Paper Recommendation using GraphX

Jeremy Chen
University of Waterloo
jeremy.chen@uwaterloo.ca

Junyi Zhang
University of Waterloo
j823zhan@uwaterloo.ca

Yuqing Xie
University of Waterloo
yuqing.xie@uwaterloo.ca

ABSTRACT

This is abstract

1 INTRODUCTION

With the rapid publication of scientific literature, conducting a comprehensive literature review has become more challenging. Keeping up with current development of a certain also requires huge effort. However, citation network would help to improve the efficiency and quality when we want to survey on a certain field. Since the citation network is a graph structure, we will explore the new framework **GraphX**, which makes graph computation easier. The objective of this project is to learn a new framework GraphX and implement network algorithms to accomplish **paper recommendation**. Our system will recommend users related papers based on input keywords, citation network and reading history.

2 RELATED WORK

This is related work.

3 PROBLEM DESCRIPTION

We will solve the problem of paper recommendation based on keywords, citation network and user reading history. We will have an offline dataset consisting of paper citation relation and papers' content.

For each query, we have the following as the input and output.

Input.

- **Keywords**: a list of strings where each string is an interested field, e.g. ["machine learning", "computer vision"].
- **Reading history (optional)**: a list of papers where each paper is represented as a unique id, e.g. ["journals/cacm/Szalay08"].

Output.

- **Recommended papers**: a list of papers in descending recommending order.

4 FRAMEWORK AND ALGORITHMS

4.1 GraphX

4.2 Algorithms

The basic idea is to first extract features for each paper, filter the matching papers based on keywords that the user entered, then construct the graph of citation, compute the "paper rank", find frequent patterns, and at last recommend papers to the user.

In details, there are three problems we are going to solve, and we propose an algorithm for each of the problems.

- **Extracting Features**. We need features that can help to classify the papers according to keywords. Our idea is to implement an algorithm to first compute the TF/IDF and then collect keywords for each paper. We will also take the original author-generated keywords into consideration.
- **Paper Ranking**. In this stage, we will first use GraphX to construct the citation network. Each node represents a paper while each directed edge represents a citation relation between two papers. After we constructed the graph, we will implement the classic PageRank algorithm to establish a paper ranking.
- **Paper Recommendation**. The goal of the project is to recommend papers to researchers according to their interest. There are two main features to represent users' interest. The first will be keywords. We will filter the papers according to the target keywords and construct a subgraph, then compute the paper rank accordingly. The second will be researcher's reading history. We will find common patterns such as diamond pattern to do recommendation. For example, when a user has read paper A, and paper A cites both paper B and C. Paper B and C both cite paper D. Then, we will recommend paper D to the user if the paper has not been read by the user. This is easy to understand because paper D may be an important work in the reader's interested field. We will also consider other interesting patterns according to experiments.

5 EXPERIMENTAL EVALUATION

Although the focus of this project is to learn a new graph processing framework, we still describe a little bit how we are going to evaluate our proposed solution.

5.1 Dataset

We are planning to use the DBLP Computer Science Bibliography dataset, which can be downloaded here. The raw dataset is in XML format, and we will convert it into a graph structure. Since our project includes a keyword filtering feature, we are also planning to crawl the paper contents for text processing and analysis. For each paper in the DBLP data, a link to the entry page of the paper has been provided. We can crawl the contents of the entry page and obtain the abstract as an offline dataset for text processing. For the feature that is based on the user reading histories, we plan to use (1) our own reading histories, and (2) synthetic histories.

5.2 Methodology

The evaluation methodologies for both keyword-filtering and reading-hisotry recommendation features are similar. Since evaluating the effectiveness of the recommendations by automatic testing is challenging, one way to evaluate our results is manual evaluation. We are going to perform our proposed algorithms on the DBLP dataset described in Section 5.1. For keyword-filtering recommendation, a keyword from a list of randomly generated keywords is going to

be an input. For reading-history recommendation, a reading list is going to be an input. Then, we manually evaluate how relevant the results are to the input keyword or reading history, respectively.

The other way to evaluate our results is based on the number of citations of a paper. We can evaluate our results by checking if the recommended papers are popular/important (i.e. large number of citations). The recommendations are effective if they are among the most-cited papers.

5.3 Experimental Results

This is result.

6 CONCLUSION

This is conclusion.