DBLP's Collaboration Network Based Academic Recommendation System

Mingyue Sun, Fei Chen, Tim Lawrance Fiebrantz, Xiangru Zhou, Xiangdong Wu, Yanjun Xiong
School of Engineering and Computer Science
University of Texas at Dallas, Richardson, TX, 75080, USA

***Abstract*—When researchers are to write a new paper, they often seek co-authors who are knowledgeable on the subject of the paper. But they regularly encounter challenging tacks of selecting the co-author for joint publication, or searching for authors, whose papers are worth reading. Nowadays, a lot of significant information on publication activities of research communities are contained in the modern bibliographic databases. So this project focuses on solving these two problems by building a DBLP's collaboration network based recommendation system(engine). Researchers can use this recommendation engine to select co-authors to cooperate with and to find papers they might be interested in. We applied collaborative filtering ALS algorithm using DBLP computer science bibliography data to build recommendation model. The original data was abstracted into undirected acyclic graph to shrink the scale of data and fits the API which spark MLlib provided. The final result is evaluated by the mean squared error values for both training and testing. And further improvement approach is proposed.**

1. Introduction

One of the main aims of a researcher, is to strive for success and a solid reputation, besides developing knowledge and understanding. Cohesive relationship in a topic-driven community foster researchers success. So, when they are to write a new paper, they always seek co-authors who are knowledgeable on the papers subject.

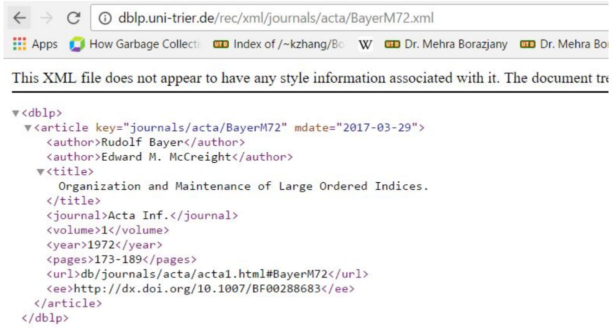
On the other hand, researchers have to deal with hundreds of papers to become familiar with the study fields. But, the number of the papers exceeds human abilities to read them all. The most common way to select relevant papers is by sorting a list of all articles according to a citation index and choosing some articles from the top of the list. However, this method does not take the author professional specialization into consideration. Another way is to choose articles of well-known authors. This project focus on solving these two problems by building a DBLP's collaboration network based recommendation system(engine). To be more specific, the project recommends authors to be collaborating with the given author, and help to predict and choose the papers published by the given well-known author.

This report will cover the overall and detailed description of the project, including the dataset description and preprocessing, solution framework and methods, experimental results and analysis, and conclusion and future work.

2. System Design

2.1. Data Description and Preprocessing

In this project, we used DBLP as our data source. The DBLP computer science bibliography is the on-line reference for bibliographic information on major computer science journals and conferences. It has evolved from an early small experimental web server to a popular open-data service for the computer science community. The dataset is defined in XML document in a tree structure with DTD available. Figure 1 below shows an example (one entry) of the dataset. In the latest release of DBLP dataset, it contains 3,749,969 publications written by 1,899,133 authors and published on 4,112 conferences and 1,525 journals. Figure 2 below shows a pie diagram, and it describes the distribution of the different publication types. As we can see, conference papers and journal articles are the main streams of the dataset.



This XML file does not appear to have any style information associated with it. The document tre

```
▼<dblp>
  ▼<article key="journals/acta/BayerM72" mdate="2017-03-29">
     <author>Rudolf Bayer</author>
     <author>Edward M. McCreight</author>
    ▼<title>
       Organization and Maintenance of Large Ordered Indices.
     </title>
     <journal>Acta Inf.</journal>
     <volume>1</volume>
     <year>1972</year>
     <pages>173-189</pages>
     <url>db/journals/acta/acta1.html#BayerM72</url>
     <ee>http://dx.doi.org/10.1007/BF00288683</ee>
   </article>
 </dblp>
```

Figure 1. Distribution of the Dataset

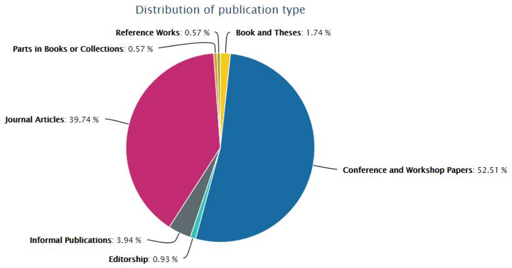
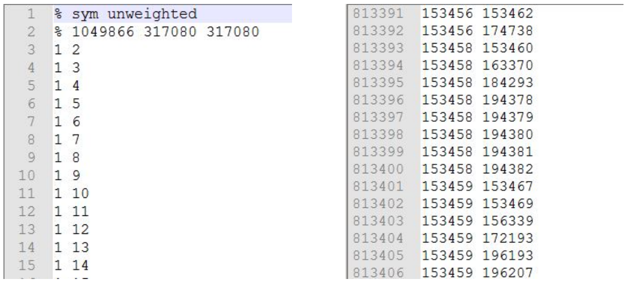


Figure 2. Distribution of the Dataset

We will use ALS collaborative filtering algorithm to build our model in spark using spark MLlib package. The implementation of the ALS program in this library only receive integers as the id of rating records. Therefore, we need to extract the author information from the original data and serialize and reformat the author information. The approaches for dataset processing are similar. So, we will explain the author collaborative dataset preprocessing as an example.

The solution is finding a dataset that constructs a co-authorship network where two authors are connected if they publish at least one paper together. Publication venue, for instance, journal or conference, defines an individual ground-truth community; authors who published to a certain journal or conference form a community. It regards each connected component in a group as a separate ground-truth community and remove the ground-truth communities which have less than 3 nodes.



```
% sym unweighted
% 1049866 317080 317080
1 2
1 3
1 4
1 5
1 6
1 7
1 8
1 9
1 10
1 11
1 12
1 13
1 14
```

```
153456 153462
153456 174738
153458 153460
153458 163370
153458 184293
153458 194378
153458 194379
153458 194380
153458 194381
153458 194382
153459 153467
153459 153469
153459 156339
153459 172193
153459 196193
153459 196207
```

Figure 3. Training Data Sample

This approach will create an AUG(Acyclic Undirected Graph). It has two advantages. For one thing, the weak connections are filtered. For another, it reduces the size of dataset and improved the training efficiency.

2.2. Alternating Least Squares Method

Generally, there are two approaches to build a recommender system: content based filtering and collaborative filtering. The idea of content-based filtering is to examine features of the items recommended. It compares the features of the items and the preferences of a user. Collaborative filtering takes a different approach that it compares preferences and obtained similarities between different users. During our process of building the dblp recommendation system, we refer to some methodologies and approaches introduced in the chapter Movie Recommendation with MLlib of ampcamp Berkeley.

In order to solve this problem, here we applied the well-established latent factor model and using the plain alternating least squares (ALS) algorithm.ALS algorithm is often used in the recommendation system based on matrix decomposition. The algorithm divides the user and item score matrix into two matrices: One is the user preferences for the item preferences matrix, the other is the matrix of hidden features that the item contains. In this matrix decomposition process, score missing items have been filled, and we can be based on the fill of the ratings given the user recommended the best goods.

$$r_{ui} = q_i^T p_u \tag{1}$$

$$min_{p,q} \sum_{(u,i)\in k} (r_{ui} - q_i^T p_u)^2 + \lambda(||q_i||^2 + ||p_u||^2) \tag{2}$$

The reason we chose ALS is that it can be easily parallelized on distributed computation system. The core part of our coauthor recommendation system is to provide coauthor specific ratings for one author. With the vector or multi-authors preference for one coauthor, we can provide better recommendation for a group of authors. In order to solve this problem, here we applied the well-established latent factor model and using the plain alternating least squares (ALS) algorithm. The latent factor model treats the review rating of an item (denoted as i) given by a user (denoted as u) as the dot product of item vector qi and user vector p_u (equation 1). The ALS algorithm calculate the user vector q_u and item vector q_i that can minimize the cost function (Equation 2) by rotating between fixing the q_u and p_u. By applying this strategy, ALS algorithm converts the non-convex problem to two quadratic problems and solves them by least-square algorithm.

In this project, we recommend coauthors based on the preferences of the similar authors calculated by their previous publication. So, what we need is to train model that can find the fittest similarity among authors. We use AlS.train() method to train the modal with the parameters from the prespecified set, where rank from 10, 15, 20, 25, 30, lambda from 0.01, 0.03, 0.05 and number of iterations from 5, 10, 20. Using the well-trained model, we validate with a given author id, and pick the author with the highest likelihood.

The whole dataset was split into three parts, a training dataset with 60% data, a test dataset with 20% data and a validation dataset with the rest 20% data. Only the training and test dataset was used during the training process and the validation data set was used for the final validation. The rank parameter (the number of latent factors) was tuned during the training process. The final model with rank set to 30, gave a RMSE of 0.1312 on validation data set.

3. Experiment Results and Analysis

For the experiments, we split randomly the datasets with the ratio 0.6, 0.2, 0.2 into training set, validation set, and test set. The best model of co-author dataset is trained with parameters: rank is 30, lambda is 0.05 and the number of iterations is 20. The training MSE is 0.1307, and the test MSE is 0.1312. The best model of author-paper dataset is trained with parameters: rank is 30, lambda is 0.05, and the number of iterations is 20. The training MSE is 0.3165, and the test MSE is 0.3161. The result indicates that the model fits well for testing data. For both datasets, the training error and testing error are close which means the overfitting problem is avoided because of our tuning strategy.

Based on the proposed recommendation system, we tested to recommend authors and papers to the author No. 2. Table 1 is the top 5 recommended co-authors for author No. 2. It means that if author 2 would like to seek co-authors, the recommendation system will suggest him/her to contact the author 69, 33, 76, 53 and 90 for the first five choices. Table 2 is the top 5 recommended papers related to author 2. It means if someone is interested in the papaers of author 2, the system will suggest him/her to read paper No. 1028261, 177389, 3089874, 3089874, 1116558, and 612379 first.

TABLE 1. THE TOP 5 RECOMMENDED CO-AUTHORS FOR AUTHOR 2

Recommended Author ID	Similarity(co-author)
69	0.9537912992961237
33	0.9537733908470782
76	0.9537681739381699
53	0.9537655908380562
90	0.953725408929529

TABLE 2. THE TOP 5 RECOMMENDED PAPERS RELATED TO AUTHOR 2

Recommended Paper ID	Similarity(papers)
1028261	0.9660981505957621
177389	0.9562704475171694
3089874	0.9561791891635812
1116558	0.9554525073151288
612379	0.9554156049754379

4. Conclusion and Future Work

The recommendation system built successfully to recommend and predict authors and publications of a given author id. The recommendation model is trained well with our data as it gives minimal mean square error to our dataset. However there are still some improvements can be made to make our recommendation system better. For example, as we mentioned in our data processing section, the rating for each connection (author-author / author-paper) were abstracted into 1 and 0 after filtering by the number of co-existence. It's not accurate enough to reflect how tightly those two components are connected. For instance, author A and author B cooperated in three papers, author A and author C cooperated in ten papers. In this scenario, the connection rating for both A/B and A/C are 1. So the recommendation accuracy may further improved by weighting ratings by the number of connections and normalize them. Furthermore, we can construct additional data files that link each author id to its author name, and link each publication id to its publication name. And we could also add some more features our recommendation system, which is to recommend, based on the publication genres.

References

[1] DBLP computer science bibliography. http://dblp.uni-trier.de/

[2] DBLP Author-Publication Dataset. http://konect.uni-koblenz.de/networks/dblp-author

[3] DBLP Co-authorship Dataset. http://konect.uni-koblenz.de/networks/com-dblp

[4] Movie Recommendation with MLlib of ampcamp Berkeley. http://ampcamp.berkeley.edu/big-data-mini-course/movie-recommendation-with-mllib.html

[5] Rory L. L. Sie , Bart Jan van Engelen, Marlies Bitter-Rijpkema, Peter B. Sloep. COCOON CORE: CO-author REcommendations Based on Betweenness Centrality and Interest Similarity, Recommender Systems for Technology Enhanced Learning, 2014, 267-282

[6] Ilya Makarov, Oleg Bulanov, Leonid E. Zhukov. Co-author Recommender System, National Research University. https://www.hse.ru/data/2017/03/11/1169365288/NET_2016.pdf

[7] Rory L.L. Sie, Hendrik Drachsler, Marlies Bitter-Rijpkema, Peter S-loep. To whom and why should I connect? Co-author recommendation based on powerful and similar peers, Int. J. Technology Enhanced Learning, Vol. 4, Nos. 1/2, 2012