Asian Journal of Probability and Statistics

14(2): 1-12, 2021; Article no.AJPAS.71361

ISSN: 2582-0230

Improving Business Pages Recommendation in Social Network Using Link Prediction Methods

Watare Asaph[1] **and Shaowei Sun**[1*]

[1] *School of Science, Zhejiang University of Science and Technology, Hangzhou, Zhejiang, 310023, PR China.*

Authors' contributions

This work was carried out in collaboration between both authors. Both authors read and approved the final manuscript.

Article Information

DOI: 10.9734/AJPAS/2021/v14i230322

Editor(s):

(1) Dr. Manuel Alberto M. Ferreira, Lisbon University, Portugal.

Reviewers:

(1) Sanjay L. Nalbalwar, Dr. Babasaheb Ambedkar Technological University, India.

(2) Hijaz Ahmad, International Telematic University Uninettuno, Italy.

Complete Peer review History: http://www.sdiarticle4.com/review-history/71361

Received: 01 June 2021

Accepted: 02 August 2021

Published: 06 August 2021

Original Research Article

Abstract

Recently Social Network has become one of the favorite means for a modern society to perform social interaction and exchange information via the internet. Link prediction is a common problem that has broad application in such social networks, ranging from predicting unobserved interaction to recommending related items. In this paper, we investigate link recommendations over business pages on Facebook Social Network. More specifically, given a company in the network, we want to recommend potential companies to connect with. We start by introducing existing work in link recommendations and some link prediction models as our baseline. We then talk about the Graph Neural Network model SEAL to make a link recommendations in the network. Our results show that SEAL outperformed the compared baseline model while reaching above 94% Area Under Curve accuracy in link recommendations.

Keywords: Social network; link prediction; facebook pages; graph neural network.

2010 Mathematics Subject Classification: 68R99; 62P99.

**Corresponding author: E-mail: sunshaowei2009@126.com;*

1 Introduction

Recently Social Network has become one of the favorite means for a modern society to perform social interaction and exchange information via the internet. It allows people and organizations to share their content without any limits. This is because people can communicate and express their comments, likes and interests via social networks since it provides a fast and easy way to share a variety of information. The increasing use of the social networking has greatly transformed how organizations (businesses) carry out online marketing [1]. Many businesses host brands communities on social network platforms such as Facebook and Twitter to engage their customers and encourage user-generated content [2]. In particular, Facebook business pages is a feature launched in 2007 to help business connect and interact with their customers. From 2018 there were 90 million businesses pages on Facebook. In addition of brand marketing this is also a great platform for businesses to collaborate on a basis of sharing information to be more innovative and competitive. As the results, it creates a complex connection within the social network and this can be immediately visualized as large graphs.

Link prediction is a problem in social network analysis that focuses on predicting links that are going to appear in the near future [3]. For instance, given a large networks, say Facebook, at time t and for each user we would like to predict what new edges (collaboration) that user will create between t and some future time t' . The problem can also be viewed as a link recommendations where in our case we aim to suggest business pages to other businesses [4].

Link recommendations is important. When a new business creates a business page, it has few connection with other businesses in the networks. Making an accurate recommendations will help the business to discover potential businesses to connect with and share information based on shared customers and brand communities. Second, suggesting pages based on the competition within same geographical area or industry can provide organization with information which help to benchmark their pages performance against competitors. In addition these pages can also act as source of idea for posting contents and brand marketing. Third, recommending the right business pages to connect with, can help improve user's experience and help the businesses in case of business to business(B2B) transaction.

Link prediction and link recommendation have some challenges. First, real networks are sparse where nodes have connection to only a very small fraction of all nodes in the networks. Second challenge is that for two businesses to connect, there are many reasons for them to collaborate. For instance the Facebook Social Networks, users might be friends because they study in the same institution. However, since they met in the institution they are likely to end up in the same career and they also probably live in the same town. The challenge is how to model the characteristics of users (age, gender, hometown, occupation) to predict the future links. As the result a link recommendations system that automatically learns features from the network structure instead of predefined ones to increase the link prediction accuracy tend to perform better.

Our motivation and goal is to explore the usage of friend recommendations algorithms to business-pages recommendation in social networks. Recently some algorithms based on link prediction are proposed for friend recommendations. Existing methods [5, 4] have the following advantages and disadvantages when applied to recommend business pages. First, link recommendations methods based on similarity score to predict links are simple and fast but most of the time accuracy of link prediction is low since they rely on assumptions when the new links will exist. Also they don't work well in all types of networks. Thus finding the best measure of similarity score is a trial and error process. Second, Latent methods aim to improve link prediction accuracy taking into account only the topological structure of the network.

However, especially on business pages networks, the relation varies over time, so its not enough to use latent features for a good prediction.

Based on the some of link prediction challenges and limitations of the previous works, we build upon the idea of Zhang and Chen [6] learning from Subgraphs, Embeddings and Attributes for Link prediction (SEAL), to make business pages recommendations to businesses on Facebook Social Network. The technique is suitable for business page link recommendations due to the following reasons: First it's based on learning heuristics from local subgraph automatically without predefined features, which contain enough information to learn graph structure feature for link prediction. This is effective considering learning from the entire business pages networks is often infeasible. Second the model is able to provide opportunity to include latent and explicit features when predicting links in the business pages networks which improves the accuracy when combined with the heuristics features from the networks structure. Experimental results shows that SEAL model has better performance than baseline models when used to improve the accuracy of link recommendation.

We believe that a primary contribution of the present paper is in the area of link prediction application where effective methods for link prediction are used to analyze business page social networks and suggest promising interaction or collaboration. Extensive real data-driven experiments are conducted to evaluate the proposed algorithms. Evaluation results are shown from different perspectives to provide insightful conclusions for real-world applications.

The remainder of this paper is organized as follows. Section 2 we will introduce existing work. We discuss data collection process, give a brief analysis of the "Facebook-pages-companies" networks, introduce baseline models and give more details of SEAL model in section 3. In section 4 we talk about the experiments and results. We conclude with possible future direction in section 5.

2 Related Works

There has been various methods proposed and studied for link prediction and recommendation. Recommendation is common functionality in Online Social Network used to recommend new user to another user in the same network. A recommendations system increases user experience by giving them more favorable social environment. The system help users to find relevant information and social connection within a short time. In most cases a list is ranked in decreasing order suggesting a number of other users in the network, places, item to a new or existing user based on common interests. The ranking is based on the several factors such as user's location, time, behavior and identity Daud et al. [7]. According to Campana and Delmastro [8], main approaches used for recommendation system are Context-Aware based, Content-based, Graph-based, Model-based, Memory-based and collaboration Filtering. Then later researchers started to integrate link prediction approach. Backstrom et al. [4] worked on link recommendation in Facebook social network and large collaboration networks using Supervised Random walks model. The proposed model was not limited to link recommendation only but also in other field such as anomaly detection. Also Papadimitriou et al. [9] built Friend link algorithm which exploited local and global similarity in social network. They noted that local algorithm outperformed the existing global-based friend recommendation algorithms in terms of time complexity. Furthermore, Dong et al. [10] proposed Ranking Factors Graph Model with network structure information. The model which recommended user across different networks (Epinions, Slashdot, Wikivote, Twitter and Facebook) was network depended that needed a more generic model to suit other network. Barbieri [11] developed a stochastic topic model named as Who to Follow and Why (WTFW) to recommend user in Twitter and flicker social networks. The model provided accurate link prediction and contextualized explanation to support the predictions. Finally Song et al. [12] modeled an enhanced link recommendation in signed network (Wikipedia, Slashdot and Epinions networks) using linear time probabilistic approach. However, the work considered latent feature of the networks and did not incorporate explicit features.

Link prediction methods are based on some assumption that measure the proximity between nodes to predict whether they are likely to have a link. They are also known as "heuristic methods". For instance, the common neighbor heuristic method assumes that two vertex are more likely to have a link in case they have more common neighbors [3]. Heuristic methods are focus on using network structure to calculate similarity scores and predict links. The method are grouped based on the most distant node necessary for computing the similarity score. Common Neighbors and Preferential attachment are categorized as first-order heuristic since they just need the direct neighbors of the two nodes [13]. Adamic and Adar [5] and Resource allocation [14] are among the second-order heuristic as the need two distant neighbors to compute the similarity score. Katz measure [15], rooted PageRank [16], and SimRank [17] are some of the higher-order heuristic methods because they require to search the entire graph for all possible paths between two nodes.

Latent methods were proposed to improve link prediction accuracy. Latent features are representations of nodes, often obtained by factorizing a specific matrix derived from a network, such as the adjacency matrix or Laplacian Matrix. Interaction among such latent feature determine the probability of link to happen [6]. Related to latent methods is Network embedding methods, because they also learn low-dimensional representations for nodes to predict a link in the network. Since network embedding methods also factorize matrix representations of networks [6] regarded them as learning more expressive latent features through factorizing some more informative matrices. Some of the state-of the art Latent feature method are, Matrix Factorization, Stochastic Block Model by Airoldi et al. [18], Node2vec(N2V) [19], LINE [20], Spectral Clustering, Variation Graph Auto-encoder [21]. Nickel and Ribeiro [22, 23] noted that latent features cannot capture structural similarity between nodes and they require an extremely large dimension to express some simple heuristics. In addition latent features cannot be transferred to new networks and they are less interpretable than similarity measures.

Recently, a new method which can automatically learn suitable heuristic from a network Weisfeiler Lehman Neural Machine (WLNM) by Zhang and Chen [24] was proposed. The WLNM is the noval approach that extract subgraph around two target nodes. The subgraph is named as the *enclosing subgraph* of a link. WLNM then represents the enclosing subgraph as an adjacency matrix. After that, a neural network is trained on these adjacency matrices to learn a link prediction model. This method made it possible for various heuristics embedded in the local patterns to be learned automatically, avoiding the need to manually select heuristics. Also, for those networks on which no existing heuristic works well, we can learn new heuristics that suit them. SEAL [6] was proposed by Zhang and Chen in 2018 to fix some limitations of WLNM. SEAL framework replaced fully-connected neural network in WLNM with a graph neural network (GNN), which enables better graph feature learning ability. SEAL also incorporated heuristic, latent and explicit node features, absorbing multiple types of information for the link prediction.

3 Materials and Methods

In this section we introduce the datasets, baseline models and finally explain details of SEAL method with reference to our problem.

3.1 Datasets

The datasets we use is the "Facebook-pages-Companies" which is the information collected about Facebook pages of different companies November 2017 [25]. The first file contains a list of edges indicating which companies have a link between them and the other file contains the node ids which is a list of the companies. Each company name has been anonymized by replacing the real name ids for each company with a new value ids. Since the datasets do not have node attributes, the network will have no features and also it will be unweighted graph.

We construct an undirected graph by adding a node for each unique company id in the datasets and adding an edge between two companies if they are connected. In total we have 14,144 nodes representing companies and 52,311 edges indicating that the network has 52,311 connection between different companies.

We first analyze the degree distribution of our graph as shown in Table 1. We can observe that most companies in the network connect to seven other companies. We can also observe that the largest connection between companies is 215 links and smallest number is 1 link. This analysis shows most of nodes in the network have small links compared to possible links thus proving natural imbalances of the classes. Other analysis we can make from companies connection network is possibility of new links. As we can observe the company pages network has a density of 0.000523452, indicating that only 0.052% possible connection. The Assortativity coefficient is the person correlation coefficient of degree between pairs of linked node according to [26]. The assortive of company pages network is a positive 0.012611 (assortive mixing) indicating that companies tend to connect to other companies with similar number of connections.

Table 1. Statistics summary of companies pages network

Density	0.000523452
Average degree	7
Maximum degree	215
Minimum degree	1
Assortativity	0.0126111

3.2 Baseline

We use heuristic scoring methods to create feature for supervised learning. We construct a 5-dimension feature vector for each pair of nodes. In this case, all features we use are network structural features, which include the common-neighbors, resource allocation index, Jaccard-coefficient, adamic-adar index and preferential attachment. Given the feature we are going to train a logistic regression classifier to predict a binary value indicating whether an edge between two nodes should exist or not. As supervised learning, we expect the model to take more time during training and they can also be slow at test and evaluation time. We give the baseline Ensemble (ENS) for reference purposes.

The second model will try to predict a link as supervised learning problem on top of node embedding. The embedding will be computed with unsupervised Node2vec algorithm which train a logistic regression classifier. We call the second baseline model Node2vec.

3.3 SEAL

A network can be represented as a graph $G = (V, E)$ which contains a set of vertices V indicating nodes, a set of edges E, indicating the relationships between two nodes. Its adjacency matrix is A, where $A_{ij} = 1 \in E$ if there is a link from i to j and $A_{ij} = 0 \in E$ otherwise. In this paper, we consider undirected network.

Therefore A will be symmetric. For any nodes $x, y \in V$, let $E_{x,y}$ be target link and $\Gamma^h(x)$ represent set of neighbors from node x at distance of hops (h).

SEAL contains three steps. First, extracting enclosing subgraph based on the hop number given by the user, construct the node information matrix X for each enclosing subgraph and finally, feed the adjacency matrix (input of enclosing subgraph) and information matrix to graph neural networks to classify links. We describe the procedure for extracting the enclosing subgraph in the following.

For a given link, its Enclosing subgraph is a subgraph within the neighborhood of that link. The size of the neighborhood is described by the number of vertices in the subgraph, which is denoted by a user-defined integer K. Given a target link $E_{x,y}$, we first add their one hop $\Gamma(x)$ and $\Gamma(y)$ to node list V_K. Then, nodes in $\Gamma^2(x)$, $\Gamma^2(y)$, $\Gamma^3(x)$, $\Gamma^3(y)$,..., are iteratively added to V_K until $V_K \geq K$ or there are no more neighbors to add. Then to achieve more accuracy, we increase hop number (h) and increase the K. Hope number and K was a hyperparameter, and finding the optimal combination is a manual process. Another drawback is that when the hope distance is increased, the size of the subgraph grew exponentially and increases processing complexity. In 2018 Zhang and Chen [6] proposed SEAL, and authors considered extracting all neighbors and not limiting the number of nodes in the subgraph. They also proved that with a small hop number the subgraph already contain enough information to learn good graph structure features for link prediction. SEAL tried only up to three hops to extract subgraps and empirically verified that the performance does not increase beyond that.

The second step is to construct the node information matrix X which contains: structural node labels, node embedding and node attributes. SEAL proposed a Double-Radius Node Labeling for the GNN to tell where the target nodes between which a link existent should be predicted. We describe the Double-Radius Node Labeling by Zhang and Chen [6] as follow. A node labeling is function $f_l : V \rightarrow \mathbb{N}$ which assigns an integer label $f_l(i)$ to every node i in the enclosing subgraph. The proposed method is based on the following equation.

$$f_l(i) = 1 + min(d_x, d_y) + (d/2)[(d/2) + (d\%2) - 1], \tag{3.1}$$

where $d_x = d(i, x)$,$d_y = d(i, y)$,$d = d_x + d_y$ and $(d/2)$ and $(d\%2)$ are the integer quotient and remainder of d divided by 2, respectively. First, assign label 1 to x and y. Then, for any node i with $(d(i, x), d(i, y)) = (1, 1)$, assign label $f_l(i) = 2$. Nodes with double-radius$(1, 2)$ or $(2, 1)$ get label 3. Node with double-radius $(1, 3)$ or $(3, 1)$ get 4. Nodes with $(2, 2)$ gets value 5. Node with $(1, 4)$ or $(4, 1)$ is assigned 6. Vertex with $(2, 3)$ or $(2, 3)$ get 7 as shown in Fig. 1. When calculating $d(i, x)$, y is temporally removed from the subgraph, and vice versa. The algorithm assigns nodes with smaller arithmetic mean distance to the target nodes small labels. If two nodes have the same arithmetic mean distance, the node with a smaller geometric mean distance to the target nodes get smaller labels.

SEAL model additionally include the 128-dimension node2vec embedding in the node information matrix$X$. Node2vec [19] is a simple algorithm for learning low-dimensional embedding for nodes in a graph by optimizing a neighborhood preserving objective. The objective is to learn similar embedding for neighboring nodes with respect to the network structure. [6] proposed negative injection as follows: Given $G = (V, E)$, a set of sampled positive training links $E_p \subseteq E$, and a set of sampled negative training links E_n with $E_n \cap E = \varnothing$. To get the embedding features on $G'(V, E \cup E_n)$ obtain by injecting the negative sample to G. This process allow the positive and negative training information captured in the embeddings to avoid GNN to optimize by fitting link existing information only. Since our network for link recommendation do not have node attributes, explicit features are not included therefore the model will learn from two types of features.

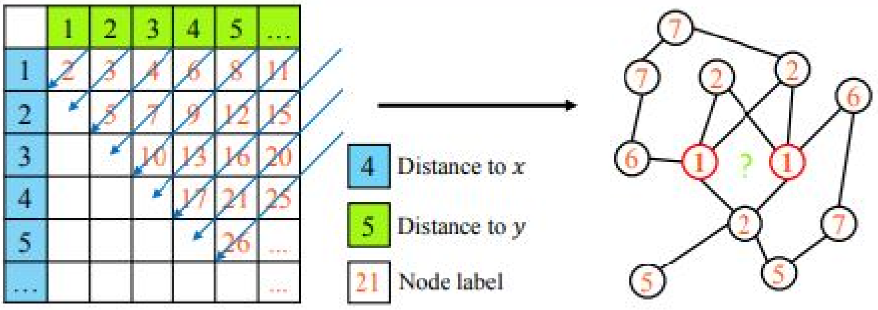

Fig. 1. Double-Radius Node Labeling source SEAL paper

The third step is feeding the adjacency matrix (input of enclosing subgraph) and information matrix to graph neural networks to classify links. SEAL model use Deep Graph Convolutional Neural Network (DGCNN) architecture [27]. DGCNN features a propagation based graph convolution layer to get node features, as well as a novel SortPooling layer which sorts node representation instead of summing them up. The sorting enables learning from grobal graph topology and retains much more vertex information than summing. Another advantage is that SortPooling layers supports back propagation, and sort nodes without using any processioning software enabling end-to-end training framework.

4 Evaluation

In this section we discuss how we construct the training and test datasets for our models, parameter tuning and the importance of difference features. We then evaluate the results by comparing the performance of baseline models and SEAL model. We use Area Under Curve (AUC) and Average precision (AP) as evaluation metrics. AUC and AP are evaluation metrics that are widely used in binary classification problem. Our baseline models and proposed model treats link recommendation as a binary classification problem by categorizing links in two groups, those that are positive links indicating existing of collaboration and negative links indicating no collaboration. A higher AUC and AP score represents the tested model is a better classifier. We also use runtime in terms of minutes (Mins) to compare the complexity of the models. We repeat the experiment for 10 times and report results for each experiment.

4.1 Ensemble models (ENS)

4.1.1 Datasets construction

We use the following process to find labeled node pairs where a pair of nodes is labeled positive if there exist an edge between them and is labeled negative otherwise. From the datasets we created the graph which has 14,114 nodes and 52,311 links. Then with the help of adjacency matrix we were able to find which pairs of node are connected. The network have 745,326 unconnected pairs. Those node pair will acts as negative sample during the training of link prediction. Then we create positive sample by removing links from connected node pairs. Here sample positive means all the business pages relations that is business pages with connected edges in the social network and sample negative all the negative relation meaning business pages with no connection edge in the business social network. The training set is the combination of positive and negative to get a datasets of relations. For feature combined each relation feature like Common neighbor are calculated and then used for training model. We use 90% observed links as training links and 10% as a testing links. Then we use logistic regression model using automatic hyperparameter selection [28] which is used to predict the probability of occurrence of an event by fitting data to a logit function.

Table 2 shows the results. We observe that the average AUC for the first baseline model score is 0.659, which means that the model prediction score is 65.9% correct. The average AP score is 55.6%. The analysis also shows that the model takes more than two hours to train and evaluate the results given the size of our network. This suggests that network with more potential links will take more time to train and evaluate.

Table 2. Evaluation of Ensemble model (Test set)

Exp.No	1	2	3	4	5	6	7	8	9	10
AUC	0.66	0.67	0.66	0.65	0.68	0.69	0.64	0.66	0.65	0.63
AP	0.58	0.54	0.55	0.56	0.55	0.58	0.54	0.56	0.56	0.54
Time (Mins)	161.0	161.2	161.3	161.5	161.0	161.1	161.2	161.0	161.2	161.0

4.2 Node2vec

4.2.1 Datasets construction

The Stellar-Graph [29] implementation of Node2Vec [19] algorithm is used to build a model that predict connection links of the business pages datasets. First we calculate link embedding of the source and target nodes for each sampled edge. Second given the embedding of the positive and negative examples, we train a logistic regression classifier to predict a binary value showing if a link between two nodes should exist or not. Third we evaluate the performance of the link classifier for each of the four operators (*Hadamand*, L1, L2 and *average*) on the training data with node embedding to select the best classifier. The paper give more details of these operators. Finally the best classifier is used to calculate the AUC and AP scores. The data was split into three sets. Training set, model selection and test set as shown in Table 3. to avoid data leakage and to evaluate the algorithms.

Table 3. Summary of different data splits

Split	No.Sample(links)	Use
Training Set	7060	Training the link classifier
Model Selection	2354	Select the best link classifier model
Test set	10462	Evaluate the best link classifier

4.2.2 Hyperparameters and results

First we generated 128-dimensional embedding from the network with 5 number of walks from each node, 20 length of each random walk, 6 number of SGD iterations (epochs) as suggested in [19]. We used L2 as the best operator and train a logistic regression with Liblinear using automatic hyperparameter selection [28].

Table 4 show the result. Node2vec shows significant improvement over Ensemble model with an average AUC score of 91.2% from 65.9% and AP score of 86% from 55.6%. Time required to train and to evaluate the model has reduced since it is below two hours an improvement from the first baseline model. This indicates that embedding model can discover the underlying structures of relations between the business pages in the network better than manually designed features thus making accurate prediction. The analysis also shows that a low dimension representation helps to save time for training and evaluation of the model and achieve better results.

Table 4. Evaluation of Node2Vec model (Test set)

Exp.No	1	2	3	4	5	6	7	8	9	10
AUC	0.913	0.910	0.912	0.920	0.910	0.913	0.910	0.913	0.913	0.911
AP	0.821	0.852	0.841	0.854	0.831	0.896	0.824	0.862	0.813	0.826
Time (Mins)	91.2	91.3	91.2	91.2	91.0	91.3	91.3	91.5	91.5	91.0

4.3 SEAL model

4.3.1 Datasets construction

We randomly remove 10% of existing links from the business page network as positive testing data. We also did a random sample of 10% unconnected node pairs as negative testing data following a standard manner of learning-based link prediction. 90% of the existing links as well 90% of additionally sampled nonexistent links to construct the training data from the remaining dataset.

4.3.2 Hyperparameters and results

In the experiment, we used Node2vec as default embedding and DGCNN architecture as the default GNN as suggested in [6]. We train DGCNN on subgraph extraction ($h = 1$) for 50 epochs, and selected the model with the smallest loss on the 10% validation data to predict the testing links.

Table 5 show the results. Firstly, we can observe that when compared with first baseline model (Ensemble), the accuracy of SEAL is more better with AUC score average of about 94.58% and that of AP 96.43% above the 65.9% and 55.6% respectively. This suggests that the SEAL model in terms of accuracy is a good at recommending pages to other companies. This indicate that heuristic features the model learns from the network are better than the one we predefined and used in our first model which works based on assumptions. We can also observe from the result that runtime has also improved when we use the SEAL model to recommend company pages. SEAL takes less than an hour unlike Ensemble model which took more than two hours. This is due to the fact SEAL train from subgraphs unlike the first model which try to learn the patterns from the entire network.

Next we compare SEAL with state-of-art embedding feature method Node2vec. Tables 4 and 5 shows the results. As we can see, SEAL with Node2vec embedding shows a significance improvement over pure Node2vec method which is our second baseline model. Node2vec has AUC score of 91.2% while SEAL accuracy AUC score is 94.5% indicating that we can improve the accuracy of link recommendation by 3% when we use graph structure and latent features simultaneously. This suggest that Node2vec model alone was not able to capture the most useful link prediction information located in the company pages network. However, in terms of runtime, second baseline model outperform SEAL model. One explanation is the fact that model does embedding before subgraph extraction. Second, negative injection $G'(V, E \cup E_n)$ will increase embedding time which important procedure of the model thus making a slight increase of runtime compared to Node2vec model. The pioneer of SEAL model noted that compared to SEAL without doing Node2vec embedding does not improve the accuracy of model thus for the purpose of our study which is to increase user experience among the company pages network SEAL proves to be working perfect.

Table 5. Evaluation of SEAL model (Test set)

Exp.No	1	2	3	4	5	6	7	8	9	10
AUC	0.944	0.943	0.932	0.950	0.955	0.950	0.944	0.949	0.946	0.945
AP	0.968	0.963	0.968	0.966	0.954	0.961	0.962	0.965	0.971	0.964
Time (Mins)	100.7	100.0	100.9	100.8	100.2	100.3	100.3	100.5	99.5	98.9

5 Conclusion

In this paper we introduced link prediction based on Graph Neural Network model SEAL to make recommendations to business pages on Facebook social networks which is not only relevant for the new businesses but will also improve user's experience and increase revenue generation in case of business to business transaction. The recommended businesses pages generated by proposed SEAL link prediction model utilize features learned from both the graph structure of the network and latent features simultaneously which ensure there is improvement in the accuracy of these recommendations. We have compared the Ensemble model and Node2vec embedding model on the same datasets with proposed SEAL model. Furthermore, we have evaluated all the three models on Area Under Curve (AUC) and Average Precision (AP) to compare the accuracy performance. The AUC shows that the recommendations by SEAL model are more improved than that of Ensemble and Node2vec baseline model. Therefore, the proposed SEAL model gives better results with high accuracy to recommend new connection.

Our datasets did not have node attributes thus limiting SEAL to use only graph structure and embedding. However, future work will be focus on the case including node/edges features to increase the accuracy and make more explainable link recommendations.

Disclaimer

The products used for this research are commonly and predominantly use products in our area of research and country. There is absolutely no conflict of interest between the authors and producers of the products because we do not intend to use these products as an avenue for any litigation but for the advancement of knowledge. Also, the research was not funded by the producing company rather it was funded by personal efforts of the authors.

Acknowledgement

The acknowledgements to people who provided assistance.

Competing Interests

Authors have declared that no competing interests exist.

References

[1] Aral, et al. Introduction to the special issue social media and business transformation: a framework for research. Information Systems Research. 2013;24(1):3-13.

[2] Borle, et al. The impact of Facebook fan page participation on customer behavior: an empirical investigation. Indian School of Business; 2013.

[3] Liben ND, Kleinberg J. The link-prediction problem for social networks. Journal of the American Society for Information Science and Technology. 2007;58(7):1019-1031.

[4] Backstrom L, Leskovec J. Supervised random walks: predicting and recommending links in social networks. 2011;635-644.

[5] Adamic LA, Adar E. Friends and neighbors on the web. Social Networks. 2003;25(3):211-230.

[6] Zhang M, Chen Y. Link prediction based on graph neural networks. arXiv preprint arXiv:1802.09691; 2018.

[7] Daud, et al. Applications of link prediction in social networks: A review. Journal of Network and Computer Applications. 2020;102716.

[8] Campana MG, Delmastro F. Recommender systems for online and mobile social networks: A survey. Online Social Networks and Media. 2017;3:75-97.

[9] Papadimitriou, et al. Fast and accurate link prediction in social networking systems. Journal of Systems and Software. 2012;85(9):2119-2132.

[10] Dong, et al. Link prediction and recommendation across heterogeneous social networks. 2012 IEEE 12th International Conference on Data Mining. 2012;181-190.

[11] Barbieri, et al. Who to follow and why: link prediction with explanations. Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2014;1266-1275.

[12] Song, et al. Efficient latent link recommendation in signed networks. Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2015;1105-1114.

[13] Newman ME. Clustering and preferential attachment in growing networks. Physical Review E. 2001;64(2):025102.

[14] Zhou, et al. Predicting missing links via local information. The European Physical Journal B. 2009;71(4):623–630.

[15] Katz L. A new status index derived from sociometric analysis. Psychometrika. 1953;18(1):39-43.

[16] Brin S, Page L. Reprint of: The anatomy of a large-scale hypertextual web search engine. Computer Networks. 2012;56(18):3825-3833.

[17] Jeh G, Widom J. Simrank: a measure of structural-context similarity. Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2002;538-543.

[18] Airoldi, et al. Mixed membership stochastic blockmodels. Journal of Machine Learning Research; 2008.

[19] Grover A, Leskovec J. Node2vec: Scalable feature learning for networks. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2016;855-864.

[20] Tang, et al. Line: Large-scale information network embedding. Proceedings of the 24th International Conference on World Wide Web. 2015;1067-1077.

[21] Kipf TN, Welling M. Variational graph auto-encoders. ArXiv preprint arXiv:1611.07308; 2016

[22] Nickel, et al. Reducing the rank in relational factorization models by including observable patterns. NIPS. 2014;1179-1187.

[23] Ribeiro, et al. Struc2vec: Learning node representations from structural identity. Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2017;385-394.

[24] Zhang M, Chen Y. Weisfeiler-lehman neural machine for link prediction. Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2017;575-583.

[25] Ryan AR, Nesreen KA. The network data repository with interactive graph analytics and visualization; 2015.
AVAILABLE:Http://network repository.com

[26] Newman ME. Assortative mixing in networks. Physical Review Letters. 2002;89(20):208701.

[27] Zhang, et al. An end-to-end deep learning architecture for graph classification. Proceedings of the AAAI Conference on Artificial Intelligence. 2018;32(1).

[28] Fan, et al. LIBLINEAR: A library for large linear classification. The Journal of machine Learning Research. 2008;9:1871-1874.

[29] CSIRO's Data61. StellarGraph machine learning library. GitHub Repository; 2018.
Available:https://github.com/stellargraph/stellargraph

Peer-review history:
The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)
http://www.sdiarticle4.com/review-history/71361