-GPU: EG-Mini implemented using GPU.
- EG-Mini-CPU: EG-Mini implemented using CPU.

As Figure 8 shows, we can find how the SVD embedding scales down the time in a big manner, and how close the EG-GPU and EG-Mini-GPU even that EG-GPU recalculate the embedding in each iteration. The figure also reflects the behaviour of the compared methods on very big datasets where the time will be growing exponentially in CENALP, CENALP-MA and EG-CPU while it is near-linear in EG-GPU, EG-Mini-GPU and EG-Mini-CPU.

Conclusion

We have proposed a new embedding technique that applies single graph embedding algorithms on one graph (the Expanded Graph) generated from multiple graphs. Our method variations have significantly improved link prediction precision by 50% using EG-Lap in addition to the improvement in link prediction recall by 500% in some datasets using EG-FST. Besides that, we used calculated embedding that has scaled down the time taken to be near-linear on big social networks. Thus, we improved the method pioneered by CENALP to achieve more accurate results while applying network alignment and link prediction simultaneously. There is still too much work to do as future work, such as supporting directed graphs, weighted graphs, attributed edges in addition to improving the measurements to further increase link prediction precision and recall.

Acknowledgements
Not applicable.

Author contributions
MSN has first author role so he performed the literature review, implemented the proposed model, carried out the experiments and wrote the manuscript. MSD has a supervisory role, He contributed to the overall planning, provided

critical feedbacks and helped organizing the research and manuscript. Both authors read and approved the final manuscript.

Funding
Not applicable.

Availability of data and materials
Source code and used datasets are available at the author's Github https://github.com/samnemo94/Expanded-Graph.

Declarations

Ethics approval and consent to participate
Not applicable since the social networks datasets are anonymous and only include nodes and edges without any other attribute.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Author details
[1]Department of Informatics, Higher Institute for Applied Sciences and Technology, Damascus, Syria. [2]Faculty of Informatics and Communication Engineering, Arab International University, Daraa, Syria.

Received: 15 October 2021 Accepted: 28 March 2022
Published online: 21 April 2022

References

1. Liben-Nowell D, Kleinberg JM. The link-prediction problem for social networks. J Assoc Inf Sci Technol. 2007;58(7):1019–31. https://doi.org/10.1002/asi.20591.
2. Su Z, Zheng X, Ai J, Shen Y, Zhang X. Link prediction in recommender systems based on vector similarity. Phys A Stat Mech App. 2020;560:125154.
3. Zeng R, Ding Y, Xia X. Link prediction based on dynamic weighted social attribute network. In: International Conference on Machine Learning and Cybernetics, ICMLC 2016, Jeju Island, South Korea, July 10-13, 2016. IEEE; 2016. p. 183–188. 10.1109/ICMLC.2016.7860898
4. Sarukkai R. Link prediction and path analysis using Markov chains. Comput Netw. 2000;33(1–6):377–86. https://doi.org/10.1016/S1389-1286(00)00044-X.
5. Amin MI, Murase K. Link Prediction in Scientists Collaboration with Author Name and Affiliation. In: 2016 Joint 8th International Conference on Soft Computing and Intelligent Systems (SCIS) and 17th International Symposium on Advanced Intelligent Systems (ISIS), Sapporo, Japan, August 25-28, 2016. IEEE; 2016. p. 233–238. 10.1109/SCIS-ISIS.2016.0058
6. Yin L, Zheng H, Bian T, Deng Y. An evidential link prediction method and link predictability based on Shannon entropy. Phys A Stat Mech Appl. 2017;482:699–712.
7. Bai L, Cui L, Bai X, Hancock ER. Deep depth-based representations of graphs through deep learning networks. Neurocomputing. 2019;336:3–12. https://doi.org/10.1016/j.neucom.2018.03.087.
8. Wang H, Hu W, Qiu Z, Du B. Nodes' evolution diversity and link prediction in social networks. IEEE Trans Knowl Data Eng. 2017;29(10):2263–74. https://doi.org/10.1109/TKDE.2017.2728527.
9. Zhang J, Yu PS. Multiple Anonymized Social Networks Alignment. In: Aggarwal CC, Zhou Z, Tuzhilin A, Xiong H, Wu X, editors. 2015 IEEE International Conference on Data Mining, ICDM 2015, Atlantic City, NJ, USA, November 14-17, 2015. IEEE Computer Society; 2015. p. 599–608. 10.1109/ICDM.2015.114.
10. Konda P, Das S, Doan A, Ardalan A, Ballard JR, et al. Magellan: toward building entity matching management systems over data science stacks. Proc VLDB Endow. 2016;9(13):1581–4.
11. Lee J, Hussain R, Rivera V, Isroilov D. Second-level degree-based entity resolution in online social networks. Soc Netw Anal Min. 2018;8(1):19. https://doi.org/10.1007/s13278-018-0499-9.
12. Du X, Yan J, Zhang R, Zha H. Cross-network skip-gram embedding for joint network alignment and link prediction. IEEE Transactions on Knowledge and Data Engineering. 2020; p. 1
13. Srivastava DK, Roychoudhury B. Words are important: a textual content based identity resolution scheme across multiple online social networks. Knowl Based Syst. 2020;195. https://doi.org/10.1016/j.knosys.2020.105624.
14. Kong C, Chen B, Zhang L. DEM: Deep entity matching across heterogeneous information networks. J Comput Sci Technol. 2020;35(4):739–50. https://doi.org/10.1007/s11390-020-0139-5.
15. Srivastava RK, Greff K, Schmidhuber J. Highway networks. CoRR. 2015;abs/1505.00387.
16. Mudgal S, Li H, Rekatsinas T, Doan A, Park Y, Krishnan G, et al. Deep learning for entity matching: a design space exploration. In: Das G, Jermaine CM, Bernstein PA, editors. Proceedings of the 2018 International Conference on Management of Data, SIGMOD Conference 2018, Houston, TX, USA, June 10-15, 2018. ACM; 2018. p. 19–34. 10.1145/3183713.3196926.
17. Yang F, Liang W, Zong L. Attribute network alignment based on network embedding. In: ICCDE 2021: 7th International Conference on Computing and Data Engineering, Phuket, Thailand, January 15—17, 2021. ACM; 2021. p. 75–80. 10.1145/3456172.3456217.

18. Pandey B, Bhanodia PK, Khamparia A, Pandey DK. A comprehensive survey of edge prediction in social networks: techniques, parameters and challenges. Expert Syst Appl. 2019;124:164–81. https://doi.org/10.1016/j.eswa.2019.01.040.
19. Goyal P, Chhetri SR, Canedo A. dyngraph2vec: Capturing network dynamics using dynamic graph representation learning. Knowl Based Syst. 2020. https://doi.org/10.1016/j.knosys.2019.06.024.
20. Al Hasan M, Chaoji V, Salem S, Zaki M. Link prediction using supervised learning. In: SDM06: workshop on link analysis, counter-terrorism and security. vol. 30; 2006. p. 798–805.
21. Berberidis D, Giannakis GB. Node embedding with adaptive similarities for scalable learning over graphs. IEEE Transactions on Knowledge and Data Engineering. 2019 07;p. 1
22. Grover A, Leskovec J. node2vec: Scalable feature learning for networks. In: Krishnapuram B, Shah M, Smola AJ, Aggarwal CC, Shen D, Rastogi R, editors. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016. ACM; 2016. p. 855–864. 10.1145/2939672.2939754.
23. Rozemberczki B, Allen C, Sarkar R. Multi-Scale attributed node embedding. J Complex Netw. 2021. https://doi.org/10.1093/comnet/cnab014.
24. Chu X, Fan X, Zhu Z, Bi J. Variational cross-network embedding for anonymized user identity linkage. In: Demartini G, Zuccon G, Culpepper JS, Huang Z, Tong H, editors. CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1—5, 2021. ACM; 2021. p. 2955–2959. 10.1145/3459637.3482214.
25. Chen S, Niu S, Akoglu L, Kovacevic J, Faloutsos C. Fast, Warped graph embedding: unifying framework and one-click algorithm. CoRR. 2017;abs/1702.05764. http://arxiv.org/abs/1702.05764.
26. Ribeiro LFR, Saverese PHP, Figueiredo DR. *struc2vec*: Learning node representations from structural identity. in: proceedings of the 23rd acm sigkdd international conference on knowledge discovery and data mining, halifax, ns, Canada, August 13—17, 2017. ACM; 2017. p. 385–394. 10.1145/3097983.3098061.
27. Vannieuwenhoven N, Vandebril R, Meerbergen K. A new truncation strategy for the higher-order singular value decomposition. SIAM J Sci Comput. 2012. https://doi.org/10.1137/110836067.
28. Kollias G, Mohammadi S, Grama A. Network similarity decomposition (NSD): a fast and scalable approach to network alignment. IEEE Trans Knowl Data Eng. 2012;24(12):2232–43. https://doi.org/10.1109/TKDE.2011.174.
29. Cao X, Yu Y. BASS: A bootstrapping approach for aligning heterogenous social networks. In: Frasconi P, Landwehr N, Manco G, Vreeken J, editors. Machine Learning and Knowledge Discovery in Databases—European Conference, ECML PKDD 2016, Riva del Garda, Italy, September 19-23, 2016, Proceedings, Part I. vol. 9851 of Lecture Notes in Computer Science. Springer; 2016. p. 459–475. 10.1007/978-3-319-46128-1_29
30. Zhong E, Fan W, Wang J, Xiao L, Li Y. ComSoc: adaptive transfer of user behaviors over composite social network. In: Yang Q, Agarwal D, Pei J, editors. The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, Beijing, China, August 12-16, 2012. ACM; 2012. p. 696–70410.1145/2339530.2339641.
31. Prado A, Plantevit M, Robardet C, Boulicaut J. Mining graph topological patterns: finding covariations among vertex descriptors. IEEE Trans Knowl Data Eng. 2013;25(9):2090–104. https://doi.org/10.1109/TKDE.2012.154.

Publisher's Note