- OK writing only: can be more scientific and english needs improvement
- good data curiosity: you did a lot
- ok technical achievements: lot's of problems things did not work, but you adapted your strategy
- good use of related work
- Ok realization

28

LSDE Group 28 Project Report

CR2: Scientific Communities

Tianhao Xu
Vrije University Amsterdam
Amsterdam
t4.xu@student.vu.nl

Ruijia Lei
Vrije University Amsterdam
Amsterdam
r.lei@student.vu.nl

Yilin Li
Vrije University Amsterdam
Amsterdam
y45.li@student.vu.nl

1. INTRODUCTION

Our project is to process and analyze the metadata of academic papers from CrossRef to create a directed graph that represents the mutual citation relationship between different academic papers or journals. Afterwards, a list of scientific communities will be established by using specific graph clustering algorithm. A scientific community means a group of researchers who work in a specific field and regularly publish at the same venues (e.g., the database research community publishes at VLDB, SIGMOD, ICDE, EDBT, TODS, TKDE, and VLDBJ). In addition, we can also analyze the characteristics of the directed graph of each scientific community to find the writing habits and other characters shared by researchers in a certain subject (e.g., we can find out which communities cite each other most frequently).

In general, we will analyze the 96GB json file to get the mutual citation of academic papers in CrossRef. After preliminary preprocessing, we found that only 44,672,628 records contained citation data of academic articles and can be utilize by this project (these valid records accounted for about 1/3 of all academic paper records). It is expected to use these metadata from CrossRef to complete the establishment of the relationship diagram, build scientific communities with graph cluster algorithm, analyze the characters of every single community, and evaluate the connections between several different communities.

Our project is quite useful and interesting. For scientific researchers, it is very necessary to understand the development and achievements of their research field. CrossRef, which can help researchers realize their research field and inspire them make progress through analyzing the connection between different papers, registers DOI for new and past journal articles and provide inquiry service for searching valuable information. All papers of various publishers registered with CrossRef will automatically establish citation links to each other, thus guiding readers to browse the journals and visit the websites of various publishers conveniently, which greatly increases the chances of journal papers being accessed and cited.

The metadata derived from CrossRef can not only help users find citation data more efficiently, but also assist users to find their own subject communities. Therefore, we hope to be able to further analyze and process the metadata of academic papers, use graph clustering algorithms to establish community list of different research fields and provide a variety of inquiry services related to citation data. In addition, after consulting the literature, we found that few researchers conduct community detection algorithms on such large-scale citation data. The results of our project can evaluate the performance of different algorithms on the complex network of citation relationships.

2. RELATED WORK

Community detection and analyzing complex network remains a significant challenge for many large-scale datasets, especially for dense graphs with complex connections. A near-linear time algorithm, which simply uses the network structure, shows great potential in detecting the complex connections in the network and dividing network into multiple communities [7]. This algorithm, which is a label propagation algorithm, first assumes that node x has neighbors x1,x2 ,...,xk and each neighbor has a label indicating the community they belong to. Then x joins the community to which its maximum number of neighbors belongs. This process will be performed iteratively until all nodes belong to different communities. Considering that it only spends near-linear time on detecting communities, we intended to utilize this algorithm to process the massive metadata from CrossRef. But due to the unsatisfactory performance of the label propagation algorithm on large-scale dataset (i.e., its performance is not good on the dataset of "Actor collaboration" which contains over 200,000 nodes), we still needed to verify the feasibility of this algorithm during our project (which contains over 100M nodes). However, the quality of community detection on larger-scale network remains challenging due to massive sample sizes and insufficient tests on the small set of simple benchmark graphs [3]. According to Fortunato's work, Girvan-Newman (GN) algorithm has good decomposition ability for complex networks and the proportion of correctly identified nodes reaches about 90% (Z_out=6).

The GN algorithm [4] is a classical community discovery algorithm that belongs to the class of split hierarchical clustering algorithms. The basic idea is to continuously remove the edges in the network that have the largest edge meshes relative to all source nodes, and then, recalculate the edge

Let's see...

avoid "very" in scientific writing

lot's of "bla bla" here! keep it to the point

is this a description of future work?

"shed light on"

your ??? experiment? or cited results?

what is "correct" here?

meshes of the remaining edges in the network relative to all source nodes, repeating the process until all edges in the network, have been removed. The GN algorithm is a cohesive community structure discovery algorithm. The algorithm progressively removes edges between communities to obtain a relatively cohesive community structure based on the characteristics of high intra-community cohesion and low inter-community cohesion in the network as Figure 2. The algorithm uses the concept of edge mesonumber to detect the location of edges, and the edge mesonumber of a particular edge is defined as the number of times the shortest path between all vertices on the network passes through that edge. From the definition, if an edge connects two communities, then the number of shortest paths between these two community nodes through the edge will be the highest and the corresponding edge mesonumber will be the largest. The GN algorithm is based on this idea of iteratively calculating the shortest path of the current network, calculating the number of edges meshes for each edge, and deleting the edge with the largest edge meshes. Finally, under certain conditions, the algorithm stops and the community structure of the network is obtained. Considering that the GN algorithm can classify closely connected nodes, we intended to use this algorithm for complex networks based on CrossRef.

Our main tasks are: ? really ?

(1) Verifying the feasibility of the GN algorithm. Since the time complexity of this algorithm is $O(n^2m)$, we proposed to perform special methods on large-scale datasets to successfully execute this algorithm, which means we need to design a reasonable plan to reduce the size of the data.

(2) In addition, verifying whether the GN algorithm can solve the problem of imbalanced community division (few communities occupy most of nodes) is another task of our project. Moreover, Blondel and his partners proposed the fast-unfolding algorithm [1], which is a heuristic method based on modularity optimization, can complete community detection in a shorter time. To evaluate the division of communities, Newman et al. [4] proposed the concept of modularity Q, using modularity to measure the quality of community division. In simple terms, it is to group the densely connected nodes into a community, so that the value of modularity will become larger. In the end, the result of division with the largest value of modularity is the optimal result. According to Newman's Newman, 2003 and Blondel's work, assume that A_ij is the weight of the edge between node i and node j, $k\_i = \sum\_j A\_ij$ is the sum of the weights of the edges attached to vertex i, δ is a function which ensure $i \neq j$. Then, modularity Q can express as Formula 1.

$$Q = \frac{1}{2m} \sum_{I,j} [A_{ij} - \frac{k_i\ k_j}{2m}] \delta(c_i, c_j) \quad (1)$$

We planned to use the fast-unfolding algorithm and optimize the modularity Q to get an ideal result of community detection and compare with the result of PageRank [2] for evaluation. In addition, we also tried to solve a common problem in the process of community detection, which is the imbalanced community division.

3. RESEARCH QUESTIONS

(too) many RQ's here

(i) How to analyze and process the metadata in CrossRef? How to establish citation relationships between different papers? How to establish citation relationships between different journals?

(ii) How to execute the community detection algorithms and generate a list of scientific communities in a limited time?

(iii) If the cluster cannot process all the data of citation relationships, how should we simplify the directed graph and reduce the size of the data?

(iv) Compare the effects of different algorithms on the community detection of citation data. According to the investigation on previous part, we believed that the effect of fast-unfolding algorithm is better than that of Girvan and Newman algorithm and the effect of Girvan and Newman algorithm is better than that of label propagation algorithm.

(v) Based on the result of community detection, which communities are the most closed communities (which means papers in those communities seldom cited papers in other communities)? How to define the closed community? ??

(vi) Based on the result of community detection, which communities cite each other most frequently? How to define the frequently cited community?

(vii) How to prove that our community division is reasonable? Can we use PageRank to judge whether our community division is reasonable or not?

(viii) Which tool is more suitable to complete our project (GraphFrame, iGraph or Gephi)? If the citation data is changed or updated, how should we deal with this situation and ensure the robustness of our solution?

is it possible to separate algo evaluation from tool evaluation ?

4. MATERIALS AND METHODS

4.1 The Overview of Scientific Community Detection

Figure 1 summarized the overview for the preprocessing and detection phases of our work. The preprocessing phase (Figure 1, top red panel) of extracting valuable data for detection consists of three steps:

(1) Evaluating the characteristics of metadata

(2) Extracting the necessary data (i.e., DOI, journal's title, references, etc.) and establishing "two-tuple" representing the relationship of reference (e.g., <A.DOI,B.DOI> means paper A refers to paper B).

(3) Building reference relationships between different journals based on the result of step (2).

In step (1), we analyzed and processed a total of 40,229 json files, about 96GB. ~~Due to~~ Because some metadata lacks core elements, we first delete this part of the metadata and analyze the "references" column in detail to extract the DOI of the articles cited in each paper. Then, we utilized "DOI" of each paper and generated the table to illustrate the citation relationship between different papers in step (2). In step (3), considering that we got more than 100M papers in previous step, we found that the total number of citation

What?

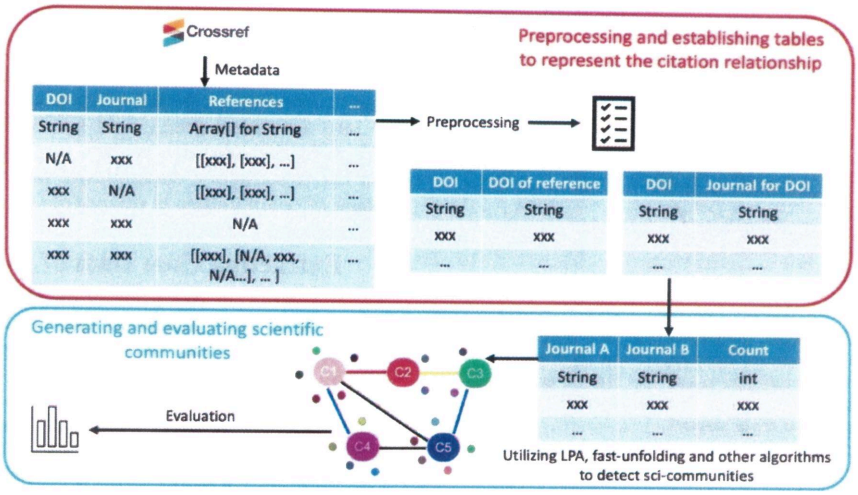

Figure 1: Overview of scientific community detection. Red panel: Extracting "references" data from json files and establishing the citation relationships between papers. Blue panel: Establishing the citation relationships between journals and utilizing community detection algorithm to group journals into sci-communities.

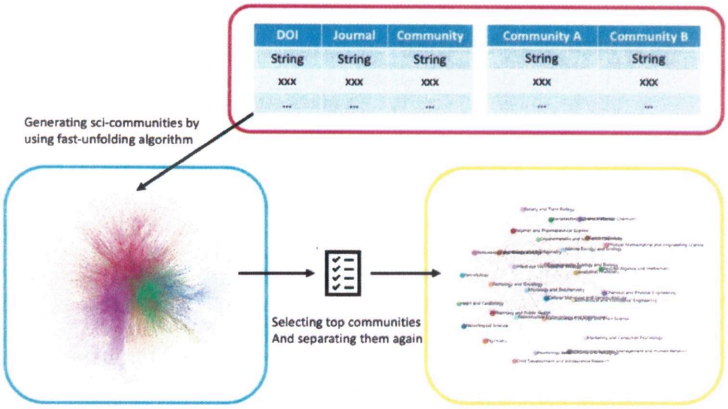

Figure 2: Overview of generating scientific communities. Yellow panel: After selecting several communities which contain too many papers, we implement algorithm on those communities again in order to enhance the effect of community detection.

relationships of these papers will be an "astronomical number" and it is impossible for us to use community detection algorithms on all nodes. Therefore, we generated another table to illustrate the citation relationship of each journal. In this table, each journal contains thousands of papers, and <A.Journal,B.Journal,count> means the papers in Journal A refer to papers in Journal B for "count" times. We intended to implement the community detection algorithm on this table.

The detection phase (Figure 1, bottom blue panel) of establishing scientific communities need to input vertices and edges to build a directed graph (i.e., "vertices" means all processed journals in previous phase and "edges" means the citation relationship between different journals). Then, we selected the suitable algorithms to analyze the directed graph and establish scientific communities. The fast-unfolding algorithm on Gephi is the most properly algorithm for this dataset and we could monitor the modularity as the evaluation indicator to divide journals into different communities.

4.2 Preprocessing Metadata and Extracting Citation Relationship

A total of 120,765,146 academic papers' metadata is recorded in all json files, of which 44,672,628 papers include the "references" column. In the metadata, about 15%-20% of the data lacks the columns of "DOI", "References" or "the container-title" (i.e., the name of journal). In the process of processing metadata, we treated the metadata as input and delete the records which lacks core columns, then implement special processing on column of "references" to extract citation relationships between different papers. Specifically, the DOI of each paper corresponds to a unique list of references which can express as Rn.The list of references contains the metadata of M papers which cited by paper N (i.e., Rn = [A1,A2,A3... Am]). Similarly, each element in the list Rn also contains several columns of data to describe the cited paper, for example, A1 will express as [DOI,container-title,publisher...]. Considering that any academic paper can cite 0-N other academic papers, we processed the ref-

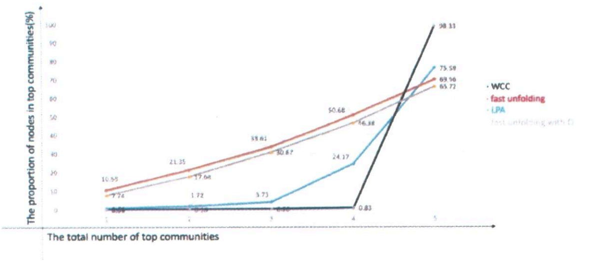

Figure 3: Line chart of network structure (i.e., the more papers "top" communities have, the more unbalanced the division of the community). X axis: represents how many top largest communities' papers have been counted. Y axis: The proportion of papers in top largest communities to total papers.

erences of each paper in the json files sequentially and use "two-tuple" format (i.e., <A.DOI,B.DOI>) to express the citation relationship between different papers. If paper B lacks the core elements, we still need to delete the row of this paper. However, considering commonly used community detection algorithms' time complexity are high, it is almost impossible for the cluster we used to implement community detection algorithm for directed graph which contains over 100M vertices.

With the aim of reducing the size of data input in cluster, we proposed to utilize "join", "groupby" and "count" from SQL to process the tables in Figure 1 and then generate the citation relationship between different journals. In detail, we regarded each journal containing several papers as a vertex and counted the number of mutual citations between different vertices. For example, if papers in Journal A cited papers in Journal B 1000 times in total, we will generate a row which contain the name of Journal A, the name of Journal B and the number of citations. The top row of Table 1 shows the characteristics and distribution of processed metadata after step (2). Through decomposing and screening the column of references, we found that there are a total of 120,644,459 papers with valid DOI, and these papers have a total of 995,613,570 citation relationships with other papers. Then, we deleted the rows which lack key elements (i.e., assuming citation relationship between A and B express as <A.DOI,B.DOI>, we will remove the row of this relationship if B.DOI lacks the name of journal or the "DOI" is null), summarized the citations between different journals according to the journal name of the paper, and recorded the number of citations. After processing the output in step (2), we summarize 534,866 different name of journals and these journals generate a total of 51,650,203 citation relationships (Table 1, middle row).

In the phase of analyzing and pre-processing metadata, we found that the types of elements stored in each json file are different. Therefore, we intended to process each json file separately and decompose the column of "references" to extract the citation relationships for each paper. If all the papers recorded in a file lack key elements such as DOI or references, we will skip this file (Figure 1, left table on blue panel shows the details). It took about 30 hours to preprocess the metadata and generate tables which can be utilized in the community detection phase. The whole process executed on gpu machine (g4dn.xlarge, 16 GB, 1GPU) for 18 hours and executed on shared cluster (i3.xlarge, 30.5GB, 4 cores) for 12 hours, which spent about \$20.8 (12×8×\$0.0936+18×1×\$0.658). In the next, we proposed to convert the citation relationship between 0.5M journals into a directed graph, and use BGLL, Louvain, LPA, WCC and other community detection algorithms to generate a list of scientific communities and verify whether the division of communities is reasonable or not.

Why a GPU machine??

4.3 Extracting Core Data of Journals

Table 2 shows the execution of different algorithms under different input data. We found that we cannot even execute the simplest algorithm, which is LPA, with over 50M vertices and 100M edges on iGraph. Compared with iGraph, GraphFrame's LPA algorithm can be executed (Table 2, the eighth row), but it takes over 10,000 seconds (about 2.7 hours) to iterate one time. Considering that the LPA algorithm needs to input the number of iterations, and we do not fully understand the structure of the directed graph, we finally found it is difficult to optimize the result of community division and this algorithm will spend too much time and budget. In addition, we also realized that the result of community division obtained by the LPA algorithm is very unsatisfactory (i.e., the top three largest communities contain over 98% of vertices), if all 0.5M vertices are used as the input of the algorithm. Therefore, we intended to select the core journals among 0.5M journals and analyze the mutual citation relationships between these core journals.

What is a "core" journal?

In the previous phases, we expressed the citation relationship between different journals as <A.Journal,B.Journal,count>, where Count represents the number of times that Journal A cites Journal B. The larger the count, the more important the citation relationship between A and B. Table 3 summarized the proportion of edges with different count in all edges. According to the information in Table 3, we found about 50% of the edges' count are 1, which means there is only one valid citation relationship between these journals, and these edges cannot represent the core citation relationships in the directed graph. Therefore, we finally selected the citation relationships with count greater than 1000 as the edges of the directed graph. Although only 72697 citation relationships and 8,768 journals meet the requirements, the total number of papers contained in these citation relationships accounts for 68.09% of all papers (Table 1, the bottom row). This proves that the 8,768 journals are the "core journals" in CrossRef, and implementing community detection algorithm for them is representative. Then, we treated "core journals" data as the input to different community detection algorithms.

4.4 Label Propagation Algorithm Aided Community Detection

We used the "GraphFrame" package and "igraph" package in Pyspark to quickly establish the directed graph (based on the table in Figure 1, bottom blue panel). Before using the Label Propagation Algorithm (LPA) for community division, we first utilized the Weakly Connected Components (WCC) Algorithm for preliminary processing of the directed graph. Considering that WCC is usually used early in an analysis to understand a graph's structure [6], we intended to use WCC to decompose the directed graph into several small graphs to reduce the running time of other complex community detection algorithms (e.g., LPA, BGLL, etc.)

Table 1: The number of selected papers and journals in different steps.

Step	Number of papers	Number of journals	Number of citation relationship (two-tuples for papers)	Number of citation relationship (two-tuples for journals)
Step 2	120,644,459	N/A	995,613,570	N/A
Step 3	56,001,423	534,866	723,518,012	51,650,203
Core Journals	38,129,337	8,768	528,089,286	72,697

Table 2: The result of different algorithms executes with different package. *: executing on large-scale dataset

Package	Algorithm	Vertices	Edges	Running Time (s)
iGraph	Newman*	56,001,423	120,472,859	OOM error
iGraph	Newman	8,768	72,697	N/A
iGraph	BGLL*	534,866	51,650,203	N/A
iGraph	BGLL	8,768	72,697	35.93
iGraph	LPA*	56,001,423	120,472,859	OOM error
iGraph	LPA	534,866	51,650,203	N/A
iGraph	LPA	8,768	72,697	17.56
GraphFrame	LPA	534,866	51,650,203	10,632 (iteration=1)
GraphFrame	SCC	534,866	51,650,203	N/A
GraphFrame	WCC	8,768	72,697	126.53

Table 3: The number of citation relationship between journals (filter by "count")

Count	Number of citations (two-tuples for journals)	Percentage (%)
All count	51,650,203	100.00
Greater than 1	24,821,351	48.05
Greater than 10	6,202,084	12.00
Greater than 1000	72,773	0.13

Table 4: The division result of community detection and "PageRank"

Communities - fast-unfolding	Communities - "PageRank"
Cellular and Biology	Medicine
Biochemistry	Biology
Medicinal Chemistry	Botany
Nephrology and Biochemistry	Nature
Chemical Physic	Chemistry
Material Chemistry	Materials
Botany	Sociology and Psychology
Mathematics	Mathematics
Applied Physic	Physics
Nature and Biology	Statistic

which will be used in the following steps. However, according to the result summarized in Figure 3 (black line), over 97.5% vertices were grouped in one community, which means 8,549 vertices connected with each other and the citation relationships between them is complex. Therefore, we proposed to select a community detection algorithm which is sensitive to the structure of directed graph.

Considering that LPA can smoothly find multiple significant community structures on the dataset of World Wide Web (WWW) and reasonably divide nodes into different communities [7]. We first select LPA to detect communities on our dataset of "core journals". The main idea of the Label Propagation Algorithm: if node i has n neighbors, and each neighbor has a label to indicate the community they belong to. Then node i join the community to which the label with the most neighbors belong. At the beginning, the label of each node is different, and the label is spread in the whole graph.

Finally, the closely connected nodes will be grouped into the same community after several iterations. As a result, LPA divided 8,768 vertices into 162 different communities and the top five largest communities contained about 75.59% of all vertices (Figure 3, blue line). The result reveals that LPA still has the problem of imbalanced community division, which means it will group too many vertices are grouped in few huge communities, instead of successfully dividing the closely connected vertices into several different communities.

4.5 Fast Unfolding Algorithm Aided Community Detection

Compared with LPA, Fast Unfolding Algorithm, which is a heuristic method based on modularity optimization and community aggregation [1], can better process and divide closely connected directed graphs into communities. This algorithm consists of two steps: The first step is called Modularity Optimization, which mainly divides each node into the community where its connected nodes are located. In this step, the value of modularity will gradually increase; the second step is called Community Aggregation, which mainly aggregate the divided communities into points, that is, to reconstruct the network. This algorithm will iterate the above steps until the amount of change in the modularity ($\triangle$Q) no longer increases. As a result, Fast Unfolding Algorithm divided 8,768 vertices into 86 different communities and the top five largest communities contained about 69.56% of all vertices (Figure 3, red line). In addition, we also found that the slope of the red line is lower than the slope of the black line, which means that 8,768 vertices are more evenly distributed to different communities.

Since Fast Unfolding Algorithm requires multiple iterations and judges whether the community division is reasonable through monitoring $\triangle$Q, we can adjust multiple parameters in Fast Unfolding Algorithm to further optimize the

Table 5: The leaderboard of the most frequently cited communities

Community A	Community B	Coefficient
Cellular, Molecular and Genetic Biology	Micrology and Boichemistry	2.230385155
Micrology and Biochemistry	Nephrology and Biochemistry	1.430194044
Neuroscience, Cytology and Brain Science	Micrology and Boichemistry	1.233786139
Nephrology and Biochemistry	Cellular, Molecular and Genetic Boilogy	1.214184273
Nanotechnology and Materical Chemistry	Organometallic and Medicinal Chemistry	1.079479762
Micrology and Biochemistry	Immunology and Rheumatology	0.96907418
Cellular, Molecular and Genetic Biology	Nanotechnology and Material Chemistry	0.90266603
Nanotechnology and Material Chemistry	Physical, Mathematical and Engineering Science	0.778672011
Chemical Physic	Organometallic and Medicinal Chemistry	0.60845795
Micrology and Biochemistry	Infectious Diseases and Virology	0.586808216

Table 6: The leaderboard of the most closed community

Closed Community	Coefficient
Occupational Therapy	0.95275374
Physical Education	0.941343761
Public Administration	0.941034976
Linguistic	0.93965033
Economic History	0.924067277
Hygiene	0.92163631
Philosophy	0.919644342
Wound Care and Nursing	0.905041836
Educational Psychology	0.897713377
Financial Economics	0.896256405

result of the community division. After consulting many documents, we found that "Gephi" provides an optimized Fast Unfolding Algorithm that can monitor and adjust the value of Q. Our optimization plan is divided into two steps: (1) We generate as many communities as possible while ensuring that Q is not less than 0.5.
(2) Use Fast Unfolding Algorithm again for the communities which contain excessive vertices to further divide the scientific communities (Figure 2, right yellow panel).

As a result, this optimized method divided 8,768 vertices into 83 different communities and the top five largest communities contained about 65.72% of all vertices (Figure 3, grey line) Then, we implemented the algorithm again for the top five largest communities and finally generate 128 scientific communities. In general, considering that only 65.72% of all vertices were grouped in five communities and the largest community only contained about 300 vertices, we realized that Fast Unfolding Algorithm can better process the directed graph and generate communities based on "core journals".

4.6 Visualization Motivation

In the visualization part, to increase its features, we added interactive experience and animation effect in four pages, we believe a beautiful page and interesting content can not only increase the user operability but also strengthen the visual perception and diversify the way of data demonstration. We presented a word cloud integrated by journals and publishers on the main page, a dot matrix of community detection generated by different layout functions, and a drag-and-drop interactive map of the most frequently cited communities. Moreover, by clicking the navigation buttons, you can find the statistics and analysis of other data. There are two pie charts which can be exploded based on the amount of individual scientific references related to the range of indegrees and outdegrees. The degree can show the intensity of the is-referred or refers others relation.

Also, we think it is sensible to demonstrate the PageRank result within the field distribution that reflects which kind of research fields are most influential. The most frequently cited communities are presented in a big table on one page which is the target that the project wants to achieve. Finally, there is a column chart containing the rank of the most closed communities, it will be extended animatedly and ordered by the closed coefficient. All the data is included in a folder for creating this visualization. Great data products should be presented well-organized and concisely for no matter data scientists or users. For the addition of animation effect and interactivity, we believe it will be beneficial for capturing the audience's attention. The visualization web pages can be accessed by the URL https://billysen.github.io/.

5. RESULT

5.1 Evaluation of the Scientific Communities

There are three goals in our project:
(1) Processing raw metadata and generating scientific communities through analyzing citation data and using properly community detection algorithm.
(2) Finding the most closed communities, which means papers in those communities only cite few papers outside their communities.
(3) Finding a pair of communities which cite each other most frequently. Based on the citation data, we have achieved the identification and division of the scientific communities through the previous steps.

According to statistics, we divided 8,768 core journals into 128 different scientific communities. The research topics of these communities are mainly in the fields of medicine, biology, cytology, mathematic, applied physics and materials chemistry (Table 4). We also summarized the academic fields which their papers have high value of PageRank [5]. As shown on Table 4, the most influential fields are medicine, biology, botany, nature, chemistry, and materials science, etc. We found that the academic fields of the generated scientific communities basically overlap with the most influential fields. Considering that PageRank is an efficient way of measuring the importance of academic papers, we believed that we not only accurately generate scientific communities, but also accurately extract core journals from all journals

Table 7: Computation time and cost

Step	Cluster	Machine Time(h)	Cost($/h)	Total
Preprocessing metadata	GPU	18h*1*2	0.658	
Preprocessing metadata	Shared	12h*8*2	0.0936	
Extracting "core journals"	Shared	2h*8*2	0.0936	
Establishing citation relationship	Shared	4h*8*2	0.0936	
Creating graph for WCC	GPU	1h*1	0.658	
Executing LPA for all journals	Shared	3h*8	0.0936	
Executing LPA, fast unfolding	GPU	3h*1*2	0.658	
Finding closed communities	GPU	2h*1	0.658	
Finding frequently cited communities	GPU	1h*1	0.658	
Saving results and other testing work	GPU	15h*1*2	0.658	
Saving results and other testing work	Shared	15h*8*2	0.0936	
Storage 300GB	N/A	1 month	6.9	
		552 * 0.0936 + 76 * 0.658 + 6.9 =		$108.57

in CrossRef. In addition, we have completed an additional work based on the results of the community division. According to the tables (Figure 2, top red panel) we generated, we can enter the DOI of the cited paper to query the community it belongs to. But because each community contains too many journals and it is difficult for us to name all communities, we finally did not visualize this query system.

The second goal is to find the most closed communities, which means we need to find the communities with the most internal citations and only cite few papers outside their communities. Suppose that community A contains a total of N papers from M different journals. Among them, the number of times N papers cited each other is Cn, and the number of times N articles cited other papers belonging to other communities is (Cn). Then, the degree of closure of community A (Closed-Coefficient) can be expressed as:

$$Closed - Coefficient(A) = \frac{C_n}{C_n + \widehat{C_n}} \quad (2)$$

The higher the value of Closed-Coefficient(A), the higher the proportion of internal citations in community A to the total citations, which indicates that community A is a closed community. According to the result shows on Table 6, the most closed communities are Occupational Therapy, Physical Education, Public Administration, Linguistic, Economic History, Hygiene, Philosophy, Wound Care and Nursing. The number of internal citations in these communities accounts for 90% of the total number of citations.

The third goal is to find a pair of communities which cite each other most frequently. With the aiming of complete this target, we first transferred the directed graph to an undirected graph. There are 128 communities will be treated as vertices and 603 edges will be generated. Each edge represents the citation relationship between different communities and the "weights" of each edge means the number of citations between these two communities. Assume that community A (which contains S papers) has citation relationship with i other communities. Then, the number of mutual citations between each community and A can be expressed as:[N1,N2... Nk... Ni]. Similarly, the number of papers contained in each community can be shown as: [S1,S2... Sk... Si].Then, the closeness of communities A and Ai can be expressed as Formula 3.

$$Reference - Coefficient(A, A_i) = \frac{\widehat{N_i}}{S_i + S} \quad (3)$$

Reference-Coefficient(A,Ai) means the average number of mutual citations between community A and community Ai for each paper. According to the result shows on Table 5, we found scientific communities which frequently cite by each other usually belong to the fields of biology, chemistry, medicine, and applied physics.

In general, by comparing the results of the community division with the most influential fields in CrossRef, we found that the scientific communities we generated are reasonable. At the same time, the reasonable results also proved that our plan to extract core journals and divide the community is feasible.

5.2 Computation Time and Cost

See Table 7 for details.

5.3 Evaluation of packages and tools

Spark provides RDD, Delta Lake and Parquet Format helps speed up processing progress and reduce data processing time a lot. GraphFrame is a graph processing framework that supports python. Co-work with Spark Dataframe, users can generate graphs easily with cleaned and extracted data by Crossref dataset. However its built-in Label Propagation Algorithm performs not that good.Igraph-CDlib combination is the further option we selected for implementing algorithms and we recommend it for engineers.

After conducting scientifically validation for different feature oriented algorithms execution result, we abandoned some most representative unbefitting algorithms including Walktrap, Fastgreedy, LPA and Grivan-Newman. Louvain and Fast unfolding are fast and they can make communities well-categorized base on the modularity.Gephi is the master of visualizing the community distribution with its powerful and beautiful layout function which helps a lot for the validation stage of our work.

6. CONCLUSION

The first phase in our project is to analyze and preprocess the metadata from CrossRef. Due to the structure of metadata is complex and all references of each paper are contained in an array, we process all json files sequentially

and transfer citation relationships from the metadata to the "two tuples" format, which express as <A.DOI,B.DOI>. However, we totally summarized over 700M rows of citation data and over 50M papers. Considering that no community detection algorithm can executed smoothly on this scale of data, we intended to merge all papers into about 0.5M groups. If paper A and paper B are divided into the same group, that means these two papers are published on the same journal.

The second phase is to group academic papers into different journals and establish the citation relationships between different journals. The citation data of journals expresses as <A.Journal,B.Journal,count>. We finally generated about 50M rows of citation relationships and over 500,000 different journals. Then, we found the size of citation data still need to be reduced due to few the community detection algorithms can execute on this size of data and the problem of imbalanced community division.

Then we analyzed the distribution of the citation data of journals in order to find the core journals which can represent the structure of the network but contain less vertices and edges. According to the result shown on Table 3, over 50% of papers are grouped in only 8,768 journals and over 25M citation relationships between different journals are "weak" (i.e., these journals only cited each other once). Therefore, we expected to utilize these "core" journals to represent the structure of the citation network. And we also verified the effect of community detection with the result of selecting influential field by PageRank in the following step.

After generating the citation data of core journals and reducing the size of input, we utilized several different community detection algorithms to separate groups of frequently mutually cited academic papers into the same scientific communities. Figure 3 and Table 4 shows the result of community detection through executing different algorithms, we got two conclusions based on the test results:
(1) The fast-unfolding algorithm is the most ideal one to detect scientific communities. As shown in Figure 3 and Table 2, this algorithm not only executes efficiently, but also can solve the problem of unbalanced community division. Specifically, the five largest communities of the LPA algorithm contain 75% of journals. That means these large communities are not completely divided and contains many journals belongs to different fields. Compare with LPA, the top five largest communities of fast-unfolding algorithm only contain about 65% of all core journals.
(2) Many journals in CrossRef are closely connected, and it is difficult for general community detection algorithms to reasonably divide these journals into different communities. According to the results of the WCC algorithm, about 98% of journals can be represented on a weakly connected component. That means we need to iteratively use community detection algorithm and divide communities one by one according to the closeness of connection between different journals (Figure 2).

Then we also found the most closed communities based on the result of community detection. All the details are shown on 5.1. As a result, Occupational Therapy, Physical Education, Public Administration, Linguistic, Economic History, Hygiene, Philosophy, Wound Care and Nursing were selected as the most closed communities which only cite few papers belongs to other communities.

About finding a pair of communities which cite each other most frequently, we designed "reference-coefficient" to represent frequency of two communities cite each other. All the details are shown on 5.1. Besides, to verify our result is reasonable or not, we calculate the value of "PageRank" for papers from CrossRef. "PageRank" is an efficient algorithm to analyze the impact of papers. A paper with a high PageRank-value means that many other papers will cite each other around it and form a closely related group (i.e., similar with organize a scientific community). Therefore, we can compare the result of PageRank with the result of community detection to prove our division is reasonable or not. In general, our classification results are basically consistent with the PageRank's result. This also proves that our community division is feasible and reasonable.

In this project, we also found some new questions. First, due to the complexity of our network, we found that some scientific communities contain many vertices, and this will influence the effect of community detection. If the network is not preprocessed, the community detection algorithm will not execute smoothly. At the same time, Fortunato also pointed out that few researchers currently study how to evaluate the effectiveness of community detection algorithms. Therefore, in future research, we hope that researchers can focus on how to reasonably divide closely connected network and explore methods to assess the rationality of the division. In addition, due to the large amount of data from CrossRef, we found that it is completely infeasible to directly use the community detection algorithm on the citation data. We need to reduce the number of nodes and edges in the network to efficiently complete the tasks of detecting communities (even if we use better instance, the running time of the algorithm is still unacceptable). We hope that future research can focus on how to effectively use community detection algorithms for large-scale and complex networks.

7. REFERENCES

[1] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[2] Z. Chen, I. I. Pottosin, T. A. Cuin, A. T. Fuglsang, M. Tester, D. Jha, I. Zepeda-Jazo, M. Zhou, M. G. Palmgren, I. A. Newman, et al. Root plasma membrane transporters controlling k+/na+ homeostasis in salt-stressed barley. *Plant physiology*, 145(4):1714–1725, 2007.

[3] S. Fortunato. Community detection in graphs. *Physics reports*, 486(3-5):75–174, 2010.

[4] M. Girvan and M. E. Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.

[5] R. Nag, D. C. Hambrick, and M.-J. Chen. What is strategic management, really? inductive derivation of a consensus definition of the field. *Strategic management journal*, 28(9):935–955, 2007.

[6] M. Needham and A. E. Hodler. *Graph Algorithms: Practical Examples in Apache Spark and Neo4j*. O'Reilly Media, 2019.

[7] U. N. Raghavan, R. Albert, and S. Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical review E*, 76(3):036106, 2007.

have all your research questions been answered?

Table 8: Contribution of group members

Member	Contribution of work
Ruijia Lei	Project Pipeline Design Data Extraction, Transformation, Clean Coding (Graph part) Tough Problem Debugging Algorithm Implement and Validation Web page with Visualization Data Product Design
Yilin Li	Project Pipeline Design Data Pre-processing Data Extraction, Transformation, Clean Coding(Raw Data part) Tough Problem Debugging Algorithm Implement and Validation Critical Decision Change for Data Utilization
Tianhao Xu	Project Pipeline Design Organization of related materials LateX Layout Reliability availability analysis Data Product Design