

VIS Milestone 4 - Final Dashboard

Motivation

Students do not study in the way/order the study is designed, but instead take individual routes/paths through their studies, some with more success, some with less. The overall question is: how do students study? But more refined subquestions of that overarching question are:

- Can we find similarities in the way students finish their studies?
- Can we find patterns that tell student who will successfully finish apart from students that will fail/drop out?
- Are there multiple viable ways of finishing?
- Do clusters of similar paths form, and if yes what characterizes these clusters?

The targeted user for this application is in charge of teaching in a faculty of the university. He has an in-depth knowledge of courses and programmes offered at this faculty. Example users include the dean, vice-dean and head of study programme, alongside others concerned with the quality of teaching in the faculty.

Currently this application is working with the dataset from the Faculty of Computer Science, of the University of Vienna. This dataset consists of individual, not connected events, such as “Student X received grade in subject Y” or “Student X discontinued Study Y for reason Z”. The events are directly exported from the database of the university and have to be heavily restructured to be usable for further analysis.

Related Work

In the implementation of this tool individual paths are shown using lines. M.Oppermann, et. al [1] also deal with selecting interesting shapes out of a large number of lines. They chose an interface which lets the user draw an outline on top of the graph showing all lines, which then enables the user to choose only lines with a similar shape then the drawn one. This approach was not feasible for this application, since the lines are not of equal length and do not contain many data points, which makes it very hard to draw using mouse gestures. I chose a tabular view with filters to depict all paths and sort them by relevant features, which I enhanced using Sparklines [3] to easily see the paths.

M.Raji, et. al [2] do an analysis paths through minors/majors at the University of Tennessee, where they show taken paths in a radial tree map, where nodes in the tree are different subjects, layers show the semesters and the width of the line connecting two nodes is the number of people progressing that way. This is helpful for insights in how many students there are within a major/minor and the number of students that switch majors, but this tool focuses on paths within a single field of study (respectively major) and not the university-wide overview.

Approach

Overview first - Browse and filter - Details on demand
→ in two levels!

A dataset of more than 2000 individual paths, consisting of 100k individual events cannot be easily grasped. It is important to get an overview of all paths first, which is realized in a table that is enhanced with sparklines. This table is sortable and filterable to reduce the amount of paths to a set of interesting ones. Once it is reduced enough all relevant paths can be selected to then be further analyzed.

By clicking the analyze button the interface changes and a comparison view displays the distance of the selected paths. Distance can either be shown as a scatterplot or as a heatmap. All graphs are interconnected by brushing and linking. Selected paths in this view now get shown in great detail, down to depicting single courses.

This concludes the two level analysis interface - overview of all paths with easy to compute filter metrics shown, and a comparison view (which is more effortful to compute) showing only a subset of relevant paths.

Implementation

Server side:

An Ubuntu 16.04 server, running Apache2, which is used for Port-Forwarding to an internal Node.js server. The Node.js server is hosting an Express simple http server, on top of which a Socket.io instance is running.

Client side:

The client interface is realized in HTML and JavaScript, used for this purpose are the following libraries: Socket.io client for communication with the server to exchange data, D3 for drawing all plots, Bootstrap for styling and form functionalities, as well as jQuery to enhance Bootstrap's functionality and the overall user experience. The table from the filter view is enhanced with functionality using the DataTables extension, and the range-sliders are created using the bootstrap-slider addon component.

Links to all above mentioned projects can be found in the Reference section.

Issues:

A big issue was the loading time of the project. It is currently working with a dataset of almost 100.000 events, which have to be aggregated, analyzed and enriched to give insight into the dataset. This process was done on the client side in the first place, which took (using a fairly high-end machine) more than a minute before anything was shown. This whole aggregation and enrichment process is now happening on the server. The server takes almost 3 minutes to process everything, but once this is done it keeps the already processed data in memory and the website only takes the time to receive the data from the server and draw it.

Results

Filter settings:

- No. Semesters: [Slider]
- ECTS: [Slider]
- Max ECTS in one Semester: [Slider]
- Mean ECTS/ Semester: [Slider]
- First Semester: [Slider]
- Mean Grade: [Slider]
- Median Grade: [Slider]
- Status:

Show entries

ID	StudyID	No. Semester	ECTS	Max ECTS in one Semester	Mean ECTS/ Semester	First Semester	Mean Grade	Median Grade	Status
0	521	6	180	100	30	2013S	3.14	3	beendet
7	521	1	4	4	4	2016W	4	4	offen
10	521	11	176	45	16	2009W	2.96	3	beendet
11	521	3	8	8	2.67	2015S	2.75	3	offen
12	521	4	107	35	26.75	2015S	2.55	3	offen
13	521	4	23	14	5.75	2010S	3.17	4	beendet
17	521	10	191	32	19.1	2010W	2.81	3	beendet
18	521	9	160	30	17.78	2012W	2.44	3	offen
20	521	7	171	38	24.43	2006W	2.52	2	beendet
23	521	1	8	8	8	2016W	3	4	offen
29	521	2	6	3	3	2007S	2.5	3	beendet
36	521	12	196	29	16.33	2008W	2.87	3	beendet
38	521	9	179	48	19.89	2007W	2.65	3	beendet
39	521	9	180	30	20	2010W	2.53	2	offen
40	521	3	24	16	8	2015W	2.88	3	offen

Showing 1 to 15 of 2,050 entries

Previous 2 3 4 5 ... 137 Next

Image 1: The filter view, where all available paths are displayed in a table to sort and filter.

Filter bar settings:

- No. Semesters: [Slider]
- ECTS: [Slider]
- Max ECTS in one Semester: [Slider]
- Mean ECTS/ Semester: [Slider]
- First Semester: [Slider]
- Mean Grade: [Slider]
- Median Grade: [Slider]
- Status:

Image 2: Filter bar

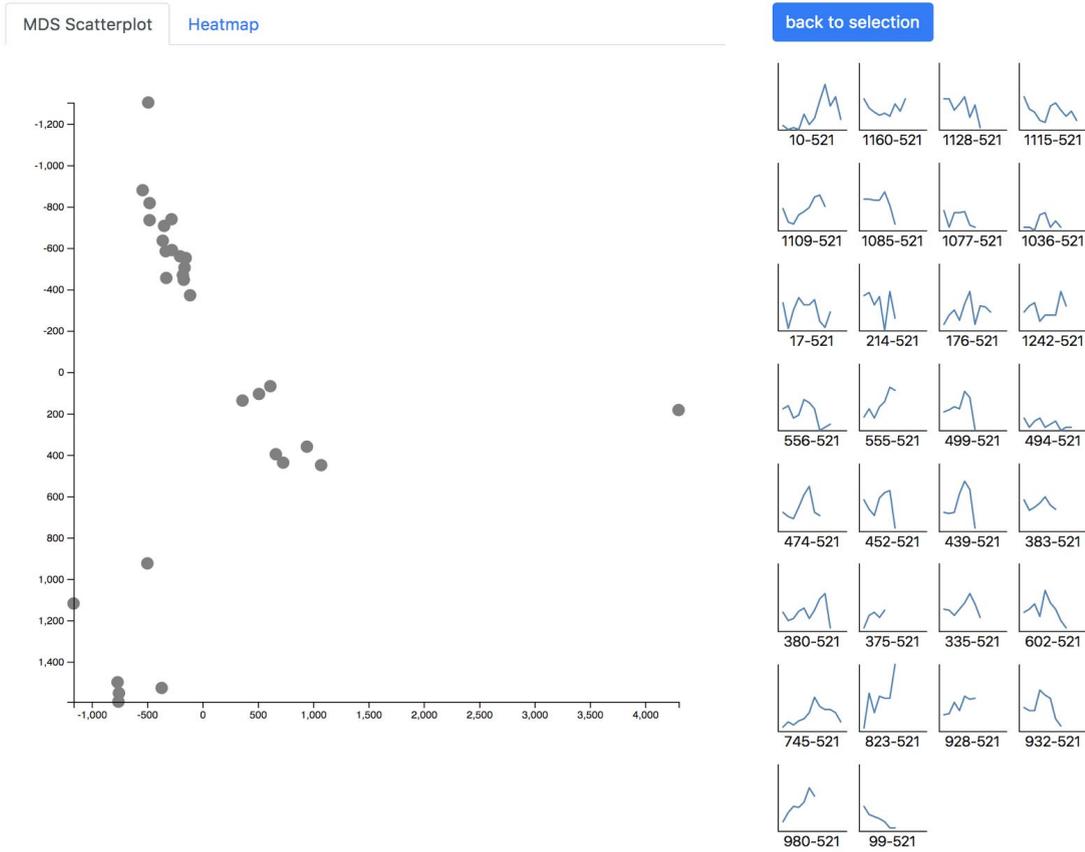


Image 3: The exploration view (scatterplot chosen per default)

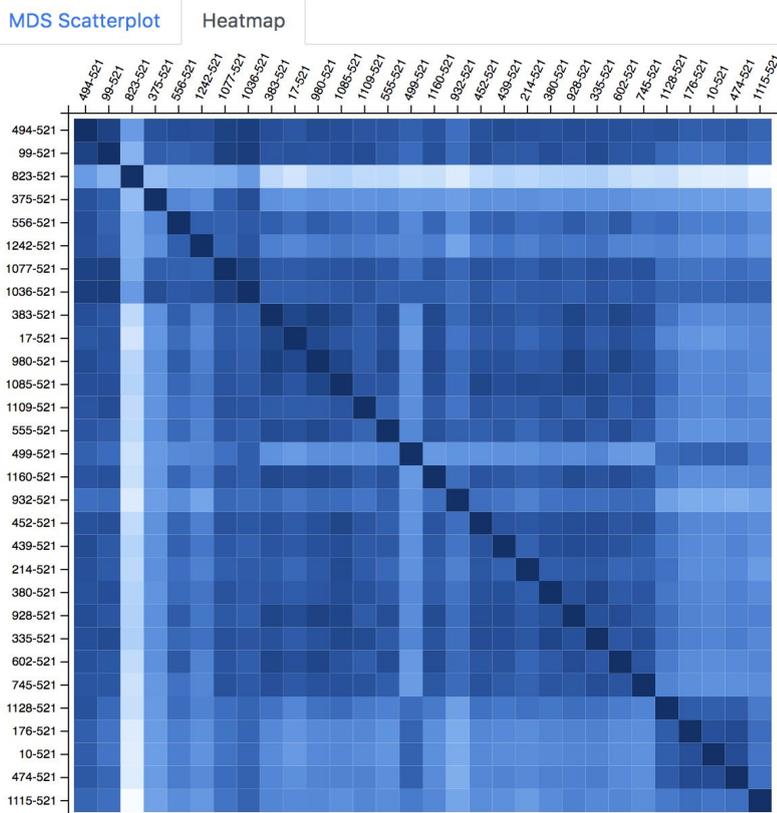


Image 4: The heatmap view

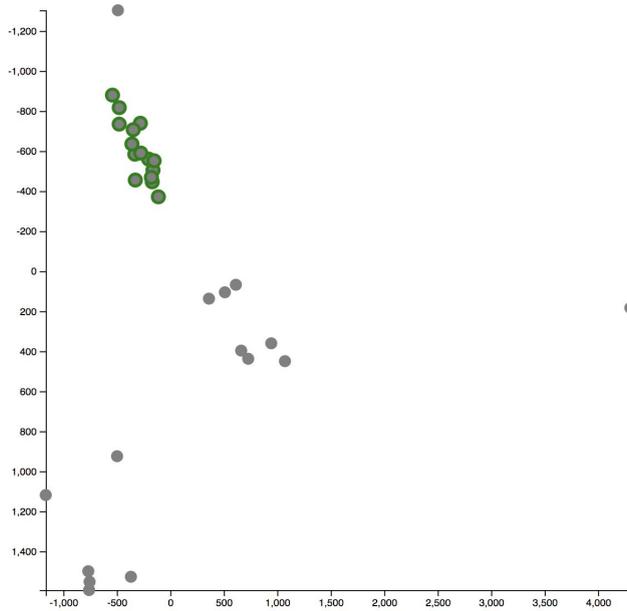


Image 5: selecting in the scatterplot by either clicking on the dots, or drawing a circle around multiple dots (lasso not shown)

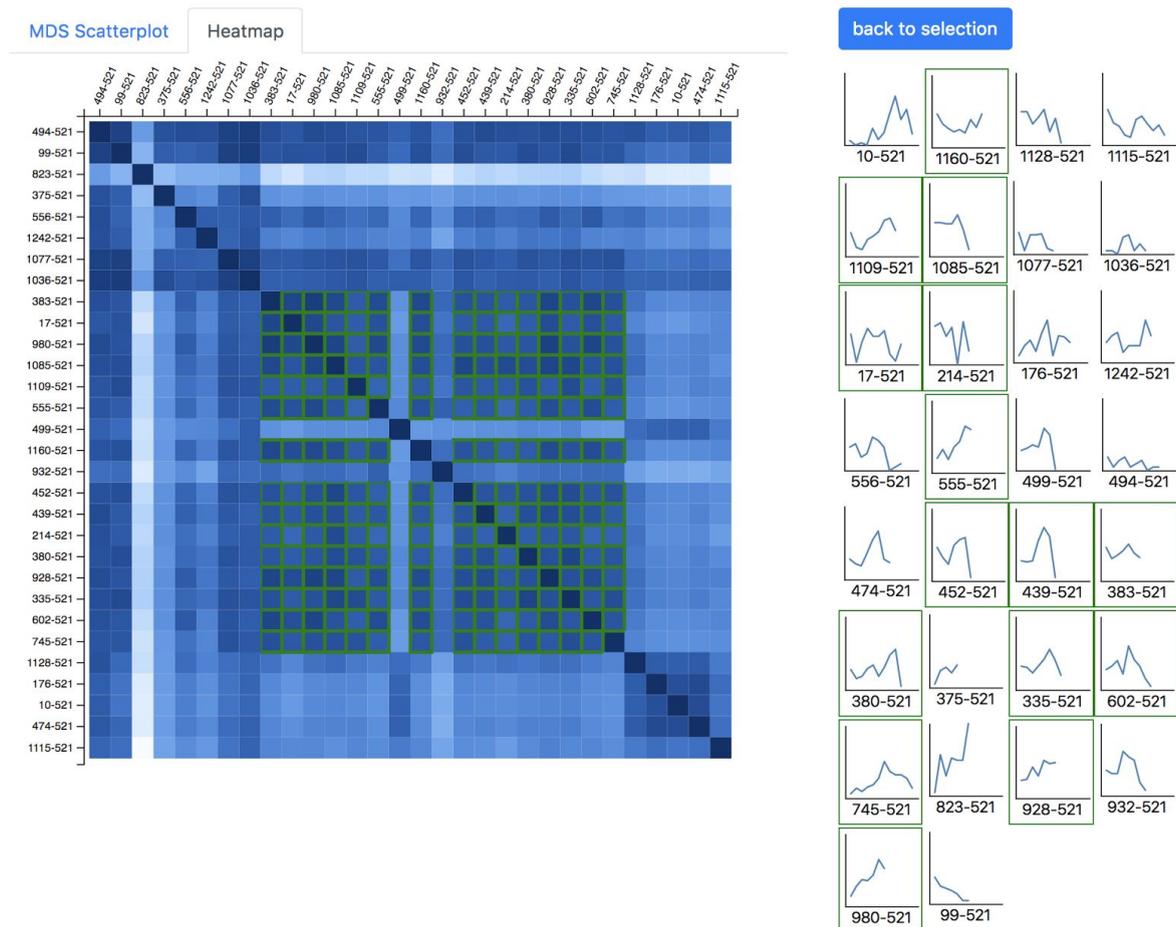
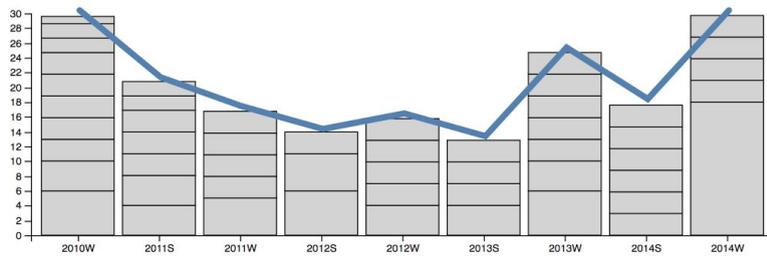
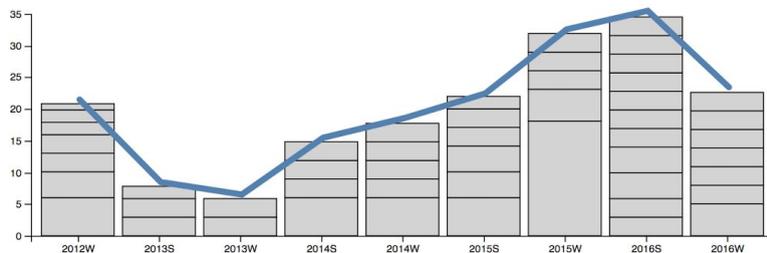


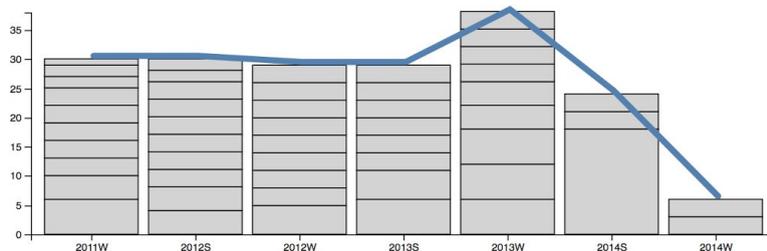
Image 6: Highlighted items can be chosen in the scatterplot, the heatmap or the small multiples on the right hand side - highlighting is linked across these views.



ID: 1160
StudyID: 521
No. Semester: 9
ECTS: 184
Max ECTS in one Semester: 30
Mean ECTS/ Semester: 20.44
First Semester: 2010W
Mean Grade: 2.46
Median Grade: 2
Status: beendet



ID: 1109
StudyID: 521
No. Semester: 9
ECTS: 180
Max ECTS in one Semester: 35
Mean ECTS/ Semester: 20
First Semester: 2012W
Mean Grade: 2.58
Median Grade: 3
Status: beendet



ID: 1085
StudyID: 521
No. Semester: 7
ECTS: 186
Max ECTS in one Semester: 38
Mean ECTS/ Semester: 26.57
First Semester: 2011W
Mean Grade: 1.69
Median Grade: 1
Status: beendet

Image 7: All selected paths are shown in a detail view below the scatterplot/heatmap area. Individual courses can be shown by mousing over the elements of the bars.

Use Case

The expert starts at the sort and filter view (Image 1). He filters the table by selecting feasible ranges in the filterbar (Image 2), sorts the table by clicking a column header and then selects all interesting paths that he wants to analyze. Once he has chosen all paths to view he clicks the “analyze selected” button and the exploration page gets generated (takes few seconds).

The default view of the exploration page shows an MDS scatterplot as well as all selected paths as small multiples (Image 3). For more detailed comparison between different paths a non-dimensionality-reduced view is also available - the heatmap view (Image 4).

Paths can be selected in either of the three views (scatterplot, heatmap, small multiples) to be further analyzed (Images 5 & 6). Each selected path gets displayed in detail below the exploration view. By mousing over segments of the different paths courses can be tracked, and seen whether they are similar throughout the paths.

If the experts wants to see different/more courses he can switch back to the filter view and update his selection.

Performance

The performance of the system has greatly improved since the introduction of a server. It is still very resource consuming, since the calculation of the distances between paths uses a lot of computational power. Therefore I had to limit the system to show 100 paths at the most, because the server RAM is too small to handle larger distance matrices.

Feedback from Evaluations

Evaluations introduced a lot of new views/possibilities to the tool. A view that was not included at all in the last iterations was introduced because of feedback, that the system was not scalable to a higher amount of paths - the filter view. By introducing sparklines in a table to show all paths it was possible to individually select paths, and not just randomly sample from all available paths.

Another part was included because of feedback - the heatmap now gets sorted by a reducing the dataset to a 1D with MDS. This tends to show clusters in the heatmap a lot cleaner than alphabetical sorting.

Discussion

Strengths and Weaknesses

The tool is very good at finding clusters in analyzed paths, but it is still not capable to do so for all paths from the faculty. Also the comparison of individual paths is still lacking functionality and will be extended in the future - plans include histograms of course distribution over semesters, course based filtering and difference highlighting.

Lessons learned

I learned a lot about coming up, testing and proving a metric. Even though this is not the main VIS contribution here, it is definitely the thing I learned from that.

The project showed furthermore that the programming capabilities of a single person are very limited, and so is time - I should definitely improve my time management as well as my estimation of effort. (would have tackled a smaller part, but more efficiently)

D3 and JavaScript are very tricky and not always easy to understand - this task definitely improved my skills!

URL

The working tool can be found at: <http://raph.cs.univie.ac.at/vis/pro/>

References

- [1] M.Oppermann, et. al. - 2017 - Bike Sharing Atlas: Visual Analysis of Bike-Sharing Networks
- [2] M.Raji et. al. - 2017 - Visual Progression Analysis of Student Records Data
- [3] E.R.Tufte - 2006 - Beautiful evidence.

Software

Ubuntu 16.04 Server - <https://www.ubuntu.com/download/server>

Node.js - <https://nodejs.org/>

Express - <http://expressjs.com/>

Socket.io - <https://socket.io/>

D3 - <https://d3js.org/>

D3-extended - <https://github.com/wbkd/d3-extended>

Bootstrap - <https://getbootstrap.com/>

jQuery - <https://jquery.com/>

DataTables - <https://datatables.net/>

Bootstrap-slider - <https://github.com/seiyria/bootstrap-slider>