Interaction Graph-Based Community Recommendation on Subreddits

Scott Sutherland Ryan Don Felicia Chan Pravar Bhandari

University of California, San Diego

Background and Goal

- Reddit is a social media platform where users can submit content on communities called subreddits that are based on specific topics and interests. Within these subreddits, users can engage in discussions with each other in "comment chains" by replying to comments. As such, users can be linked to other users and subreddits via interaction.
- Our goal: leverage graph-based representations of user interaction activity in order to make subreddit recommendations for those users.
- To accomplish this, we use TigerGraph, a fast and scalable graph database and machine learning suite of tools.

Why Graph?

Graph theory has existed for decades, but advancements in computing power and introduction of large-scale graph databases like TigerGraph have allowed applications of graph algorithms to a wide range of domains. Traditional data methods, like representing data tabularly, are less effective in representing complex relationships; rows are observations, columns are variables. This can be limiting for data that has many associated variables and complex relationships. We believe that user-subreddit/user-user relationships are better represented as graphs, which makes representation flexible.

Graph Schema and Data

Within TigerGraph's Graphstudio user interface, we can create a blueprint of our graph database called a graph schema, which defines our types of nodes, edges, and attributes. Schemas provide structure for the data and enable data integrity and efficient querying. Our graph schema consists of 2 types of nodes and 2 types of edges, shown below in Figure 1.

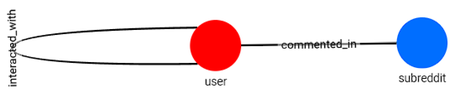

Figure 1. Schema of the Reddit User-Interaction Graph.

- Each node class represents a user or a subreddit, and each type holds their own unique attributes. Attributes are used in feature representation and learning.
- Each edge class represents a relationship between nodes. They have unique attributes as well.
- In total, our graph contains 3,800,000 total vertices and 8,879,493 edges. The data was obtained from an archive of Reddit data pushshift.io from December 2010.

The data is heavily biased towards a small set of subreddits. That is, most user interactions on Reddit at that time took place in a few large subreddits. Chains of user interactions are also biased in that the distribution of their lengths are right skewed.

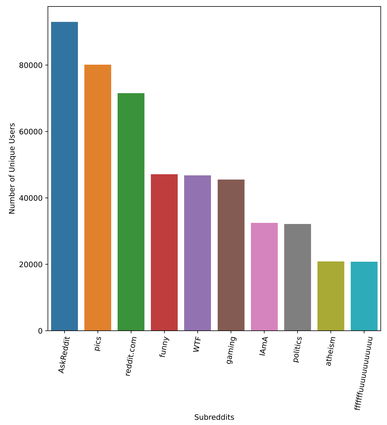

Figure 2. 10 most common subreddits

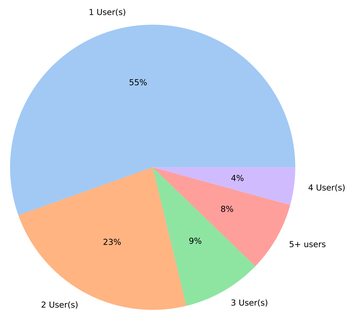

Figure 3. Distribution of Unique Users in Chains

Baselines and Methods

Non-Graph Based Algorithms

Our baseline models are not graph-based, which allow us to compare standard recommendation performance to graph-based recommendation performance. The baselines consist of algorithms such as simple K-nearest neighbors algorithms with cosine similarity (1), Jaccard similarity (2), and a simple popularity recommender model, which recommends the most popular subreddits from the dataset that a user is not already subscribed to.

$$cos(\theta) = \frac{A \cdot B}{||A||||B||} \tag{1}$$

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \tag{2}$$

Graph-Based Algorithms

We then calculate several graph-based metrics and run algorithms in TigerGraph:

Centrality Algorithms: finds the most important nodes in the graph based on connections with other nodes

- **Degree Centrality:** number of edges a user or comment node has.
- **PageRank:** counts number and quality of links to a comment to determine the importance of the comment.

Community Algorithms: group together nodes or edges that satisfy a rule for being connected to one another

- **Louvain:** optimizes modularity of graph by merging nodes into communities based on connectivity patterns.
- **Label Propogation:** quickly propogates labels through graph and forms communities.

Network Statistics Model

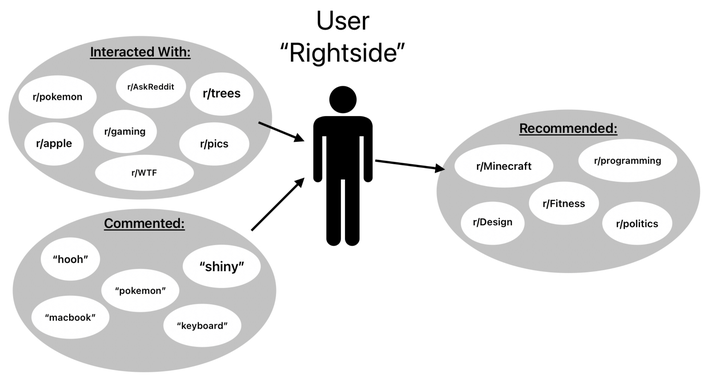

Figure 4. A visual example of recommendations made by the Network Statistics model for user "Rightside"

Our final model is the Network Statistics Model. This model is a combination of some hand-picked algorithms listed in the previous section with fine-tuned hyperparameters, to ensure an optimized and comprehensive final model. The network statistics model calculates a PageRank, Degree, Louvain, and Label Propagation score for each user node. In addition, the top 25 most influential keywords from each User's comments are found using a TFIDF vectorizer, and then a pre-trained word2vec model with an embedding size of 50 is used as additional embeddings, so 25 x 50 is the total number of embeddings for each User (1250 embeddings). In addition to these keyword embeddings, each user then has an embedding of these 4 scores, which respectively represent:

- PageRank: How influential a user is
- Degree: How active a user is based on number of interactions
- Louvain: What community a user belongs to based on the Louvain algorithm
- Label Propagation: What community a user belongs to based on the Label Propagation Algorithm

Using these embeddings, K-Nearest Neighbors is utilized to calculate the two most similar users to the input user. We then recommend subreddits that the two neighboring users are active in that the input user isn't already apart off.

Results

To evaluate our models, we use Precision@k as our evaluation metric. Precision@k is the proportion of the recommended items in the top-k set of recommendations that are relevant.

$$\text{Precision@k} = \frac{\text{\# of recommended items @k that are relevant}}{\text{\# of recommended items @k}} \tag{3}$$

Algorithm	Graph-Based?	@1	@3	@5	@10	@25
Popularity Recommender	No	0.1000	0.0944	0.0729	0.0475	0.0251
Jaccard Similarity	No	0.0250	0.0201	0.0195	0.0204	0.0205
Cosine Similarity KNN	No	0.0604	0.0423	0.0372	0.0387	0.0402
Network Statistics Recommender	Yes	0.0000	0.0207	0.0338	0.0712	0.0755

Table 1. Precision@k results for all models (scores are averages across users)

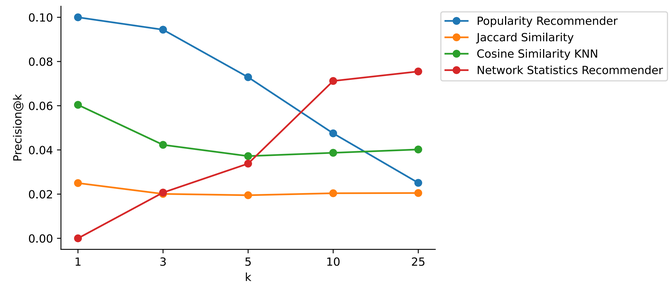

Figure 5. Precision@k for different k values on all models.

We calculate these values by pulling data from 1 year in the future from the training data and examining all the Subreddits the training users interact in that they did not interact in before. Precision quantifies how well the recommendations we make match those true interactions.

Conclusion

When @k is 10/25, our final model outperforms standard recommendation techniques. However, our model fails to surpass standard models when @k is 1/3/5. We believe that this under performance may be due to the bias of users only interacting in the most popular subreddits at the time and not exploring new and upcoming subreddits based on their personal interests. Our model was trained on data from 2010, when Reddit was relatively new and most user interactions were happening in the most popular subreddits. Our project showcases the potential of graph-based recommendations, which are a relatively new concept in comparison to decades-old methods like k-nearest neighbors. However, the ability for graphs to handle complex relationships, as well as increased efficiency in data storage and computation for graphs creates a huge advantage, and the increased metrics from our final model definitely demonstrate the effectiveness of interaction graph-based recommendations for Reddit.

Future Work and Next Steps

Future applications include graph-based and interaction-based recommendation on other social media networks such as Twitter or Facebook, where potential communities or topics can be recommended instead. For expansions done to this project in particular, we could use a more recent dataset in order to overcome the bias of popular subreddits. We hypothesize that our model will have improved results when trained on new data. We used a smaller subset of data from earlier years as otherwise the graph sizes would be too large. We could explore further algorithms within TigerGraph and keep tuning the hyperparameters for the algorithms to ensure the optimal precision@k and recall@k.

Accompanying Work

Github

Website