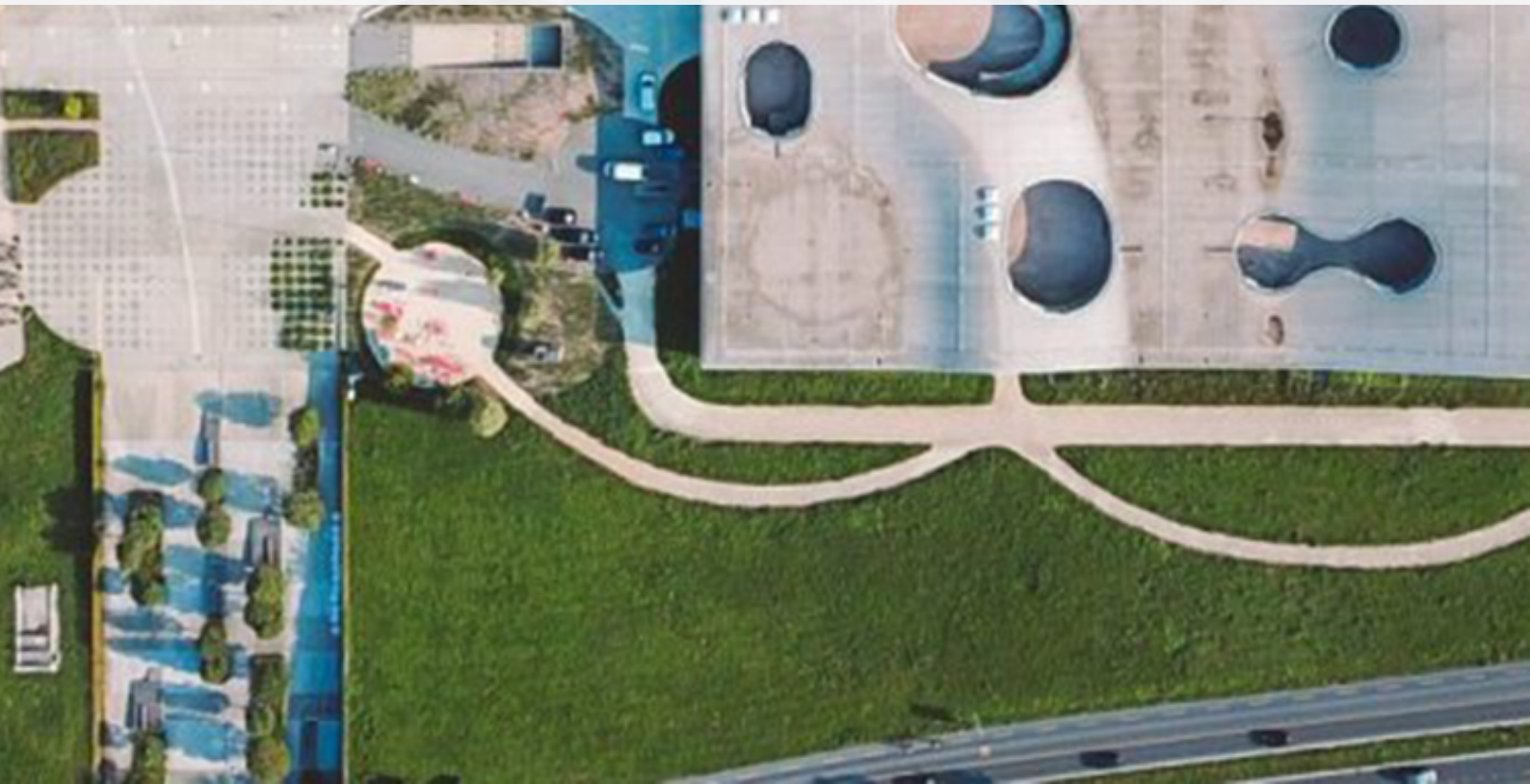
ISN'T-ACADEMIA

A website to help students to select courses at EPFL

Costanza Volpini
Eleonora Rocchi
Wei Jiang



Contents

1. Introduction

Meet Bob. He is a computer science student starting his master at EPFL; he is really happy of his admission but he is also scared. What is waiting for him? How will he choose his study plan?
EPFL offers a wide range of courses, but he is not sure what will fit best his interests. He would really like to know more about the experience of other students, which choices people from his major did in previous years? This information is actually made available from EPFL but it is accessible in a really database-like format which doesn't help the investigation Bob wants to conduct.

1.1 IS-ACADEMIA public access

Is-academia public access offers a wide range of information from previous years: the list of student for each section by semester, the list of available courses and the students enrolled; even past exam's timetable by student.
All this information are **publicly** available and can be **freely consulted by anyone** who is looking for them. But here lays the problem: what the user should look for, to be able to extract useful information?
The current visualization format makes difficult to aggregate those data and extract information which may be useful to a student who is trying to choose courses for next semester. This is partially due to the fact that this is probably not the main intent with which those data are made available but, nevertheless, they constitute a gold mine of information for the user if they are provided through a different visualization.

1.2 Our idea: ISN'T-ACADEMIA

In the past, we personally used *'Is-academia public access'* to obtain information regarding some specific course: "Is it true that the Machine learning course is so popular nowadays?" or "Which courses did student X pick last year? We are in the same specialization, I may benefit from his experience".
Now, a bit of manual labor is required in order to recover more complex insights about a course and, this may discourage most users from even start exploring the data.
In this context, we decided to collect and reorganize this data set to provide a useful tool to our fellow students who will be choosing exams for their study plan next semester.
We articulated our work in order to answer two main questions:

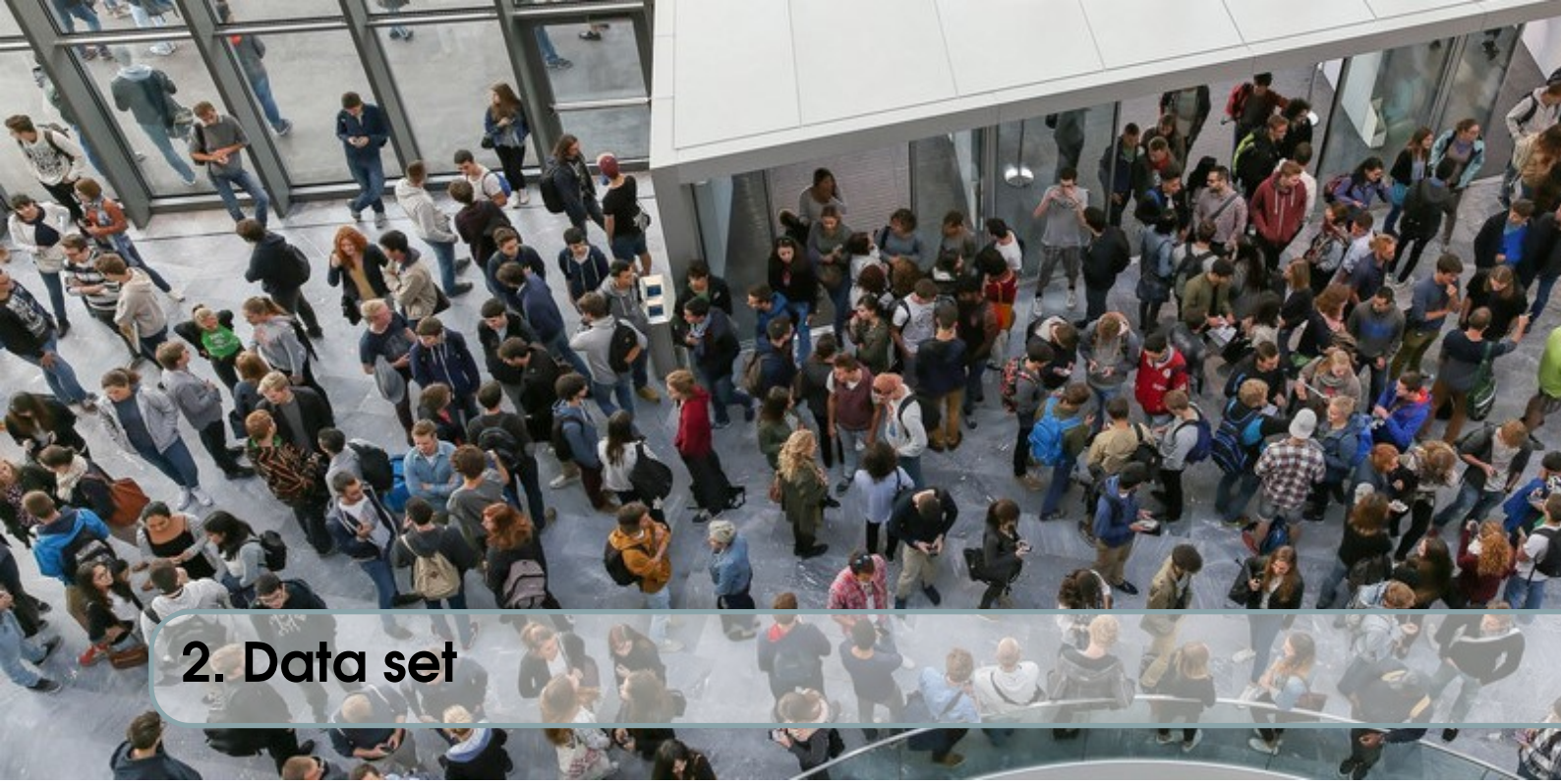
1. *Which courses are the most popular ones?*
   Each research area has some "hot topics", which shape their field. Sometimes, it may be interesting to take a course related to those topics even if they do not fit best our interests: knowledge about them is probably expected in any work/research environment. An example could be the 'Machine learning' course; it is one of the courses with the highest number of students and it is considered by many CS students a "must" in their study plan.

2. *Which courses should I choose for my study plan based on previous students experience?*
   Leveraging older student experience is, surely, an important skill of any seasoned student. Everyone is looking for a different experience during his studies and, others opinions can be quite insightful. Our data set does not provide any qualitative information about students feedback for courses, but "reputation" and "spread word" strongly influence people's decision. Therefore, while we can not directly see what those are, we can observe "trending" among students enrollment and use them to suggest our users.

All those considerations do not take directly into account the course's content and addition to the student skills. Nevertheless, we took into account some practical aspects which may be really insightful. So, we build *ISN'T ACADEMIA* where the name contraception with the original website name underline our student-like approach to course's choice, contrasted to a more course-book based one.

1.3 Peer assessment

In the whole process we retain that all of us have contributed equally to the project keeping a clear view of the overall. It was exciting to work on different parts to build our website. We enjoy our discussion and learn a lot from the project.



2. Data set

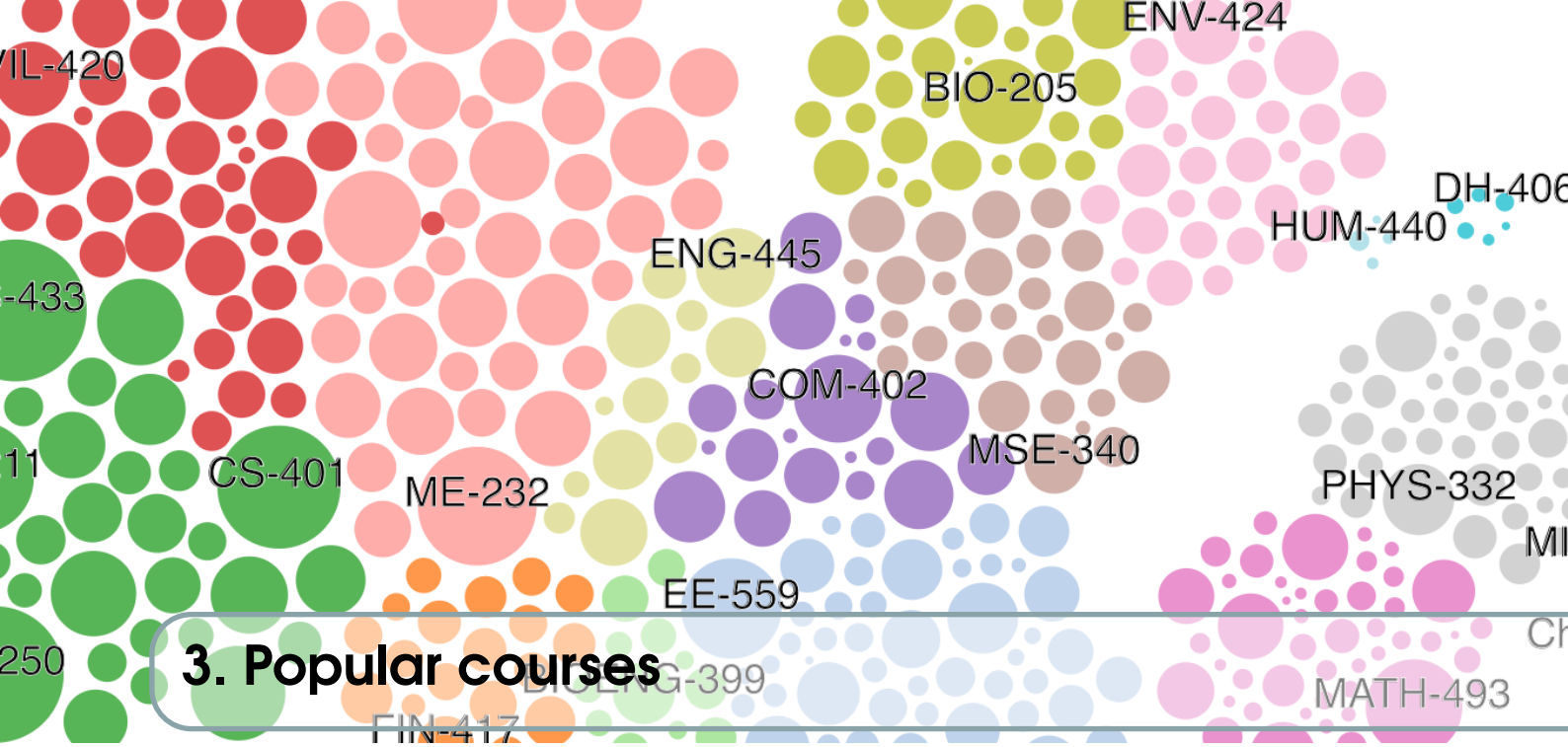
Our main data set comes directly from EPFL and it is publicly available at Is-academia public access for courses and Is-academia public access for sections.
This data set contains the list of enrolled students for each course offered by EPFL from the academic year 2004-2005. The website allow to lookup the students who were enrolled in a specific course allowing the user to select the course name and the academic period.

2.1 A first look at our data

Our data set has been generated by crawling the original website. The collection of the data took around 7 hours and it allowed us to recover 6995 excel files. The crawler, developed with JavaScript and python, try to recover all the documents available on the server generating all possible courses identifiers and inserting them into a standard query to recover a single course's document. We have been able to crawl all the bachelor and master courses offered by EPFL from 2004 until now.
After the collection phase, we had to parse our data set: first, we selected only the files which contained information about enrolled students (e.g. ignoring timetables of classes), then we analyzed them parsing the nested HTML structure.

2.2 Data reorganization for faster performance

After obtaining a first version of our data set, we realized that while this format is very general and allow to perform any operation needed with our data set, it is not the best way to maintain data for our visualization. For example, in order to recover the most popular courses we needed to sum the number of students in each course every time. We realized this is not needed as we are only interested in the aggregated data and we never use the singular entry in the enrollment table. Precomputing the number of student for each course grouped by major and semester, allows us to perform any operation we need for our website without creating a bottleneck performance on the server side.



3.1 A first sketch

The first attempt of bubble diagram is to show the top N popular courses in one major, like Computer Science to give students an overview of popular courses. As showed in figure 4.3, the number of enrollments is expressed by the legend of the shape of the bubble. The plot does not include much information and no interactions at this time. So in the following milestones, we give trials to show more contents and generate some interactions with user to give them more customized information.

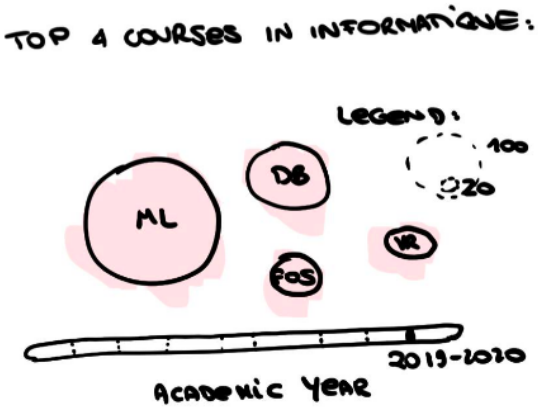


Figure 3.1: Primary idea for bubble chart in milestone-1

The plot allows more interaction in milestone 2. We came up with an idea to allow user to select the courses they want to see and delete the one they do not like. By clicking the bubble, the statistics regarding to the course will expand. It will present more information about the course trend and students' major intake of the courses.

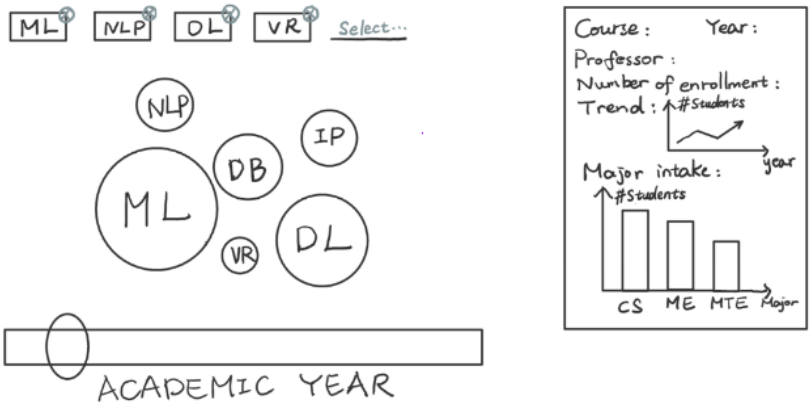


Figure 3.2: Improved idea for bubble chart in milestone-2

3.2 Final results & changes

As showed in figure 3.3, our final plot changed significantly from the first scratch. The main change is in two aspects. First, instead of only showing the popular course in one section, we try to show the most popular course in each section. Besides, we let users to know about other courses. By clicking the bubble, the information for the course, such as the professor, course trend, course major intakes in every year will be showed. We tried our best to place the bubble clusters and make the plot clear to read. From the size and number of bubble, users can get an overview of course information in each section. Instead of inserting course names in milestone 2, the checkboxs for the selecting section is used.

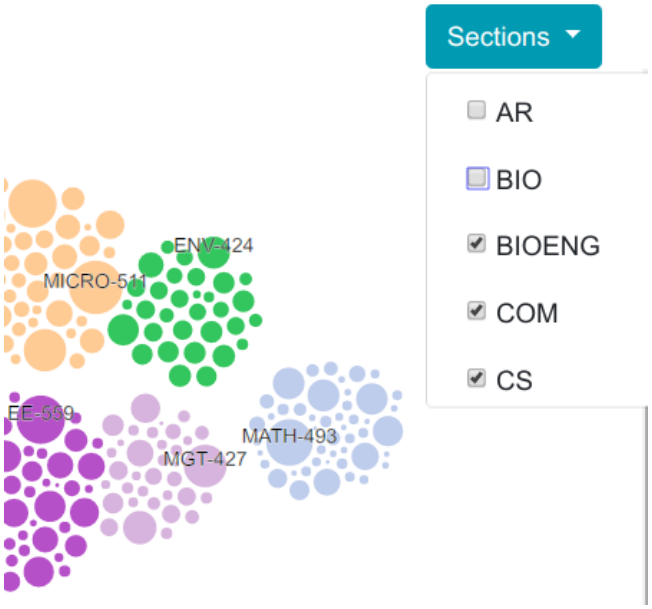


Figure 3.3: Dropdown and checkbox for selecting the sections

3.3 Building the bubble diagram

The first step is to collect and clean the data for the diagram including the information of sections, courses and number of enrollments. Then we used the forced cluster diagram in d3.js to present section clusters. A sliding

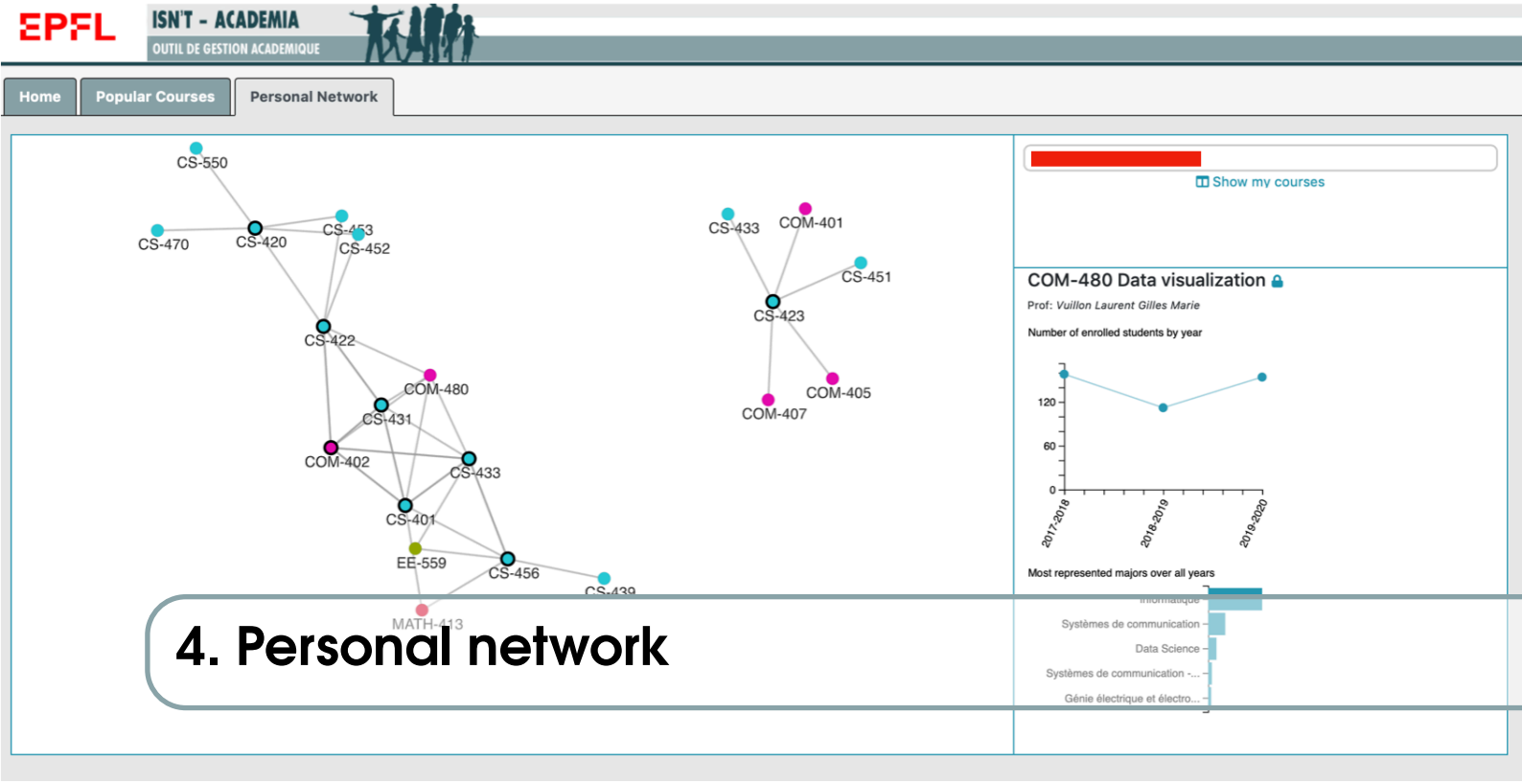
bar for selecting year is put under the bubble diagram. The section selection is putting in the top which can be hided and expanded if users need to choose the sections they want to take a look. The statistics about number of course in the sections and the average number of student enrollments of the section appear automatically after the users choose the year in the sliding bar. This information is consistent to the section they choose. By choosing the radio button, the user can switch between the information about number of courses and average student enrollments. By clicking the bubble, the professor, trend of enrollments of the selected course and major intake is showed. We show the most popular course id in side the bubble in each section as our legend. Besides, the course name with the number of enrollments above a threshold for some sections like computer science with many popular courses are also showed.

3.4 Implementation challenges

When showing the bubble cluster in each section by using force graph in d3.js, it came to the problem of showing the bubble legend. It is unreadable if we label each bubble with a text inside. Also putting the legend list it not smart for long list with 20 category. So we decided to label the bubble which has the largest number of student enrolment in each section. In this way, we could show the legend and also the most popular courses in each section.

In terms of statistics, the problem is to integrate plots as we have many statistics to show. We wanted to show the information about both section and single course which is difficult to design a good layout. Our design is as the follwing. By default we showed the section information, then switch to statistics for single course by clicking the bubble the graph. However, for some very small bubbles, it is difficult to click so we expand the bubble size when user's mouse move over the cluster.

Instead of plain checkbox for choosing the sections, we combined it with "dropdown" button since our list is very long. When the users click the checkbox, the bubbles belonging to that section appear and on the contrary, disappear when they uncheck it. Correspondingly, bubble diagram will take actions to add or delete the clusters. Also this will pass to the statistics plot to give a corresponding statistics.



The most powerful information contained in our data set is the exact list of courses each student has been enrolled for. This allow us to suggest new courses for each student without asking him any personal information other than his name. Also, a students can look up the study plan of different friends and see whether there are common interests.

4.1 A first sketch

Our first idea was to create a network in which the courses would represent the nodes and the connection will depend on course's similarity. Given a student name, we wanted to visualize the portion of the network which contained his courses and the immediate suggestions. In our first milestone we presented the following sketch:

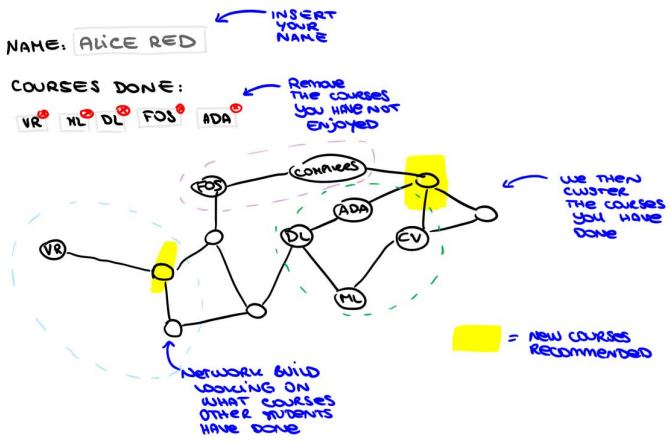


Figure 4.1: First sketch of the course network

The implementation of this personal network has been challenging for multiple reasons. First, we met some problem while deciding the topology of the network and then regarding some functionalities we added to expand/reduce the network.

4.2 Build the network

Define the topology of our network can be, at first look, like a matter unrelated to visualization, but we discovered that we had to be vary careful in the way we built our graph. The personal network did not have a fixed structure, we decided to pick as measure of affinity between two courses the Jaccard similarity between their sets of enrolled students. Then, we had to find a benchmark between the number of recommendations and our ability to effectively visualize the network. In fact, we encountered the classical problem one encounters when he wants to visualize a graph: it was not planar and its visualization was quite messy because of crossing edges. Therefore, we had to carefully model some forces to create repulsion between links without separating too much different connected components. (Fig. 4.2)

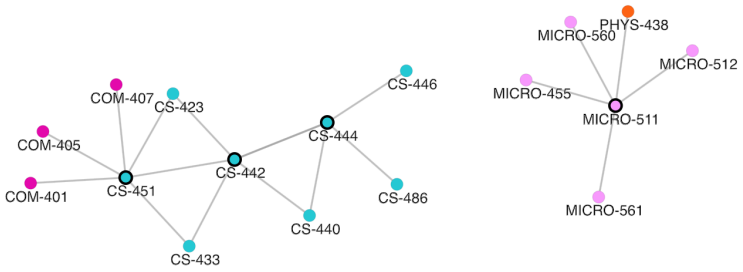


Figure 4.2: Wei Jiang's personal network: we can see the locked courses have a wider stroke. Courses are naturally grouped in cluster from similar sections.

4.3 Navigate the network

Another matter of great importance was how to put the course's name on the nodes; make a node recognizable has a great impact on the ability of navigating the network as the user has a clearer understanding of the meaning of transformation he is visualizing. Here, we decided to use a solution similar to the one proposed for the bubble's cloud in chapter 3: visualize the short course name provided by the section and show only on mouse over the full name of the course. Also, being able to retrieve the section allowed us to color courses nodes accordingly to their section; this approach naturally created clusters of nodes of the same color (Fig. 4.2) Last, we increased the ability of the user to navigate our network by allowing him to block/unblock some courses depending on his personal feedback; this would allow him to model his own network starting from the default one we proposed him. (Fig 4.3)

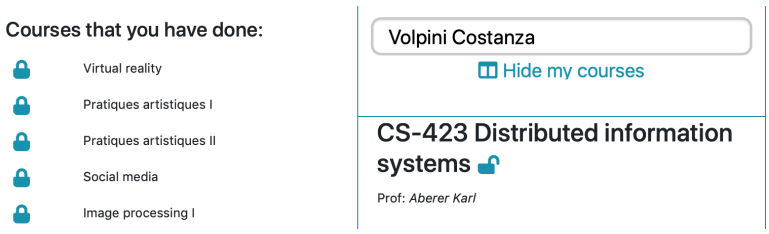


Figure 4.3: Sample of personal network locks: Costanza's courses are listed on the left and they are all locked, the course of which she is looking at statistics is instead unlocked