Project Specification

Support

Project related email: hanchen.wang@uts.edu.au

Hanchen Wang (Instructor)

Xubo Wang (Instructor)

Han Chen (Tutor)

Yin Chen (Tutor)

Team/group

The students are required to work in groups of 2-4 people.

Please

- Start to explore the possible topics and establish/join a group **as soon as possible**.
- Nominate a group leader, who will send the instructor an email about the group members as well as the proposed project topic. (by Week 5)
- In the lab, we will leave around 30-40 minutes for group discussions (staff may be invited to join)

Important Date

- **Friday, Week 5:** The groups should be formed. The group leader should send an email to the following email account (hanchen.wang@uts.edu.au) and a group id will be allocated. The email should also include the proposed topic. If the topic is not the suggested topics in the specification, you need the approval from instructors/tutors beforehand.

- **Friday, Week 12:** Group project due. (Submit via Turnitin)

Project Consultation

- Encourage group discussions in the Lab time, instructors or tutors will provide support.
- Make appointments with Tutor/Instructors

Project topics

In this group project, you need to apply network analysis tools and algorithms to solve real-life problems. To allow you to follow your own interests, the project does not restrict to a specific topic. It could be any topic that is relevant to social and information networks, such as

- **Network analysis and visualization**. Analyze an interesting network from different aspects, such as degree distribution, network centrality, community detection, network evolution and graph visualization.
- **Algorithm**. Implementation of algorithms for processing graphs that solve real-life problems.
- **Application**. Develop a novel application to offer a new function in real-world problems based on network analysis.

To help students with the project topic, we provide some topic candidates that the students are free to pick from:

- **Selected topics**. We offer 7 selected topics, which include social network analysis using LLMs, friend recommendation, movie recommendation, POI recommendation, search engine prototype and cycle detection problem. The details of each selected project will be presented in the Appendix.

- **Some other possible topics**.
  1. Related Tasks from Online Competitions (e.g., data from Kaggle https://www.kaggle.com/datasets)

  2. Network Visualization and Analysis. Here are some examples from Stanford course: Analysis of the YouTube Channel Recommendation Network, Network analysis on Startups and VCs and Analysing the development of Wikipedia. There is an interesting blog which analyses the characters in the *Game of the thrones: Network of Thrones*.

  3. Implementation and possibly improvement of an existing research paper related to networks. You may find interesting research papers from the following top conferences such as WWW, KDD, WSDM, IJCAI, AAAI, VLDB, and SIGMOD.

  4. Topics related to graph node classification (see https://hpi.de/fileadmin/user_upload/fachgebiete/mueller/courses/graphmining/GraphMining-06-NodeClassification.pdf) or graph embedding (See https://cs.stanford.edu/~jure/pubs/graphrepresentation-ieee17.pdf).

  5. **Your own topics.** It must involve some practical work with Network Analysis (Networks other than Social and Information networks should be fine). **Note:** you need the approval of the instructors not later than Week 5 regarding your project topic.

Evaluation Criteria

Students should have a publishable or near-publishable report for their projects. The report may include abstract, introduction, related work, methods (e.g., algorithms or network metrics used), dataset, results (e.g., experimental report, analysis, and visualization), conclusion and references.

The project will be evaluated based on:

- The technical quality of the work: does the technical material make sense? Are the things tried reasonable? Are the proposed algorithms or applications clever and interesting? Do the authors convey novel insight about the problem and/or algorithms?
- Significance: Did the authors choose an interesting or a "real" problem to work on, or only a small "toy" problem? Is this work likely to be useful and/or have impact?
- The novelty of the work, and the clarity of the write-up
- Presentation of the results. Well-formatted, well-organised, spell-checked and grammar-checked documents.
- **Plagiarism check**
- **Members in the same group will receive equal marks**. If some of the group feel that other members are not contributing, the instructor should be informed and a group meeting should be held to produce a solution **4 weeks before the deadline**. **No complaints about group operation will be considered after the project has been handed in**.

==== More details ===
1. Technical Quality (**40%**)
Are the results technically sound?
Are there obvious flaws in the conceptual approach?
Are claims well-supported by theoretical analysis or experimental results?
Are the experiments well thought out and convincing?
Will it be possible for other researchers to replicate these results?
Is the evaluation appropriate? Did the authors clearly assess both the strengths and weaknesses of their approach?

2.Quality of writing (**30%**)
Is the paper clearly written?
Is there a good use of examples and figures?
Is it well organized? Are there problems with style and grammar?
Are there issues with typos, formatting, references, etc.?

3. Novelty and Significance **(30%)**
We will recognise and reward papers that propose genuinely new ideas. Novel combinations, adaptations or extensions of existing ideas are also valuable.
Is this a significant advance in the state of the art?
Is this a paper that people are likely to read and cite?
Does the paper address an important problem?
Is it a paper that is likely to have a lasting impact?

Technical Report

- How to write a technical report. https://www.uts.edu.au/current-students/support/helps/self-help-resources/academic-writing/report-writing
- Subject 32144: Technology Research Preparation
- Approximately 10-15 pages for single column, single space format.

Resources of some real-life network data

Gephi	https://github.com/gephi/gephi/wiki/Datasets
SNAP	http://snap.stanford.edu/data/
Mark Newman	http://www-personal.umich.edu/~mejn/netdata/

You may find social and information network data from other resources, or crawl/generate the network data by yourselves.

Appendix

Selected Topic 1: Social Network Analysis using Large Language Models

Description

Real-world social network frequently encompasses rich textual information, such as comments and reviews in social networks, which can be naturally represented as text-attributed graphs. Recent advancements in large language models (LLMs) have demonstrated their remarkable capabilities in textual understanding and generation, presenting new opportunities for enhancing the analysis of textual attributes within network data.

This project aims to investigate the power of LLMs in social network analysis. There are two potential tasks. (1) Analysis of Citation networks. The citation networks, where the nodes are the papers, and the edges are the reference links, usually associate with the text information within the paper. In this task, we aim to improve the performance on several traditional network analysis tasks, such as node classification (e.g., categorizing papers based on content and topological information) and link prediction (e.g., inferring potential reference relationships) by leveraging the text information. (2) Fraudulent review detection on e-commerce networks. Fraudulent activities—such as fake reviews, deceptive account interactions, and abnormal transaction behaviors—often leave discernible traces in both textual content (e.g., misleading product descriptions) and network structures (e.g., sudden dense connectivity among suspicious accounts). This task seeks to develop more effective detection methods by using LLMs compared to traditional approaches.

Datasets

This project provides the Cora and Amazon-Review datasets.

Cora Dataset

The Cora dataset is a widely used citation network dataset for research in graph machine learning. It consists of 2,708 papers with 5,429 citation relationships. These papers are categorized into seven classes, such as Neural Networks, Reinforcement Learning, and Probabilistic Methods. The dataset provides both pre-computed features and the original text content (titles and abstracts) for each paper.

The details:

A. Original Features and Citation Data:

cora.content contains descriptions of the papers in the following format:

<paper_id> <word_attributes> <class_label>

The first entry in each line contains the unique string ID of the paper followed by binary values indicating whether each word in the vocabulary is present (indicated by 1) or absent (indicated by 0) in the paper. Finally, the last entry in the line contains the class label of the paper.

cora.cites contains the citation graph of the corpus. Each line describes a link in the following format:

<ID of cited paper> <ID of citing paper>

Citing Paper ID | Cited Paper ID

Each line contains two paper IDs. The first entry is the ID of the paper being cited and the second ID stands for the paper which contains the citation.

B. Text Data:

Mapping file is located at `cora_orig/mccallum/cora/papers`, this file provides a mapping between each paper ID and its corresponding text filename.

The actual text for each paper (including its title and abstract) is stored in the directory `cora_orig/mccallum/cora/extractions/`. Each text file contains lines starting with "Title:" and "Abstract:"; the title and abstract can be extracted and concatenated (using a newline separator) to form the text data for the paper.

This dataset is ideal for tasks such as node classification and link prediction in graph neural network research, leveraging both the structured citation information and the rich textual content of the publications.

Data Download:
https://drive.google.com/file/d/1hxE0OPR7VLEHesr48WisynuoNMhXJbpl/view

Amazon-Review Dataset:

In the Amazon-Review dataset, users receiving more than 80% helpful votes can be identified as benign entities, while those with fewer than 20% helpful votes can be identified as fraudulent entities. The objective is to conduct a fraudulent user detection task on the Amazon-Review dataset, which is a binary classification task.

Data Download:

https://snap.stanford.edu/data/amazon/productGraph/categoryFiles/reviews_Musical_Instruments.json.gz

Evaluations

The evaluation method is detailed in the papers "Harnessing Explanations: LLM-to-LM Interpreter for Enhanced Text-Attributed Graph Representation Learning" and "Group-Based Fraud Detection Network on E-Commerce Platforms," as cited in the references.

References

1. He, Xiaoxin, et al. "Harnessing explanations: Llm-to-lm interpreter for enhanced text-attributed graph representation learning." arXiv preprint arXiv:2305.19523 (2023).
2. Yu, Jianke, et al. "Group-based fraud detection network on e-commerce platforms." Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2023.
3. Dou, Yingtong, et al. "Enhancing graph neural network-based fraud detectors against camouflaged fraudsters." Proceedings of the 29th ACM international conference on information & knowledge management. 2020.

Selected Topic 2: Friend Recommendation on Social Networks

Description

Social networks are usually highly dynamic; they grow and change quickly over time through the addition of new edges and the removal of old ones. Identifying the mechanisms by which they evolve over time is a fundamental question. In this project, we focus on the link prediction problem on evolving social networks, which aims to predict the future links between nodes by utilizing node features and network features.

Let's take the Facebook "People You May Know" feature as an example. Facebook periodically recommends new people to users such that users can make more new friends. You may wonder how Facebook recommends friends to you. Are these people just randomly selected, or do they have many common places with you? Actually, Facebook follows the simple intuition that "similar" users are more likely to get connected in real life than the "dissimilar" ones, and thus should be recommended to each other. Following this idea, Facebook recommendation is achieved by mining the implicit online relationships between users, which might finally lead to offline friendship in the future. For example, if two people have lots of common friends, live in the same city or go to the same university, they are very likely to be friends in the future. In this project, you need to investigate various features that may contribute to the connection between two people by exploring network structures.

Datasets

A dataset will be provided for this project. We will provide two graph snapshots. The old snapshot is used for algorithm training while the new one is used for evaluation.

Astro Physics collaboration network dataset

Arxiv ASTRO-PH (Astro Physics) collaboration network is from the e-print arXiv and covers scientific collaborations between authors of papers submitted to Astro Physics category. If an author i co-authored a paper with author j, the graph contains an undirected edge from i to j. If the paper is co-authored by k authors this generates a completely connected (sub)graph on k nodes. The dataset covers papers in the period from January 1993 to April 2003 (124 months). It begins within a few months of the inception of the arXiv and thus represents essentially the complete history of its ASTRO-PH section.

The data file contains 18772 nodes (i.e., authors) and 198110 edges (i.e., collaborations). Each line of the data file contains two values representing an edge. The first value is the fromNodeId, and the second value is the toNodeId.

Data Download:
https://www.dropbox.com/sh/3blf33x0mtfb3f4/AAC1HGSCgJwD7yO9kGccS-dma?dl=0

Evaluations

The evaluation method can be found in the paper "The link prediction problem for social networks" as shown in the reference. Basically, it counts the intersection of the predicted friends and true friends. The higher the counter is, the better the prediction result is.

References

4. Liben Nowell, David, and Jon Kleinberg. "The link prediction problem for social networks." Journal of the Association for Information Science and Technology 58.7 (2007): 1019-1031.
5. Backstrom, Lars, and Jure Leskovec. "Supervised random walks: predicting and recommending links in social networks." Proceedings of the fourth ACM international conference on Web search and data mining. ACM, 2011.

Selected Topic 3： Movie Recommendation

Description

Recommendation systems are used to predict the "rating" or "preference" that a user would give to an item. Recommender systems have become increasingly popular in recent years and are utilized in a variety of areas such as movie recommendation. In this project, you will be given a MovieLens dataset which includes the information of movies and users and the rating a user gives to a movie. Based on the dataset, you can build a simple recommendation system to predict which movies a user may like and predict the rate the user would give to a movie.

Datasets

MovieLens dataset will be provided for this project. The data was collected through the MovieLens web site (movielens.umn.edu) during the seven-month period from September 19th, 1997 through April 22nd, 1998. The dataset consists of 100,000 ratings (1-5) from 943 users on 1682 movies. This data has been cleaned up – users who had less than 20 ratings or did not have complete demographic information were removed from this data set.

The details:

u.data: The full u data set, 100000 ratings by 943 users on 1682 items. Each user has rated at least 20 movies. Users and items are numbered consecutively from 1. The data is randomly ordered. This is a tab separated list of user id | item id | rating | timestamp. The time stamps are unix seconds since 1/1/1970 UTC

u.info: The number of users, items, and ratings in the u data set.

u.item: Information about the items (movies); this is a tab separated list of movie id | movie title | release date | video release date |IMDb URL | unknown | Action | Adventure | Animation |Children's | Comedy | Crime | Documentary | Drama | Fantasy |Film-Noir | Horror | Musical | Mystery | Romance | Sci-Fi |Thriller | War | Western |

The last 19 fields are the genres, a 1 indicates the movie is of that genre, a 0 indicates it is not; movies can be in several genres at once.

The movie ids are the ones used in the *u.data* data set.

u.genre: a list of all the genres.

u.user: Demographic information about the users; this is a tab separated list of

user id | age | gender | occupation | zip code

The user ids are the ones used in the *u.data* data set.

u.occupation: a list of the occupations

Data download: https://www.dropbox.com/s/7a1rpq684c33nca/ml-20m.zip?dl=0

https://www.dropbox.com/s/ip7x5v26a5kvixg/ml-100k.zip?dl=0

For more details: https://grouplens.org/datasets/movielens/

Evaluation

The data set has 80%/20% split of training data and test data. You can just use the test data to evaluate your result or you can split the data by yourself using the timestamp to evaluate your recommender system.

You can evaluate it by searching for the low prediction error (RMSD) and high recall coverage. For details you can click the link in references. In your report you need to give the RMSD and recall of your recommender system.

RMSD

The root-mean-square deviation (RMSD) is a frequently used measure of the differences between values (sample and population values) predicted by a model or an estimator and the values actually observed.

The RMSD of predicted values for times t of a regression's dependent variable is computed for n different predictions as the square root of the mean of the squares of the deviations:

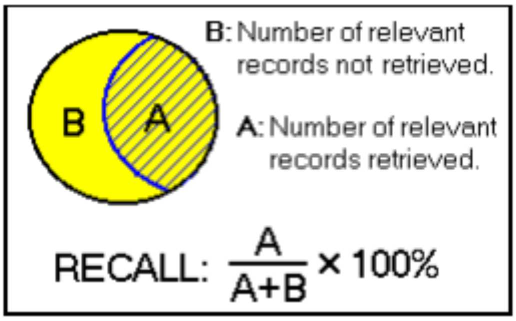
$$\operatorname{RMSD} = \sqrt{\frac{\sum_{t=1}^{n} (\hat{y}_t - y_t)^2}{n}}.$$

Recall:

Recall (R) is defined as the number of true positives (T_p) over the number of true positives plus the number of false negatives (F_n).

$$R = \frac{T_p}{T_p + F_n}$$

e.g.


B: Number of relevant records not retrieved.
A: Number of relevant records retrieved.
B
A
RECALL: A / A+B × 100%

References

1. BN Miller, I Albert, SK Lam, JA Konstan. MovieLens unplugged: experiences with an occasionally connected recommender system. IUI 2013.
2. Jure Leskovec, Anand Rajaraman, Jeff Ullman, Mining of Massive Datasets Chapter 9, Cambridge University Press, second edition, 2014 (can be downloaded via http://www.mmds.org)
3. Evaluating Recommender Systems. (https://medium.com/recombee-blog/evaluating-recommender-systems-choosing-the-best-one-for-your-business-c688ab781a35)

Selected Topic 4: POI Recommendation

Description

Point-of-interest (POI) recommendation has become a major issue with the rapid emergence of location-based social networks (LBSNs). Unlike traditional recommendation approaches, the LBSNs application domain comes with significant geographical and temporal dimensions.

In this project, you can use the data from Yelp and Foursquare. Let's take the Yelp data as example. Yelp is one of the most famous LBSNs, and you will be provided the information of the shops and users, e.g. the type of the business, the location of the business, the rate a user gives to a business and the check-in information of a user. Based on this information, you can find what type of business a specified user likes. For example, one user often goes to Vietnamese restaurant and always rate highly for them, we might have the conclusion that the user likes to eat Vietnamese food. With the location information, you can recommend some nearby Vietnamese restaurants for that user.

Datasets

You will be provided with two datasets for this project. One is from the Yelp Dataset Competition, and the other is from the Foursquare which is also a LBSN. These datasets both contain the common information of business and users and the location information. You are also encouraged to find a dataset which you are interested in.

The provided data has following information.

The business information, the users' information, the review details, the check-in information, and tip information, you can get the dataset information in

https://www.dropbox.com/s/e80rv0wwm800mvq/yelp_dataset_challenge_round9.tar?dl=0

The Foursquare dataset download:

https://www.dropbox.com/sh/tf9wvlkk6ene6ry/AACOMB5QeR8BRziOh2QVPzdXa?dl=0

Evaluation

In this project you might need to predict the rate a user might give to a business and the probability a user check-in in a place. For the Yelp dataset you will need to split some data as test data. You can split it using date and use the newer dates as test data.

You can evaluate your recommender system by searching for the low prediction error (RMSD) and high recall coverage. For details you can click the link in references. In your report you need to give the RMSD and recall of your recommender system.

RMSE

The root-mean-square deviation (RMSD) is a frequently used measure of the differences between values (sample and population values) predicted by a model, or an estimator and the values actually observed.

The RMSD of predicted values for times t of a regression's dependent variable is computed for n different predictions as the square root of the mean of the squares of the deviations:

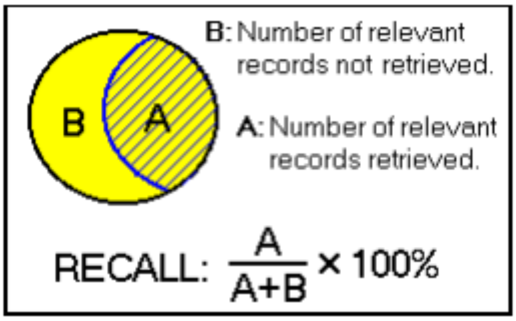
$$\mathrm{RMSD} = \sqrt{\frac{\sum_{t=1}^{n}(\hat{y}_t - y_t)^2}{n}}.$$

Recall:

Recall (R) is defined as the number of true positives (T_p) over the number of true positives plus the number of false negatives (F_n).

$$R = \frac{T_p}{T_p+F_n}$$

e.g.


B
A
B: Number of relevant records not retrieved.
A: Number of relevant records retrieved.
RECALL: $\frac{A}{A+B} \times 100\%$

References

1. Bin Liu, and Hui Xiong. Point-of-Interest Recommendation in Location Based Social Networks with Topic and Location Awareness. Proceedings of the 2013 SIAM International.
2. M Xie, H Yin, H Wang, F Xu, W Chen, and S Wang, Learning Graph-based POI Embedding for Location-based Recommendation. CIKM 16
3. Evaluating Recommender Systems. (https://medium.com/recombee-blog/evaluating-recommender-systems-choosing-the-best-one-for-your-business-c688ab781a35)

Selected Topic 5: A Simple Google Search Prototype

Description

The amount of information on the web is growing rapidly every day. Thanks to search engines like Google, users can easily find the information they want through a simple click. Most of the search engines usually return pages of results according to their relevance to the user query. One of the main factors that contributed to Google's initial success is a ranking model called PageRank. PageRank makes use of the link structure of the web to calculate a quality ranking for each web page. In this project, you need to implement a simple Google search prototype using PageRank. Specifically, given a query, such as "uts Australia", your search engine should be able to return a set of pages that contain both "uts" and "Australia", and the most relevant pages should be put on the top. The proposed ranking model should at least use the PageRank metric, and the students are encouraged to investigate other features that could be used to improve the ranking.

Datasets

We will provide a web dataset, named WebSpam. WebSpam contains about 200K web pages as well as their link structures and their raw html contents.

Specifically, we will provide three files:

1. *url_graph_file*: each node is represented by a unique URL. In this file, every unique URL in the corpus is treated as a node in the web graph, and every unique link to another URL in the corpus is stored as an edge in the web graph.
2. *url_id_mapping*: maps ids to real URLs.
3. *Webspam2011_htl.tgz*: the raw html files. You need to match the real URL to the raw html content in order to get the mapping between real URL and its html.

Data Download: https://www.cc.gatech.edu/projects/doi/WebbSpamCorpus.html

Evaluations

The evaluation will be based on whether the implemented system can achieve its desired functions. There is no ground truth in this project.

References

1. Page, Lawrence, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: Bringing order to the web. Stanford InfoLab, 1999.
2. Brin, Sergey, and Lawrence Page. "The anatomy of a large-scale hypertextual web search engine." Computer networks and ISDN systems 30, no. 1 (1998): 107-117.

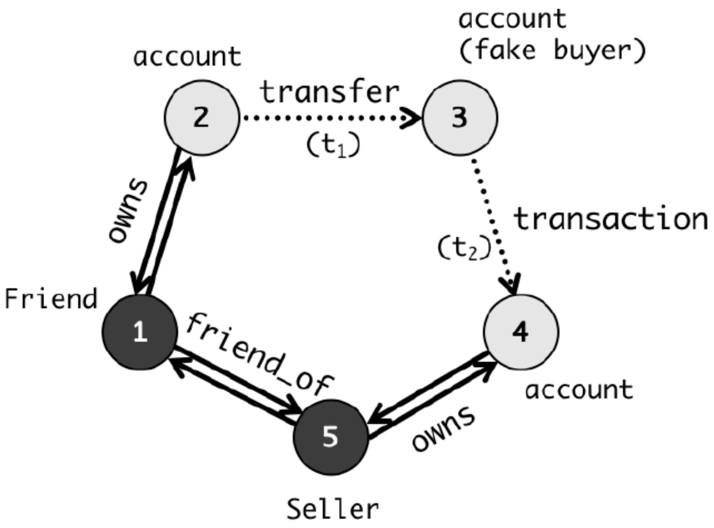
Selected Topic 6： Cycle Detection in Dynamic Graphs

Description

Data generated by an increasing number of applications is being modelled as graphs. This is because the graph structure can encode complex relationships among entities which can appear in social networks, e-commerce transactions, electronic payments, etc. Sophisticated analytics over such graphs provides valuable insights into the underlying dataset and interactions among different entities.

As one of the analytical approaches, cycle detection is the algorithmic problem of finding cycles in a graph (including a set of vertices and edges). In this project, you will be given a dynamic graph dataset including nodes, static edges, and dynamic edges. Your goal is to identify the newly generated cycles and return them for a set of continuous queries respectively for each incoming edge of the dynamic graph. Each query can ask for cycles satisfying some predefined constraints, such as length constraints.

The following is a simple example of cycle detection among buyers and sellers in an e-commerce platform. We denote individual users (buyers or sellers) and their accounts as vertices in the graph. There are two types of edges. One type of static edges (in solid lines) models the association of accounts to users and the relationships among different users, while online transactions including payment activities are denoted as dynamic edges (in dotted lines) for the corresponding vertices. In order to increase the popularity of a merchandise so as to improve future sales, fake transactions are placed to artificially bump up the number of past transactions.



In this example, this is achieved through a third-party account (vertex 3) from which a normal order is placed and its payment (edge 3 → 4) is completed at time t_2. However, the merchandise is never shipped by the seller (vertex 5) and the money used for the payment by the fake buyer (vertex 3) was previously transferred to him/her via the seller's friend (vertex 1) at time t_1 using his or her own account (vertex 2). The entire process is rather complicated involving multiple entities. Interestingly, it generates a cycle (1 → 2 → 3 → 4 → 5 → 1) in the graph, which can be returned as strong indication that a fraud may exist.

Datasets

P2p-Gnutella(04-09) series dataset will be provided for this project. The data was collected through the Gnutella peer to peer network from August 2002. The dataset consists of about 10,000 nodes and 20,000-40,000 directed edges. You can split the whole edges set into 3/4 static edges and 1/4 dynamic edges randomly to construct the query processing. You can also use other dataset including directed edges.

Dataset format:

FromNodeId ToNodeId

Dataset link: http://snap.stanford.edu/data/index.html#amazon

Evaluation

The evaluation will be based on query response times which means how long it takes for the system to return correct query results. The faster correct results return, the better the performance of the system is.

References

3. Qiu, X. , Cen, W. , Qian, Z. , Peng, Y. , & Zhang, Y. . (2018). Real-time constrained cycle detection in large dynamic graphs. Proceedings of the VLDB Endowment, 11(12), 1876-1888.

Selected Topic 7： Maximum Biclique Search at Billion Scale

Description

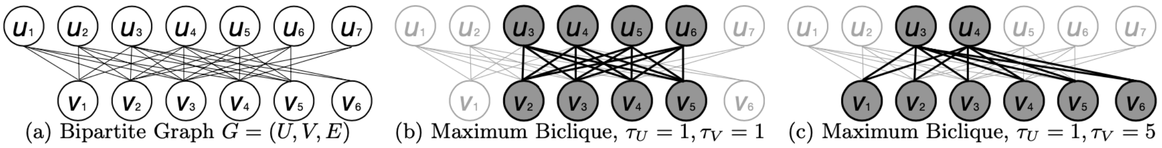
A bipartite graph is denoted by G = (U, V, E) where U(G) and V(G) denote the two disjoint vertex sets and E(G) ∈ U × V denotes the edge set. A subgraph C is a *biclique* if it is a complete bipartite subgraph of G that for every pair u ∈ U(C) and v ∈ V (C), we have (u, v) ∈ E(C).

Bipartite graph is a popular data structure, which has been widely used for modelling the relationship between two sets of entities in many real-world applications, such as purchasing relationship between customers and products model in E-commerce. Like clique in general graph, biclique is a fundamental structure in bipartite graph, and has been widely used to capture cohesive bipartite subgraphs in a wide spectrum of bipartite graph applications.

For this project, you need to implement an algorithm to find a bipartite subgraph in a bipartite graph. Specifically, given a bipartite graph G = (U, V, E), you need to find a biclique C∗ in G with the maximum size. Considering that many real applications (e.g., fraud transaction detection) require that the number of vertices in each part of the biclique C∗ is not below a certain threshold, we add size constraints τU and τV on |U(C∗)| and |V (C∗)| s.t. |U(C∗)| ⩾ τU and |V (C∗)| ⩾ τV .

Example

Fig. 1 (a) shows a bipartite graph G with U(G) = {u1, u2, ..., u7}, V(G) = {v1, v2, ..., v6}. Given thresholds τU = 1 and τV = 1, the maximum biclique C∗ (G) = C1 is shown in Fig. 1 (b), where U(C1) = {u3, u4, u5, u6} and V (C1) = {v2, v3, v4, v5}. Given thresholds τU =1 and τV =5, the maximum biclique C∗ (G) = C2 is 1,5 shown in Fig. 1 (c), where U(C2) = {u3, u4} and V (C2) = {v1, v2, ..., v6}.



(a) Bipartite Graph $G = (U, V, E)$ (b) Maximum Biclique, $\tau_U = 1, \tau_V = 1$ (c) Maximum Biclique, $\tau_U = 1, \tau_V = 5$

Figure 1: An Example of a Bipartite Graph and its Maximum Biclique

Datasets

Bipartite Stack Overflow favorite network will be provided for this project as dataset. The dataset consists of 641,876 nodes (545,196 nodes in U and 96,680 in V respectively) and 1,301,942 edges in a bipartite graph.

Dataset description: http://konect.cc/networks/stackexchange-stackoverflow/

Dataset link: https://www.dropbox.com/s/b42c7we7owp11m8/download.tsv.stackexchange-stackoverflow.tar.bz2?dl=0

Evaluation

The evaluation will be based on time cost of your algorithm which means how long it takes for the algorithm to return the maximum biclique. The faster biclique subgraph return, the better the performance of the algorithm is.

References

Bingqing Lyu, Lu Qin, Xuemin Lin, Ying Zhang, Zhengping Qian, and Jingren Zhou. Maximum Biclique Search at Billion Scale. PVLDB, 13(9): 1359-1372, 2020.