Link prediction in social networks by Variational Graph Autoencoder and similarity-based methods: a brief comparative analysis

S. Roy, A. Ranjan, S. Tomasiello

Stefania Tomasiello
stefania.tomasiello@ut.ee

UNIVERSITY OF TARTU
Institute of Computer Science

January 11, 2020

Overview

Social networks

Social networks

- model communication or interactions among many people;
- are represented as graphs, where each vertex maps to a user and each link corresponds to the relation between two users;
- are dynamic (and complex) systems, where the addition and/or deletion of several links and vertices take place many times every day.

Link prediction

Let G(V,E) be a graph of the network, where V is a vertex set and E(i,j) the link set. Let $G_{t0-t1}(V, E)$ be a graph representing a snapshot of a network during the time interval $[t_0, t_1]$ and E_{t0-t1}, a set of links in that snapshot. The link prediction problem is to find the set of links E_{t2-t3}during the time interval $[t_2, t_3]$ where $[t_0, t_1] \leq [t_2, t_3]$.

Let $\overline{n} = |V|$ denote the number of total vertices of the graph, and U denote the universal set which contains a total of $\overline{n}(\overline{n}-1)/2$ links (total node-pairs). The number $(|U| - |E|)$ represents the non-existing links, and some of them may appear in the near future. Finding such missing links is the aim of link prediction.

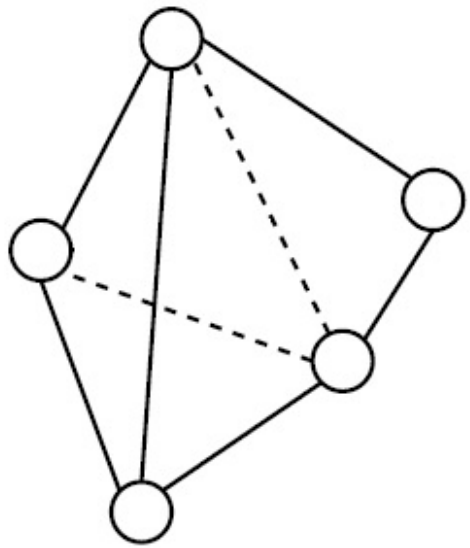

Figure: Link prediction: finding missing links

Some approaches

Several possible techniques:

- similarity-based (local, global, quasi-local);
- probabilistic-based (hierarchical random graph, stochastic block model, local model, exponential random graphs);
- dimensionality reduction-based (embedding-based, matrix factorization-based);
- other approaches (learning-based, clustering-based).

Similarity-based methods

For each pair i and j, a similarity score S(i, j) is calculated. The non-observed links (U-E) are assigned scores according to their similarities. The pair of nodes having a higher score represents the predicted link. The similarity indices can be

- local, which are generally calculated using information about common neighbors and node degree;
- global, which are computed using all the topological information of a network.

Some local similarity indices:

- Adamic/Adar index
- Jaccard's coefficient
- Preferential attachment

The Adamic/Adar index was originally designed to predict links in social networks and further adapted for the counting of common features

$$A(i,j) = \sum_{u \in \Gamma(i) \cap \Gamma(j)} \frac{1}{\log |\Gamma(u)|},$$

where $\Gamma(i)$ and $\Gamma(j)$ are neighbors of the node i and j respectively.

Jaccard's coefficient measures the probability that i and j have a common feature

$$J(i,j) = \frac{\Gamma(i) \cap \Gamma(j)}{\Gamma(i) \cup \Gamma(j)}.$$

Preferential attachment is based on the idea that the probability that a new edge involves the node i is proportional to the current number of neighbors of i. This is a very basic prediction method

$$P(i,j) = |\Gamma(i)||\Gamma(j)|.$$

According to a recent survey, this index shows the worst performance on most networks.

Variational graph autoencoder (VGAE)

The VGAE adapts the idea of Variational AutoEncoder (VAE) to handle graph-structured data. The VGAE model includes two stages:

- encoding, where it takes an adjacency matrix U, to represent the input graph, and a feature matrix Y, to represent the features of each node from the input graph, and gets a latent variable v as output,
- decoding, where it gets a reconstructed adjacency matrix according to the latent variable v.

Let us consider the sub-network H_n. Let N be the number of nodes in H_n, U_n the adjacency matrix, and C the degree matrix of U_n. In the encoding stage, the VGAE includes two layers of graph convolutional network (GCN). The first layer of GCN generates a lower dimensional feature matrix Y', with ReLU activation

$$Y' = ReLU(U'_n Y W_0)$$

where

$$U'_n = C^{-1/2}(U_n + I)C^{-1/2}$$

U' is the symmetrically normalized adjacency matrix, being I the identity matrix.

The second GCN layer generates the data distribution as follows:

$$\mu = U_n' Y' W_\mu, \qquad \log \sigma = U_n' Y' W_\sigma.$$

Then, for ε following N(0, 1), the latent variable is

$$v = \mu + \sigma \varepsilon.$$

The decoder is defined by an inner product between latent variables and the output is the reconstructed adjacency matrix $U_n^d = F(vv^T)$, where F is the sigmoid function. The loss function includes the binary cross-entropy between the target and the computed output.
A full-batch gradient descent was performed for training.

Numerical experiments

The dataset is from Google+ social circle data, publicly available at https://snap.stanford.edu/data/ego-Gplus.html.
Two ego networks have been extracted (see table below).
An ego network consists of a focal node (ego) and the nodes (alters) to whom ego is directly connected to, with the links, if any, among the alters.

Net ID	Diameter	Avg degree	Num edges	Density	Nodes
Net 1	2	20.724	3720	0.058	359
Net 2	2	45.042	14819	0.068	658

Table: Ego networks

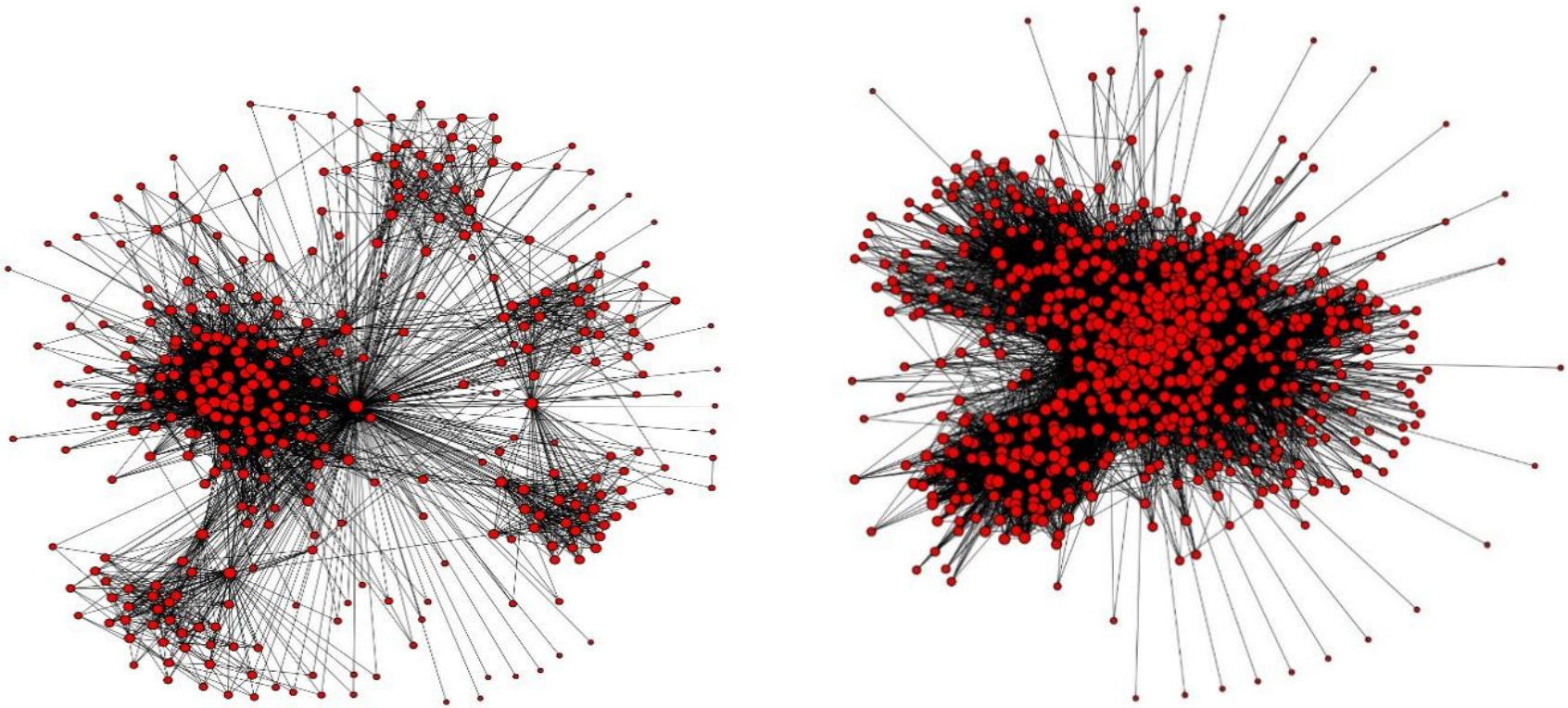

Figure: Ego Net 1 (left) Ego Net 2 (right)

Performance measures:

- precision P=TP/(TP+FP)
- recall R=TP/(TP+FN)
- F measure F= 2*P*R/(P+R)
- average precision (AP)
- area under the receiver operating characteristic (ROC) curve (ROC curve plots the true positive rate against the false positive rate)

An incomplete network was created for the model to be trained, by randomly removing some of the links. The dimension of the latent variable was 16 and 32 for the first and second network respectively. The total number of epochs for each network is 300.

approach	ROC (1)	AP (1)	ROC (2)	AP (2)
VGAE	0.898	0.910	0.861	0.867
Adamic/Adar	0.931	0.933	0.903	0.899
Jaccard Coefficient	0.813	0.745	0.747	0.703
Preferential attachment	0.868	0.880	0.866	0.856

Table: ROC (area) and AP values for different approaches and net 1 (1) and net 2 (2)

approach	P (1)	R (1)	P (2)	R (2)
VGAE	0.797	0.700	0.767	0.678
Adamic/Adar	0.964	0.448	0.901	0.530
Jaccard Coefficient	0.746	0.472	0.701	0.480
Preferential attachment	0.885	0.483	0.857	0.468

Table: Precision and recall values for different approaches and net 1 (1) and net 2 (2)

approach	(1)	(2)
VGAE	0.745	0.719
Adamic/Adar	0.611	0.667
Jaccard Coefficient	0.578	0.569
Preferential attachment	0.624	0.605

Table: F-measures for different approaches and net 1 (1) and net 2 (2)

Conclusions

A brief numerical study to compare a learning-based method (VGAE) against some local similarity-based methods (Adamic/Adar, Jaccard coefficient, preferential attachment) has been carried out.
One can notice that

- with regard to recall and F-measure, the VGAE outperformed the other models;
- with regard to the area under the ROC curve, the Adamic-Adar performs better than the other methods.

References

- Kumar, A., et al. Link prediction techniques, applications, and performance: A survey, Physica A, 553, 124289 (2020)
- Kipf, T., Welling, M. Variational graph auto-encoders. arXiv:1611.07308 (2016).
- Ding, Y., Tian, L. P., Lei, X., Liao, B., Wu, F. X. Variational graph autoencoders for miRNA-disease association prediction. Methods, in press (2020)

Thank you.