Friend Recommendation using Graph Neural Networks in Social Platforms

\*Note: Sub-titles are not captured in Xplore and should not be used

*Vasileios Karaiskos   
Laboratory team of Distributed   
MicroComputer Systems (MCSL),  
Dept. of Mathematics,  
University of Ioannina  
Ioannina, Greece  
vaskaraiskos@gmail.com*

*Tsolis Vasileios  
Department of Computer Science & Engineering   
University of Ioannina  
Ioannina, Greece  
vasilistsolis96@gmail.com*

*Abstract*—This electronic document is a “live” template and already defines the components of your paper [title, text, heads, etc.] in its style sheet. *\*CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract*. (*Abstract*)

Keywords—component, formatting, style, styling, insert (key words)

# Introduction

Social networks play a pivotal role in connecting individuals, and the task of friend recommendation within these networks is crucial for enhancing user experience. This project aims to address the challenge of friend recommendation using Graph Neural Networks (GNNs). The objective is to leverage GNNs to predict and rank potential new friends for users within social networks, particularly focusing on dataset from Facebook, Google+, and Twitter.

The primary challenge is to develop a friend recommendation system capable of providing accurate and effective suggestions within the given social networks. The anonymized nature of the data introduces complexities in understanding individual characteristics, and the large-scale network necessitates scalable algorithms.

The following sections cover…

# Related work

## Selecting a Template (Heading 2)

First, confirm that you have the correct template for your paper size. This template has been tailored for output on the A4 paper size. If you are using US letter-sized paper, please close this file and download the Microsoft Word, Letter file.

## Maintaining the Integrity of the Specifications

The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

# Methodology

The dataset was obtained through McAuley and Leskovec prior work on social circles in ego networks, currently hosted on the Stanford SNAP dataset library.

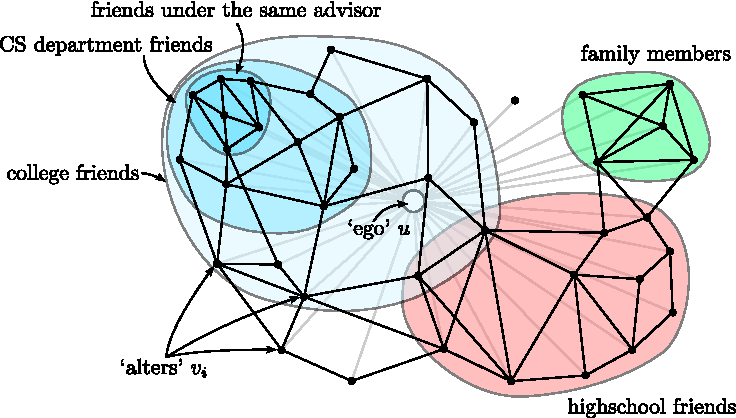


Figure 1 This ego-network contains users, features and social circles from Facebook. The task aims to adjust the features and user embeddings and calculate the ranking of friends.   
Source: http://i.stanford.edu/~julian/pdfs/nips2012.pdf

It includes user to user connections / interactions, denoted as edges (nodeId.edges), communities (nodeId.circles), features for each of the users represented as nodes (nodeId.feat), features for the ego user (nodeId.egofeat) and names of each of the feature dimensions (nodeId.featnames) from Facebook, Google+, and Twitter. Nodes, features (profiles), circles, and ego networks have been anonymized to ensure user privacy. The dataset anonymization is achieved by replacing original IDs and obscuring feature interpretations. Table 1 provides a brief description of the aforementioned datasets:

| Attributes | Data Set | | |
| --- | --- | --- | --- |
| Facebook | Google+ | Twitter |
| Nodes | 4039 | 107614 | 81306 |
| Edges | 88234 | 13673453 | 1768149 |
| Average clustering coefficient | 0.6055 | 0.4901 | 0.5653 |
| Diameter | 8 | 6 | 7 |
| 90-percentile effective | 4.7 | 3 | 4.5 |

Table 1. Dataset

## Data Pre-Processing

The data processing approach involved several key steps to enhance the dataset for subsequent analysis. The ego features were unified with the user features to create a cohesive representation of individual user characteristics. Then, the featnames were mapped to their respective feature names for each column, establishing a clear association between feature identifiers and their names. As part of the process, all networks were consolidated to create a final, integrated file. This file incorporates all features precisely aligned with their respective names.

The refined dataset serves as a valuable resource for model training and facilitates further in-depth analyzes within our science report. The processed data file is ready to contribute significantly to the exploration and creation of the Data Graph within the network.

Subsequently, a custom dataset class derived from PyTorch Geometric data processing library was defined to encapsulate the data and store them in memory of the appropriate processing device (CPU or GPU, if CUDA is available and enabled). This class object represents a graph compatible with PyTorch Geometric, suitable for training, validation and experimentation with graph neural networks.

## Graph Neural Networks Background

Graph Neural Networks (GNNs) are a category of neural networks specifically designed for processing and learning from graph-structured data. In contrast to conventional neural networks, which primarily operate on grid-structured or sequential data, GNNs are tailored to handle data represented as graphs composed of nodes (vertices) interconnected by edges (links). They have garnered significant attention due to their efficacy in modeling intricate relationships and dependencies inherent in data with irregular structures, such as social networks, biological networks, recommendation systems, and knowledge graphs. GNNs facilitate various tasks including node classification, link prediction, graph classification, and graph generation.

At their core, GNNs employ an iterative approach to update node representations by aggregating information from neighboring nodes within the graph. This iterative process enables GNNs to capture the local graph structure and propagate information across the entire graph. A multitude of architectural designs and techniques have been proposed to realize this concept, including Graph Convolutional Networks (GCNs), Graph Attention Networks (GATs), GraphSAGE, Graph Isomorphism Networks (GINs), among others.

Our approach on the friendship recommendation model is split into a GCN model to learn node embeddings, and a friend ranking model to utilize node embeddings knowledge for friendship suggestion based on higher relevancy ranking score. Finally, the GCN model effectiveness was evaluated against a baseline TruncatedSVD model for the same task of learning node embeddings.

## Graph Convolutional Network Model

In the context of modeling the learning process for node embeddings, a graphical model of a neighborhood neural network was developed, focusing on the analysis of social networks through the Facebook platform, as derived from the Facebook dataset. The model is based on a Graph Convolutional Network (GCN) and is designed to learn information about the users and calculate features for each node based on the neighbor nodes. The fundamental concept is to perform convolution on the k-neighbor node features while maintaining the structural connections between nodes. First, the model receives data from the Facebook dataset, which includes relationship connections between users and user features. The input is represented as a graph, with users as nodes and connections as edges.

Incorporated within the model are two hidden layers employing graphical convolutions, with ReLU activations and a dropout layer in between. This allows extracting important features from the graph and user features. Additionally, a Stochastic Gradient Descent optimizer is utilized as well as a StepLR scheduler to control the learning rate decay. At the final output level, the loss is minimized from the calculations of positive and negative scores against the accuracy of the predictions of relationships between users.

*a**b* 

The additional baseline model used as reference was designed based on the TruncatedSVD model. TruncatedSVD is based on dimensionality reduction while preserving the most important information in the data. Truncated SVD is applied in collaborative filtering-based recommendation systems to handle sparse user-item interaction matrices efficiently. It helps in identifying latent factors or features that represent user preferences and item characteristics, enabling more accurate recommendations, thus, making it a suitable comparison model to the GCN implementation. It features the same SGD optimizer and StepLR scheduler as the GCN model.

## Friend Ranking Module

After training the model to obtain embedded representations of the nodes, the next objective is a prediction model for estimating the "likelihood" of two nodes connecting, by calculating the dot product between the two nodes embeddings.

The Friend Ranking Module utilizes as input the edge index matrices, and the node features h, as learnt from the GCN/TruncatedSVD model. For each edge pairing, the module maps the features with the source node and the destination node.

Subsequently, it calculates the dot product between the two embeddings pairings. The dot product provides a score measurement indicating the "agreement" between these features. In the context of graph representation, this is interpreted as an indication of the probability of a connection between the two nodes, denoted as an edge between the two nodes in the graph.

# experiments

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.

## Dataset

In the context of data splitting, the approach taken involves the separation of the social network graph into a training set. This separation is determined by a non-random percentage (specifically set at 30%), specifying the number of edges to be used for creating the training set. Initially, the calculation of the number of edges to retain for training is based on the given percentage. Subsequently, the edges are randomly shuffled, and a subset is selected to establish the training set.

Following this, the positive edges of the training set are computed, and negative edges are generated using the negative sampling function from PyTorch Geometric. This separation entails the creation of a subgraph that exclusively includes positive edges, contributing to a focused training environment for the model.

Note that the separation of the graph into a training set and a test set is non-random. This is important to ensure reliable performance metrics of the model, as some metrics, such as: Hits@K, NDCG@K, MRR, require a specific separation for proper evaluation. Therefore, non-random separation ensures consistency of performance metrics when training and testing the model across iterations.

## Evaluation Metrics

This section of the report describes the evaluation process of our recommendation system's performance through prediction and ranking metrics. These metrics play a crucial role in measuring the effectiveness of our model in predicting the top items that are likely to attract users' interest. This evaluation is important, especially in applications such as social network recommendation systems, where the precise prediction of user interests is crucial. In conclusion, the evaluation metrics presented in this section are a crucial component for validating the effectiveness of our recommendation system. They provide valuable insights into the system's predictive capabilities, facilitating ongoing improvements and optimizations for enhanced user satisfaction.

The metrics utilized for evaluating the recommendation model encompass the following key measures:

### Accuracy Based Metric: the model's performance is evaluated on the test set using the Area Under the Curve (AUC) metric.

### Ranking Based Metrics: The approach used involves utilizing Graph Convolutional Networks (GCNs) to model the relationships and structures within the social network. GCNs are specialized for learning from graph-structured data, making them ideal for friend recommendation tasks. To measure the performance of the friend recommendation system, the following metrics will be used:

#### Hits@K: Measures the proportion of correct recommendations within the top K recommendations.

#### NDCG@K (Normalized Discounted Cumulative Gain): Evaluates the ranking quality of the recommendations at position K.

#### MRR (Mean Reciprocal Rank): Assesses the effectiveness of the recommendation by considering the rank of the first correct recommendation.

## Experiment Results

The presented table showcases the performance metrics for two different models, GCN (Graph Convolutional Network) and tSVD (Truncated Singular Value Decomposition), across various evaluation criteria. Here are some observations and insights based on the provided results:

| Model | AUC | Hits | NDCG | MRR | Ranking |
| --- | --- | --- | --- | --- | --- |
| GCN | **0.9596** | **0.0096** | 0.0273 | 0.0212 | Top 5 |
| 0.0095 | **0.0868** | 0.036 | Top 50 |
| tSVD | 0.8556 | 0.0043 | 0.0126 | 0.0097 | Top 5 |
| 0.0048 | 0.0448 | 0.017 | Top 50 |

1. Example of a figure caption. (*figure caption*)

#### AUC (Area Under the Curve): GCN achieves a notably higher AUC score (0.9596) compared to tSVD (0.8556). This suggests that GCN demonstrates superior discriminatory power in distinguishing positive and negative instances.

#### Top-5: For Hits@5, NDCG@5, and MRR at Top-5, GCN outperforms tSVD across all three metrics. GCN's higher values indicate better performance in accurately recommending relevant items within the top-5 recommendations.

#### Top-50: Similar to the Top-5 metrics, GCN also outperforms tSVD for Hits@50, NDCG@50, and MRR at Top-50. This implies that GCN maintains its superiority in recommending relevant items across a larger set of recommendations (Top-50).

In summary, based on the provided metrics, GCN appears to be a more effective model for friend recommendation in the given social network graph compared to tSVD. The higher AUC score and consistently better performance across various top-K recommendation metrics indicate the superior predictive and ranking capabilities of GCN in this context.

# Discussions

The preferred spelling of the word “acknowledgment” in America is without an “e” after the “g”. Avoid the stilted expression “one of us (R. B. G.) thanks ...”. Instead, try “R. B. G. thanks...”. Put sponsor acknowledgments in the unnumbered footnote on the first page.

# Conclusions

The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use

##### References

The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use “Ref. [3]” or “reference [3]” except at the beginning of a sentence: “Reference [3] was the first ...”

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

1. G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*
2. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
3. I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
4. K. Elissa, “Title of paper if known,” unpublished.
5. R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.
6. Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
7. M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.