Building a mathematics' graph

Gaëtan Ecrepont, Ilyas Glaib

May 2023

Abstract

In this report, we present a knowledge graph of mathematics which we built from scratch using both Wikidata and Wikipedia. It must not be understood as a graph of actionable mathematical knowledge to be queried but rather as a graph of historical knowledge about mathematics, seeking to answer general questions such as how mathematicians are linked to each other, how theories are structured, and what links exist between mathematicians and their theories.

Contents

1 Introduction

Resource Description Framework (RDF) graphs are a way to represent and organize data using triples. Triples are relationships of the form (*subject*, *predicate*, *object*), where *subject* represents an entity, *predicate* specifies an attribute of the

subject, and *object* is the value or object of that attribute. For instance, (Tom, LIKE, Apple) means that Tom likes apples, (Rice, IS, White) means that rice is white, etc.

RDF graphs can be used to build knowledge graphs using web data. The first successful attempt to turn web resources into actionable knowledge dates back to 2007, when DBpedia built their graph using content extracted from Wikipedia pages. A few years later in 2012, Google introduced their now famous Knowledge Graph, which was built upon DPpedia, Freebase and other sources. Since then, several large companies have built their own knowledge graph, including Airbnb, Facebook and LinkedIn.

An interesting feature is that one can build smaller graphs targeting a specific area using data from larger knowledge graphs. For instance, two students from Télécom Paris have extracted several standardized subgraphs from the main Wikidata graph, each focusing a specific topic [1]. In this project, we built a knowledge graph specifically about mathematics. Unlike graphs representing actionable mathematical knowledge such as the Wolfram Knowledgebase [3], our goal was to explore the historical dimension of the topic, which included finding famous mathematicians, tracking the theories which they have contributed to, identifying the universities where they studied, etc.

Perhaps surprisingly, it seems like there have been no attempts to build such a graph. We thus had to build our graph from scratch, and we began by searching online for data sources containing information about mathematics. In the next section we describe the structure of our graph and how we built it.

2 Graph description

Graph overview

Our graph has 4 types of nodes and 5 different relationships, which are described in the tables below.

node type	properties	count
Mathematician	name, date of birth, date of death	3753
Country	name	184
University	name	1309
Theory	name	459

relationship	subject → object	count
CONTRIBUTED_TO	Mathematician → Theory	8278
STUDIED_AT	Mathematician → University	6620
DOCTORAL_ADVISOR_OF	Mathematician → Mathematician	2160
HOME_COUNTRY	Mathematician → Country	5031
SUBTHEORY_OF	Theory → Theory	1587

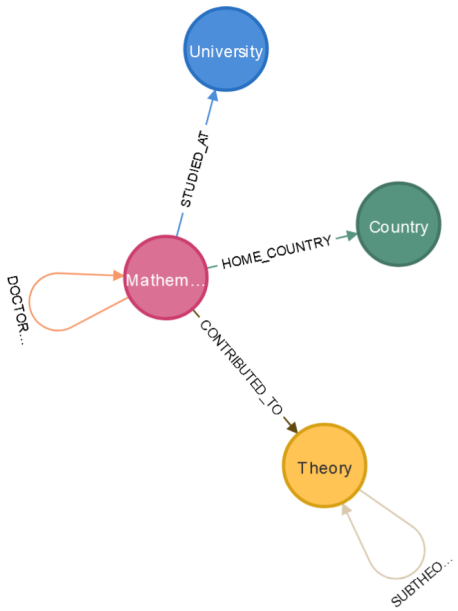

Figure 1: Graph structure

Data sources

When it comes to knowledge bases there are plenty of sources available, varying in size, quality, and focus. Since there exists no knowledge base specialized in mathematics, we decided to use the largest sources because they include, among

others, knowledge about mathematics. We thus looked at DBpedia, YAGO and Wikidata which are some of the largest open knowledge bases. After a brief review, we decided to go with the latter because although all three sources essentially contained the same mathematicians, Wikidata proved much more exhaustive, consistent and accurate. We thus used Wikidata to extract all the information we wanted regarding mathematicians, namely their date of birth, date of death, home country, universities, doctoral students and fields of work. Initially we also wanted to add all the doctoral advisors for each mathematician, but this dramatically increased the total number of rows returned and therefore triggered a timeout error from Wikidata, meaning that we couldn't retrieve the data. Since we already had all the doctoral students for each mathematician, we decided not to include the list of doctoral advisors for each mathematician[1].

Linking mathematicians and theories

One of the main goal of our project was to identify the connections between mathematicians and the theories they had worked on (CONTRIBUTED_TO edge), as well as the interdependencies between mathematical theories (SUBTHEORY_OF edge). Initially, we found that Wikidata's `field of work` attribute contained the domains which the entity (here, a mathematician) had contributed to. However, there was a problem: these domains were usually too general (e.g. mathematics, algebra, number theory) and many did not pertain to the realm mathematics (e.g. physics, philosophy, mechanics). We wanted more granular theories such as chaos theory, fractal geometry, complex analysis, etc. However few of these mathematical subtheories existed on Wikidata and even fewer had links between them. We thus needed another knowledge source to complete our graph. We knew that Wikipedia had even more data than Wikidata but of course it was much harder to retrieve: Wikidata can be easily queried using SPARQL whereas Wikipedia has no API and one must therefore perform web scrapping to extract data from it.

After a bit of research, we found a glossary of areas of mathematics on Wikipedia [2], which we used to create the theories' nodes. We then used the links in the glossary and the handy "Category" property on each theory's page

[1] Although the relationships DOCTORAL_ADVISOR_OF and DOCTORAL_STUDENT_OF should theoretically be perfectly reciprocal, in practice we've observed some discrepancies, mainly due to Wikidata's partial accuracy.

to find the edges between theories.

After obtaining the list of all relevant theories, we had to find a way to link each mathematician to the theories they had worked on. We did so by going through the Wikipedia page of each mathematician and parsing its content to retrieve all the occurrences of theories contained in our list of theories. For every theory which appeared at least once in the mathematician's Wikipedia page, we considered that they had worked on that theory and we thus added it to the list of theories that they had contributed to.

This idea worked well except when the said mathematicians had a poor Wikipedia page, which was usually the case with lesser known scientists. We also found that most of the time, the astronomers and physicians of "ancient times (*i.e.* before the 20th century) are labeled as mathematicians on Wikipedia even though there is no mention of any kind of mathematical work in their dedicated page. In fact, roughly 44% of the mathematicians present in our initial list of mathematicians had one of these issues, which means that we managed to establish relations between mathematicians and theories for only 56% of them. Still, this amounted to over 2000 mathematician nodes so we did not mind.

The issue with theorems

Initially, we also wanted to have nodes for theorems so as to link theorems to the mathematicians who had proven them. However, we quickly abandoned this idea for two reasons:

1. hard to retrieve: except for the iconic ones like the Pythagorean theorem, mathematical theorems do not exist as entities on Wikidata (nor DPpedia or YAGO), and although most of these theorems do have entries on Wikipedia, parsing the names of the mathematicians who have proven them proved to be quite challenging

2. not very useful: since most theorems are proven by a single mathematician, adding them to the graph isn't much enlightening (it essentially amounts to adding a `theorems proven` attribute to the Mathematician node)

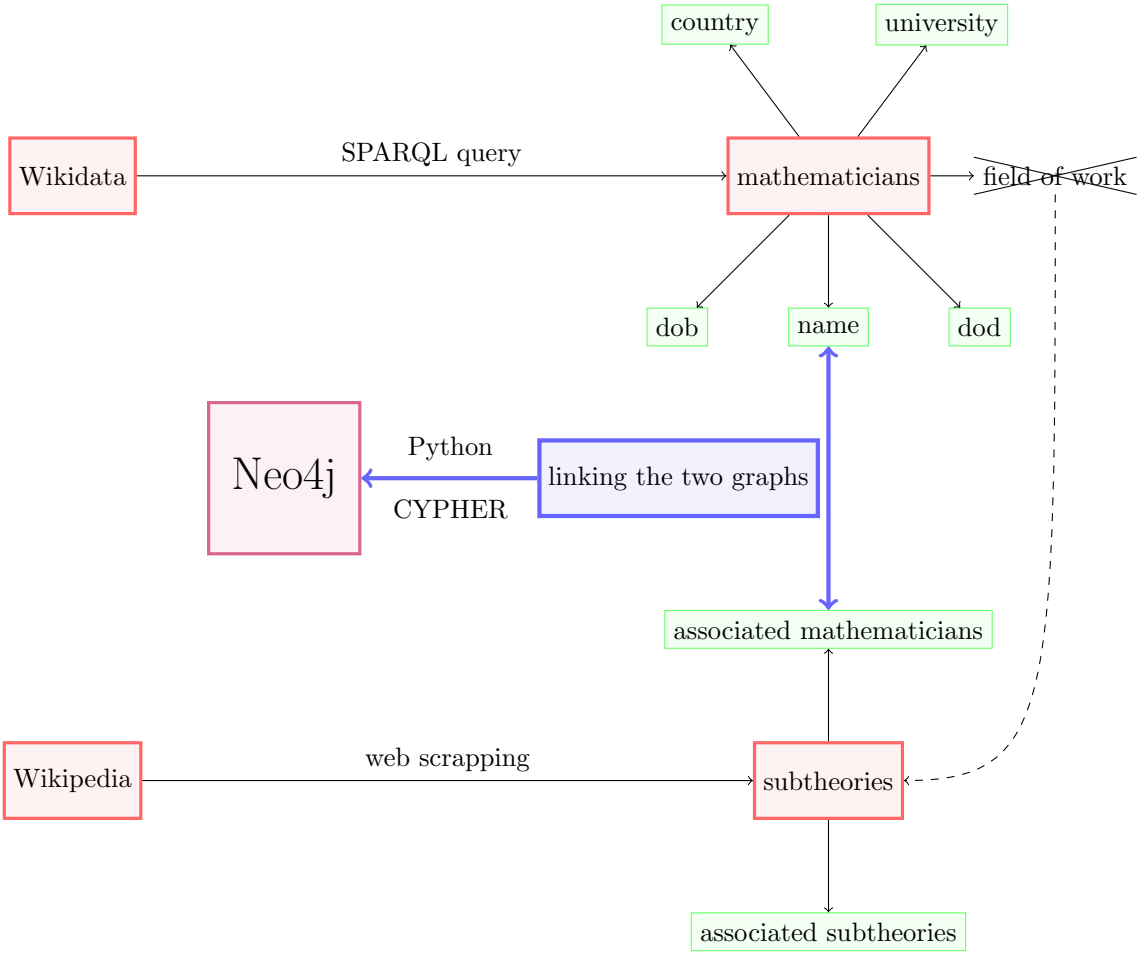

Figure 2: From raw data to Neo4j graph

3 Graph construction

Obtaining the raw data

Querying Wikidata

Using SPARQL to query Wikidata, we were able to retrieve all mathematicians for which we knew their date of birth and date of death, universities of study, fields of work and country. We made doctoral students an optional attribute since some important mathematicians do not have any. Also, note that al-

though we did not use Wikidata's `field of work` attribute, we still requested mathematicians to have that attribute so as to focus on the most famous mathematicians. Thus, out of the total 37324 mathematicians which can be found on Wikidata we filtered out about 90%, ending with 3768 distinct mathematicians and just over 200,000 rows in our export CSV file.

Scrapping Wikipedia

In the glossary mentioned before, each theory links us to another page explaining what the theory is about. At the end of such "theory" pages there is a field named "Category" that gives all the mathematical concepts that this theory is a subfield of. Each category property corresponding to this "Category" field also links to that mathematical category's page, which shares the same structure as before, meaning that we iterate this process recursively until we reach a "mega category" like algebra or until the "Category" field is empty.

Using this handy "Category" field, we could thus for each theory build a list of all theories that it is a subtheory of. And we could do so for all the categories present in the Wikipedia glossary. However, most of the time the categories in the "Category" field were not listed in the glossary (as they are sometimes too specific to constitute an entire theory). To solve this issue, we can go from a theory T to its corresponding categories, and then from each such category to its own corresponding categories, and so on. We continue this recursive process until the "Category" field is either empty or links back to its own category or to very general categories like "science" or "mathematics" which are not of interest to us. During this process, any category that belongs to the glossary will be added to T's list of supratheories.

Practically speaking, we found that we usually reached the endpoint by iterating two times (meaning finding the categories of the categories of the initial theory), so we limited ourselves to a recursion depth of 2 in our implementation of the scrapping algorithm described above. All details can be found in the source code. (in the `theory_theory.py` file)

Likewise, to establish the links between mathematicians and the theories which they contributed to, we go through the Wikipedia page of each mathematician and each time we find an occurrence in our list of theories, we consider that the mathematician has worked on this theory, hence we add it to this

mathematician's list of theories. The implementation of this scrapping algorithm is quite straightforward and can be found in the source code. (in the `mathematicians_theory.py` file)

Preprocessing the data

As expected, the Wikidata CSV was rather clean and did not require much preprocessing. We simply shortened the column names (replacing each column of the form `entityLabel` with `entity`) and trimmed the dates, keeping the year only.

During data exploration, it also appeared that a handful of mathematicians had two or even three birth year or death year. We adressed this technicality by arbitrarily keeping the lowest year, which didn't matter much anyways because the uncertainty was ± 1 year every time.

Building the RDF graph using Neo4j

Once we had our data cleaned up, we were ready to finally build our RDF graph. We used the Neo4j graph database management system to do so.

One issue that we ran into is that we couldn't simply load our big CSV file "as is" into Neo4j because

1. it was too large for our computer and would cause Neo4j to crash
2. it would have created countless duplicate edges because of the structure of our CSV file[2].

To work around this issue, we proceeded in two steps:

1. building the nodes: for each node, we selected the corresponding column in our main CSV file and dropped duplicates, which gave us a list of all distinct entity names (and the corresponding attributes in the case of the Mathematician node), which we then imported into Neo4j to create the entities

[2]Indeed, since our CSV file has several columns, most of which can take multiple different values for the **same** mathematician, projecting along two columns to create edges creates many duplicate rows.

2. building the edges: likewise, for each relationship we selected the two corresponding columns (source and target) in our main CSV file and dropped duplicates, which gave us a list of distinct edges which we then then imported into Neo4j to create the edges between entities

All the details regarding the construction process can be found in the Jupyter notebook.

At last our graph was built ; a tiny sample is shown below.

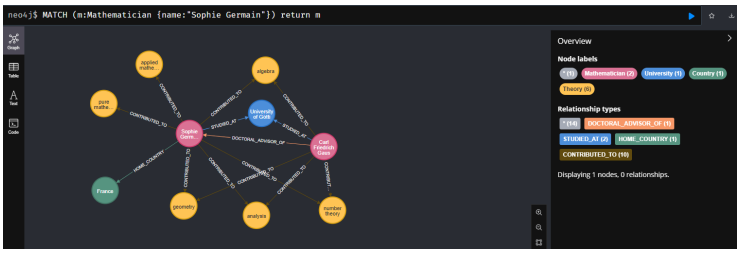

Figure 3: Tiny sample of our graph around the node corresponding to the French mathematician Sophie Germain

4 Analysis and visualization of the graph

Queries

Some natural queries immediately came to mind when we started exploring the graph. Here is a non-exhaustive list ; the results can be found in the Jupyter notebook.

- Which countries have the most mathematicians?
- Which universities have the most mathematicians?
- Which mathematicians had the most doctoral students?

Graph algorithms

RDF graphs can be queried to extract information from them, but we can also run graph algorithms to detect broader patterns in the data. However, most

graph algorithms are designed to work with simple mathematical graphs (*i.e.* graphs with only one type of node and one type of edge), not RDF graphs. Therefore one must project its RDF graph into a simple graph in order to run graph algorithms on it. Of course there are many possible projections to choose from ; we decided to analyze the two graphs that seemed most important to us:

1. the graph of mathematical theories and how they relate to each other
2. the graph of mathematicians and how they are linked to each other through the doctoral student/advisor relationship

We will briefly explain how these two graphs were created ; all details can be found in the Jupyter notebook.

The subgraph of mathematical theories

In building the graph of mathematical theories we wanted to represent the interactions between mathematical (sub)theories. We were expecting to see huge theories like algebra or analysis dominate and have many nodes pointing towards them through the SUBTHEORY_OF relationship ; these nodes would be intermediary theories, themselves pointed towards by even smaller theories, etc.

Our first approach was to simply take our CSV file containing the SUBTHEORY_OF (directed) edges and load it straight up into the Gephi software to run graph algorithms on it and to visualize it. Here we quickly summarize the results. First, note that our graph's family tree structure makes it unfit for certain graph algorithms, such as PageRank or betweenness centrality. We did attempt running these algorithms and obtained poor results as expected. Second, running modularity inference on our graph proved quite successful, as the visualization below showcase. However, the algorithm was not able to cluster theories into large domains like algebra and instead created smaller clusters, like computational mathematics. We thus ended up with 63 communities and a modularity score of 0.477, which is average. Also, graph density is very low (0.008)[3] and we have a few lone nodes which correspond to niche mathematical theories such as systolic hyperbolic geometry, stratified morse theory, bolyai-lobachevskian geometry, etc. For visualization purposes, we set node size to be

[3]Again, this is expected given the family tree structure of our graph, which forces nodes to "go down" the tree and prevents reciprocal links.

proportional to in-degree (meaning that the more subtheories a theory has, the bigger it will appear) and we colored nodes by modularity class. In Figure 4 we used the Force Atlas layout and in Figure 5 we used Fruchterman Reingold.

We felt like this first approach was a bit naive and that we could do better, especially regarding community detection. We thus created another graph, with the same Theory nodes but this time weighting the (undirected) edges $T_1 \leftrightarrow T_2$ with the number of mathematicians who had contributed to both theories T_1 and T_2. We queried our graph using CYPHER to obtain the corresponding CSV file and then loaded it into Gephi. This time the results were a lot more satisfying. Indeed, the modularity inference algorithm detected four classes only (compared to 63 with the naive graph), which we identified as algebra, analysis, geometry and combinatorics, as shown in Figure 6, were we manually separated the 4 classes so as to distinguish them clearly. Overall modularity is 0.136, which is rather low as expected because mathematics are not clustered : all domains are linked one way or another. However we were quite surprised to find combinatorics along algebra, analysis and geometry, which are much larger domains. We thought it was some sort of algorithmical artifact and we tweaked the modularity algorithm slightly (using a resolution of 1.05 instead of 1) so as to end up with only 3 classes. This time combinatorics disappeared and we ended up with the algebra/analysis/geometry triad, as hoped! Again in Figure 7 we manually separated the 3 classes to see them distinctly.

The subgraph of mathematicians

To build this graph, we only kept mathematician nodes and we used directed edges. The edge $m_1 \to m_2$ means that m_1 is a doctoral advisor of m_2. We performed this projection using the appropriate CYPHER query and then exported the results into a CSV file that we loaded into Gephi. For visualization purposes, we set node size to be proportional to in-degree (meaning that the more doctoral students a mathematician has, the bigger it will appear) and we colored nodes by modularity class. Figure 8 was thus obtained using Yifan Hu layout, which is ideal to appreciate the tree-like structure of our graph.

We immediately see three interesting patterns:

1. the modularity algorithm was able to cluster mathematicians into lineages where a mathematician's "parent" is its doctoral advisor

2. most lineages have common ancestors (corresponding to the large nodes in the center of the graph), *i.e.* ancient mathematicians whose students became prominent mathematicians themselves, and so on

3. some lineages (the ones on the edge of the circle) seem to be independent ; this could be due to a lack of data (*i.e.* we're missing some edges) or to the fact that some mathematical communities are most closed than others (e.g. the Japanese for a long time)

The modularity score for the overall graph is 0.906, which is very high. This is coherent because as we've seen, mathematicians are clustered into lineages which don't interact much with each other.

Another interesting metric with such a graph is eccentricity, which is defined as the distance from a given starting node to the farthest node from it in the network. Since edges are directional, a node's eccentricity is simply the length of its lineage, starting from itself. The eccentricity histogram below shows that most mathematicians have no doctoral students or just one. This is probably because few mathematicians actually have more than one doctoral student. However it's a bit surprising that most mathematicians have no doctoral student at all. Our hypothesis is that most do have at least one doctoral student, but these students are not famous enough and therefore do not appear on the graph as they do not exist on Wikidata.

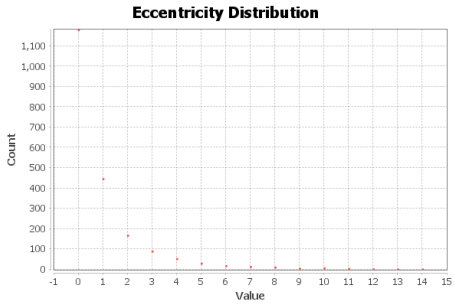

Figure 9: Eccentricity histogram

Note that the graph diameter (defined as the longest path from one node to another) is 14. This is quite impressive, as it means that there is a lineage composed of 14 mathematicians. We can find the corresponding path(s) using the appropriate CYPHER query and visualize the result directly in the Neo4j

browser.

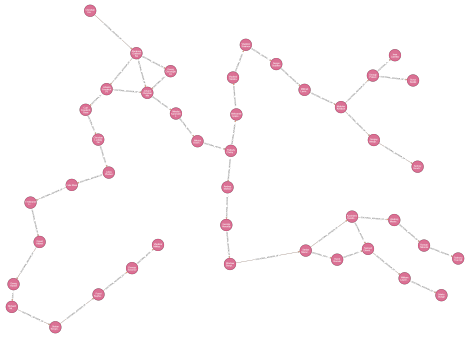

Figure 10: There are several longest paths, with the same starting nodes but different ending leaves

We see that this lineage starts with Christian August Hausen (top left node in Figure 10), a German mathematician and physicist born in 1693, and ends in several leaves, including that of Marta Bunge, an Argentinian-Canadian mathematician who died last year, in 2022! We also see that some very famous mathematicians are actually "grandparents" of other very famous mathematicians through the doctoral student/advisor relationship. In that regard, Gauss is Felix Klein's great-grandfather, and Klein himself is David Hilbert's grandfather!

One additional question that comes to mind naturally is to determine the size of Christian August Hausen's lineage, *i.e.* the number of mathematicians who count him as their ancestor. We can figure this out quite simply using another CYPHER query, and we find 978. In other words, over one quarter of the mathematicians in our graph have Christian August Hausen as their ancestor!

Finally, if we look at the number of connected components we find 112 separate connected components. The histogram belows confirms our intuition that there is one massive mathematical community with over 1600 nodes, and then a handful of much smaller, isolated lineages with no more than 50 nodes.

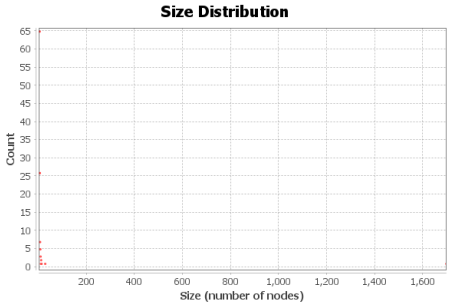

Figure 11: We see one huge connected component (tiny red dot at the very right of the histogram) and a handful of tiny ones which are independent

Also, note that again our graph's family tree structure makes it unfit for certain graph algorithms, such as PageRank or betweenness centrality. We did attempt running these algorithms and obtained poor results as expected.

References

[1] Armand Boschin. "WikiDataSets : Standardized sub-graphs from WikiData". In: *arXiv* (June 2019).

[2] "Glossary of areas of mathematics". In: *Wikipedia* (2023). URL: `https://en.wikipedia.org/wiki/Glossary_of_areas_of_mathematics`.

[3] Brian Heater. "Alexa gets access to Wolfram Alpha's knowledge engine". In: *TechCrunch* (Dec. 20, 2018). URL: `https://techcrunch.com/2018/12/20/alexa-gets-access-to-wolfram-alphas-knowledge-engine/`.

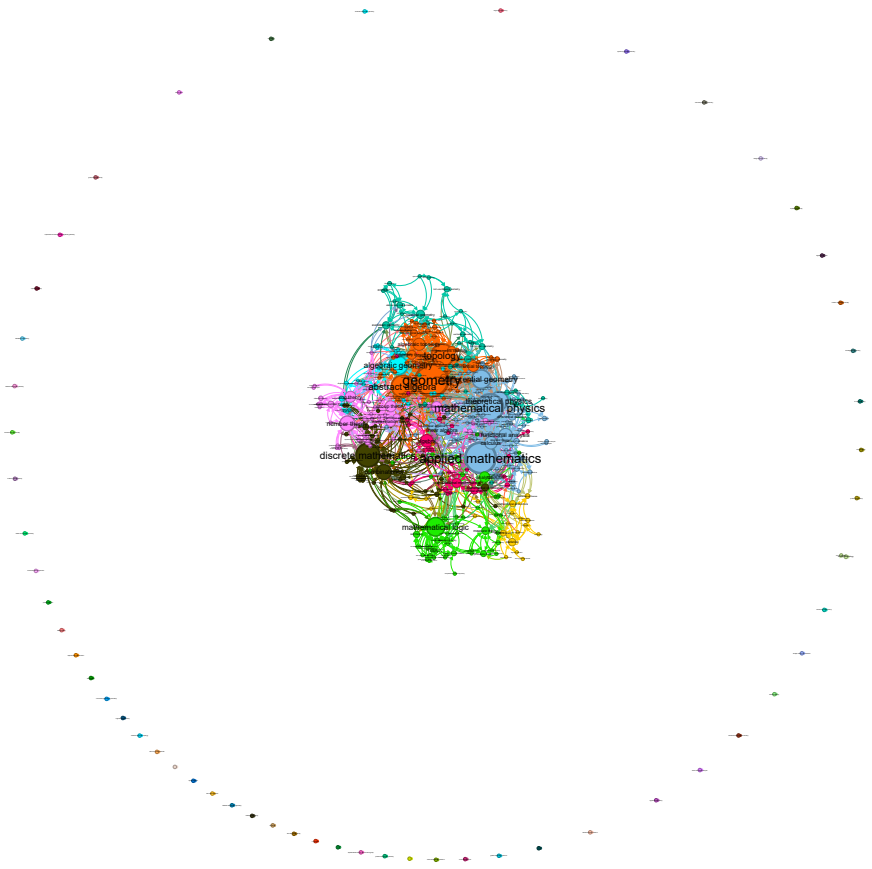

Figure 4: Mathematical theories' graph visualized using Force Atlas layout

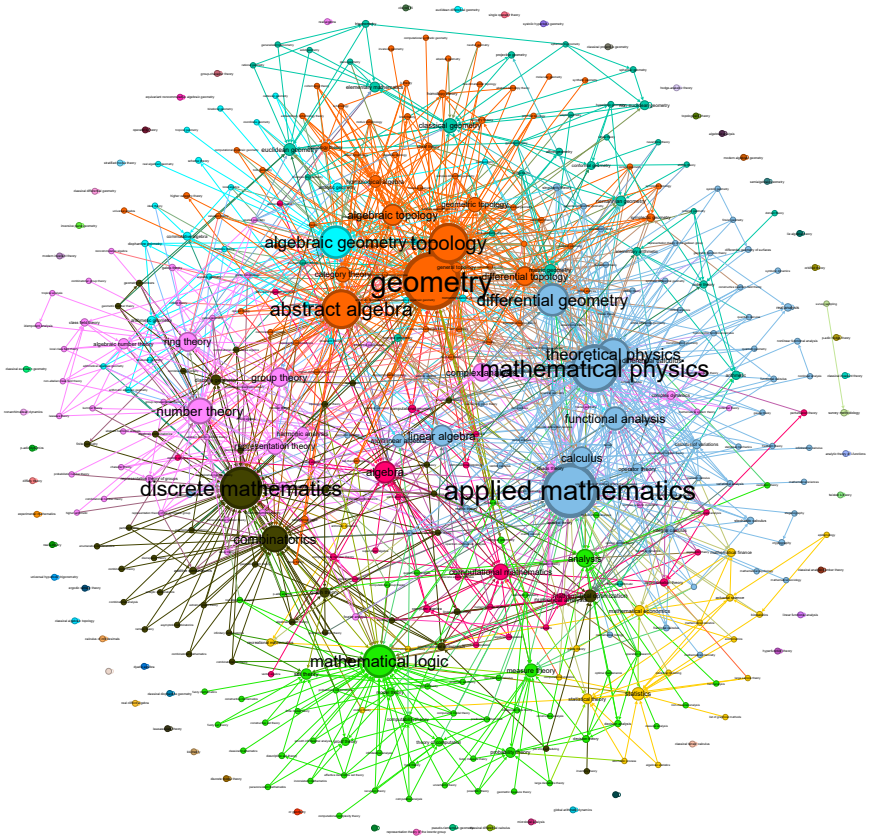

Figure 5: Mathematical theories' graph visualized using Fruchterman Reingold layout

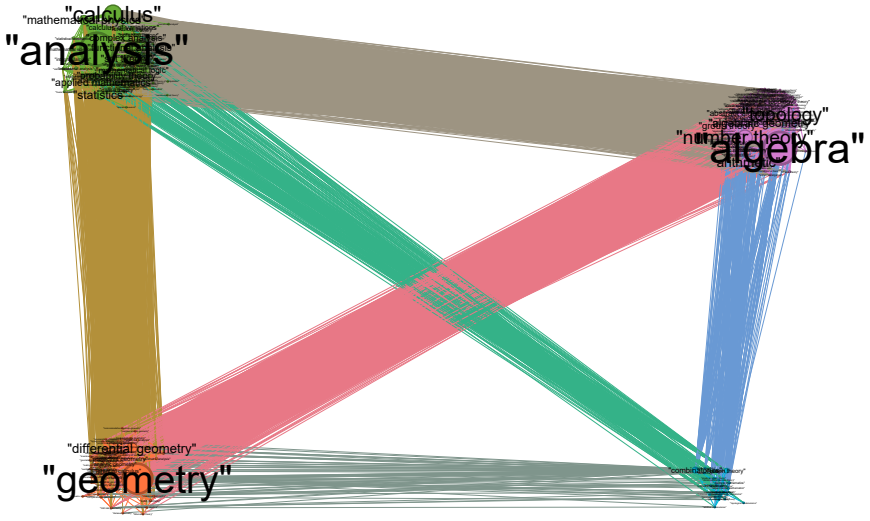

Figure 6: Mathematical theories' graph split into 4 modularity classes

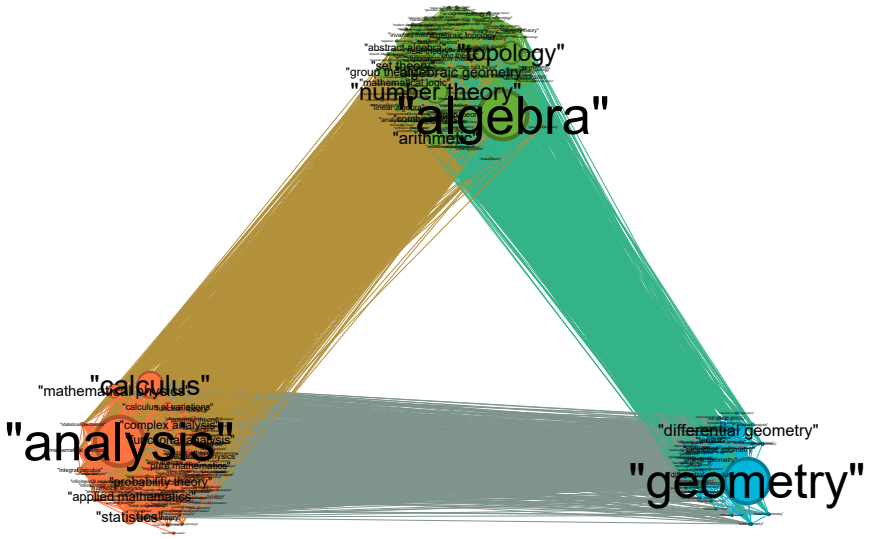

Figure 7: Mathematical theories' graph split into 3 modularity classes

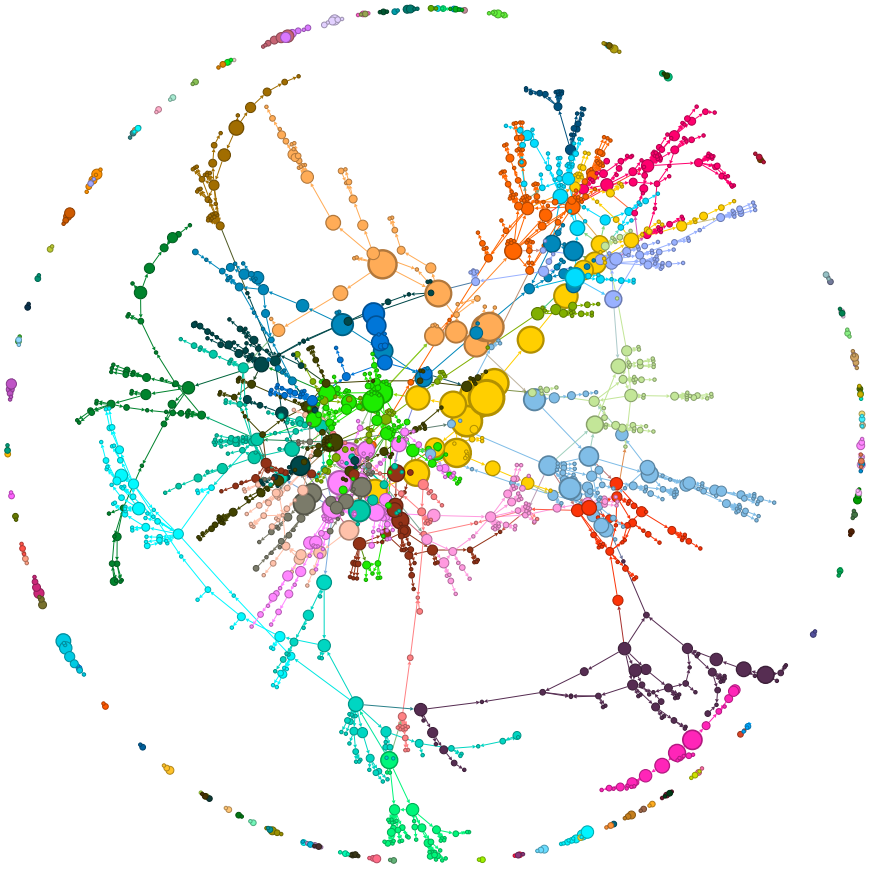

Figure 8: Mathematicians' graph visualized using Yifan Hu layout ; names were omitted for clarity