Bringing Official Stanford Course Evaluation Data to Life

Sophia Westwood
Computer Science Department
Stanford University
sophia@cs.stanford.edu

Jonathan Potter
Computer Science Department
Stanford University
jspotter@cs.stanford.edu

ABSTRACT

The data from official Stanford course evaluations has the potential to be the most comprehensive resource available to Stanford students looking to decide what classes to take. Here we present an interactive visualization of course evaluation data optimized for exploration, discovery, comparison, and quick lookup. Our visualization supports three basic actions: search, filter, and explore. Search is implemented by a robust search field which lets users find courses by description or by name, department, and/or professor. Filtration is implemented by a parallel coordinates chart showing a customizable subset of the eighteen ratable course characteristics. Finally, exploration is implemented by a dynamically changing list of courses along with their most basic information and ratings. Our main goals with this design are to prompt users to ask questions of the course ratings and to allow them to draw meaningful conclusions whether they are looking at two courses or two thousand.

INTRODUCTION

Stanford students spend a significant amount of time choosing which courses to take each quarter. They ask questions like:

1. Is this course well-taught?
2. Which of these two courses do people like more?
3. Which professor should I take this course with?

To answer these questions, many students choose to acquire information via word-of-mouth or via online forums such as CourseRank, a repository for course ratings and reviews. Very few, however, choose to utilize the data from Stanford's official course evaluations.

At the end of each quarter, students are asked by the University to evaluate their courses through a standard mechanism. Courses are evaluated on a 1 - 5 scale on eighteen characteristics:

1. Instructor's ability to set and meet clear objectives;
2. Instructor's concern about student learning;
3. Assignment quality;
4. Instructor availability outside of class;
5. Clear explanation of concepts;
6. Emphasis on conceptual understanding;
7. Worthwhile course content;
8. Overall quality of course content;
9. Clarity of evaluation criteria;
10. Fairness of grading;
11. Instructor's ability to inspire and motivate student interest;
12. Overall quality of instructor's teaching;
13. Instructor's knowledge of course material;
14. Logic in topic ordering;
15. Appropriateness of pace at which material is presented;
16. Instructor's ability to prioritize topics;
17. Instructor's ability to connect topics, and;
18. Integration of section/lab into the course.

While these evaluations are not required (students are permitted to decline to evaluate any or all of their courses), the average response rate over the last two years is approximately 85 percent. The University's collection of course evaluation responses respresents the biggest and most comprehensive data set on course quality, with 349,311 evaluations on 16,163 unique course offerings.

Unfortunately, the course evaluation data is also possibly the most under-utilized resource for helping students choose classes. When we initially presented our proposal to visualize this data to our classmates, most were not aware that the data has always been publically available. In addition to being obscure, the current view on the course evaluation data consists of a clunky interface with minimal support for exploration. We will discuss the current view in more detail below.

We set out to create a new visualization for course evaluation data. Our goals for this visualization were:

CS448B, Fall 2012, Stanford University.

1. To create an easy-to-use, fault-tolerant search experience;
2. To support as many different types of comparisons as possible;
3. To create a manageable view of the eighteen different evaluation dimensions, and;
4. To create an explorable view from which students can draw meaningful conclusions, regardless of whether they are looking at two course offerings or two thousand.

RELATED WORK

Stanford's Current Evaluation Data Browser

The figures below are screenshots from Stanford's currently supported system for viewing data from course evaluations.

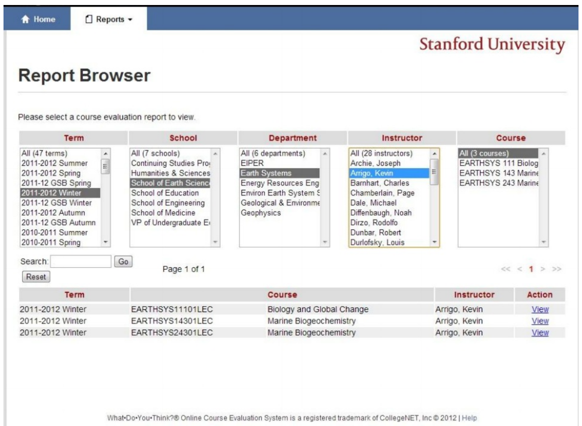

Figure 1. The search page for Stanford's current course evaluation data browser.

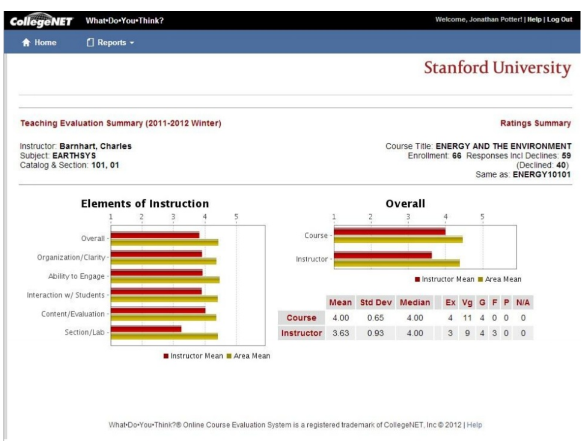

Figure 2. The results page for a single course offering.

These two screenshots show the extent of the existing data exploration interface. It does not offer much in the way of exploration. Each time a user wants to view the data for a course offering, they must either attempt to find the course using the search bar or click through each of the filters (Term, School, Department, Instructor, and Course) in order until they see the desired course appear in the list of results below. They must then navigate to a different page to see any data for that course. Should the user wish to view data for a different course, they must return to the search filter page and start all over again.

We found two main problems with this view. First, it does not allow for exploration or comparison. A user must know precisely which course offering they want to see ahead of time, and they will only see results for this offering. While each datum on the results page is shown side-by-side with a departmental average, there is no way to juxtapose the results of multiple offerings. There is also no way to see any sort of aggregation based on an attribute other than department, nor is there a way to observe trends in course ratings. The site limits users to asking only the question: "What were the average ratings for this particular course offering?" On finding the answer to this question, a user is unlikely to be inspired by the result to ask new questions.

Second, the search experience is not user-friendly. The search field has a high tendency to return either no results or a spate of results, many of which are irrelevant. The clickable filters require a user to recall up to five attributes of a course before they can view its data. Since the values of the search filters are not saved, a user must go through this rigmarole on each search.

CourseRank

When surveying existing work related to displaying opinions on courses, we also examined CourseRank [2], an online forum for rating and reviewing courses. All of CourseRank's information related to course opinions is generated by users on the site itself. Most of its information is qualitative and occurs in the form of student reviews and student questions/responses.

We see two weaknesses to CourseRank's approach. First, the quantitative data is limited. Each course only receives a single rating on a 1 - 5 point scale. Second, the qualitative data, specifically student reviews, is susceptible to response bias. One frequently sees reviews from strongly opinionated people who either loved or hated the course. Rarely does one see an ambivalent review. This makes a big-picture view of the general opinion on the course difficult to obtain.

Existing Research on Course Evaluations

We conducted a brief survey of the existing literature surrounding course evaluations. The first paper we found discusses the effects of course evaluations on course selection. It found that course evaluations do, in fact, have a statistically significant impact on course selection: "The overall effect was significant. Students chose the most highly rated course despite the greater amount of work reported." [3] This point stresses the importance of this domain and of making course evaluation information accessible.

The second paper we found is a literature review that discusses the validity of student course evaluations. It brings up a couple interesting points. First, the primary disadvantage to using online course evaluations instead of paper evaluations is that it can result in a decline in response rate. We did not pay too much mind to this concern, as the average response rate for our data set was 85 percent. Second, as suggested by the first article, “If all else is equal, a student is twice as likely to choose an instructor with ’excellent’ ratings over an instructor with ‘good’ ratings”. Thus, both papers stress the importance of making course evaluation data easily viewable. [1]

METHODS

Data Collection

We were able to procure course evaluation data for the last two years in spreadsheet form from the University registrar. We then converted these into comma-separated value files and imported them into a SQLite database.

We also scraped unit information, GER information, and course descriptions from ExploreCourses. We used ExploreCourses’ XML API to view ExploreCourses search results in an easily parsable manner. We wrote a simple python script that went through our existing course data, found corresponding information on ExploreCourses, and fetched and inserted relevant information.

Search

The first component of our application is the search component. We aimed to create a robust, flexible search experience that returns relevant results. Our search bar has two modes: regular mode and description mode.

In regular mode, the user can search by department, course number, instructor name, or course name. Each token in a query must appropriately match one of these four attributes. Thus, a space in a query is regarded as a logical “and”. In addition, users can add semi-colons to their query to separate multiple sub-queries the results of which they wish to union. A semi-colon is thus regarded as a logical “or”. Logical “ands” are always given tighter binding than logical “ors” in this case.

In description mode, the user can enter text that matches a course if it is contained as a substring in the course’s description.

In both modes, we include a word cloud of possible queries. In regular mode, this word cloud draws from a fabricated list of queries which demonstrate the different types of queries one can execute. In description mode, the word cloud draws from a generated list of the top most frequent words in course descriptions. We performed some hand-pruning on this list to remove words like “prerequisite,” which appear frequently in course descriptions but do not say anything useful about the course.

To make our search experience more fault-tolerant, we added a number of small features, such as basic punctuation stripping, course number separation (cs106a becomes cs 106a), and substring matching on course name and teacher name. We also implemented a very basic token detection mechanism that checks if a token is a department name. If it is, the token is only used in exact matching against department names. This is so that, for example, CSRE courses do not show up in search results for a query containing “CS”.

Filter

The second component of our application is the filter component. We wanted this component to be the crux of our visualization that would allow users to see an overarching picture of their search results and to narrow the results based on what they see. For this component, we used a parallel coordinates chart with axes corresponding to the ratable characteristics of a course.

When a search result is returned, each course offering in the result is added to the parallel coordinates chart as a line connecting that offering’s values for each of the axes displayed. If a user’s search result contains only a handful of offerings, the user can easily see the exact paths these offerings take from one rating to the next. If, on the other hand, a user’s search result contains course offerings for an entire department, the user can roughly see the distribution of ratings over that department for each rating based on the density of lines around each axis.

A user can then use the parallel coordinates chart to filter their search results based on one or more characteristics via brushing and linking. By dragging a rectangular range over an axis, the user narrows their search results to only those offerings with values of that particular characteristic falling within the range of the rectangle.

A user can add axes to, subtract axes from, or rearrange axes on the parallel coordinates chart. This way, a user will be neither overwhelmed by a visualization that attempts to convey all eighteen characteristics nor limited to a view which only conveys a single characteristic.

The parallel coordinates chart makes use of four visual encoding variables, which we will discuss in decreasing order of importance. First, it uses position. The position of a line’s intercept with an axis indicates the corresponding course’s rating for that attribute. Since almost all ratings occur above the 3.0 mark, we decided to combine all lower ratings under a tick marked “lower”. While this removes the reference point of a real zero and reduces our rating variables from quantitative to ordinal, we feel that this gives users a more accurate look at rating distributions.

Second, the chart uses opacity. The lines on the chart are translucent so that rating distributions (as represented by line density) can be visually approximated by opacity. This visual encoding is designed to support more high-level analysis, such as comparing departments.

Third, the chart uses slope. The slope of a line as it moves from one axis to another roughly shows the change in value across a pair of adjacent ratings. Since the axes are moveable, slope comparisons can be made between any pair of attributes. Since near-zero slopes are the most common slopes in our chart, steep positive or negative slopes stand out, particularly in less dense regions.

Finally, the chart uses area. A user can judge the quality of a large collection of course offerings (a department, for instance) in part by the amount of white area that appears at the bottom of the chart. More white area means fewer low ratings.

Explore

The third component of our application is the actual list of search results. This list exists in its own scrollable container, allowing each course offering to be seen against the parallel coordinates chart above it. Each row consists of five columns: course name/number, course rating, teacher name, teacher rating, and term. A user can sort by each of these columns in either order. Sorting by course name/number can be useful for seeing trends as courses become more advanced. Sorting by term can be useful for seeing trends of an individual course over time.

Both the course name and the instructor name are linked and can be clicked to view the search results of that particular course or instructor.

Share

We made each search result shareable by URL by including the search query as the fragment. Our review of existing literature indicates that data is most engaging when it is brought into discussion, and making visualization results shareable is one way to achieve this. In its paper, CourseRank discusses the importance of the community aspects of its site. [2] In addition, Wattenberg et al. argue that "an information visualization tool may be fruitfully viewed not as a tool but as part of an online social environment." They discuss how their application, NameVoyager, became "a stimulus to conversation an repartee." [4] We therefore made it a priority to support data sharing in our application.

RESULTS

Figure 3 is a screenshot of a simple search query along with the resulting response and visualization.

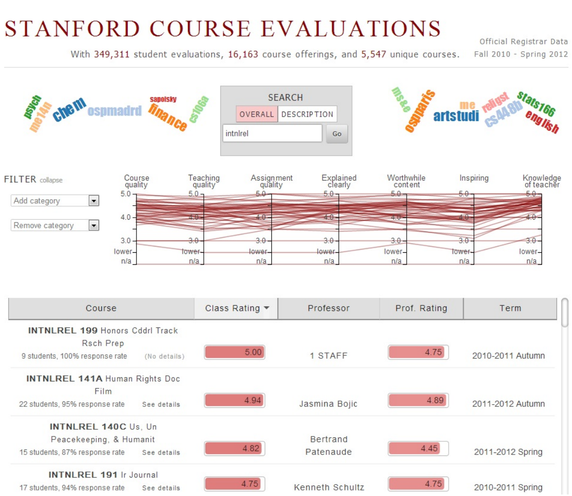

Figure 3. A search for "intnlrel".

Figure 4 shows the same result when filtered by clear explanation and the instructor's ability to inspire.

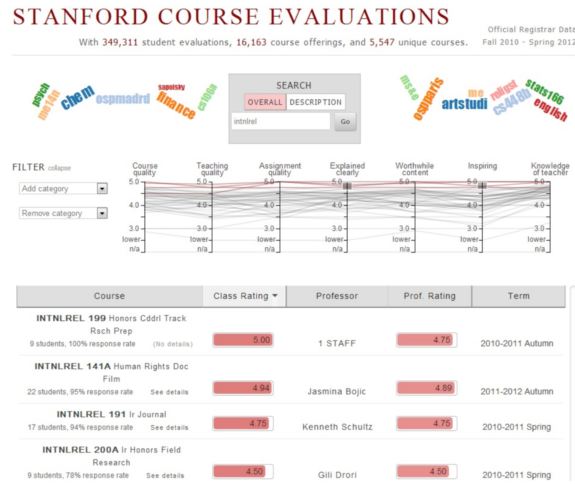

Figure 4. Simple filtering using brushing and linking.

One can highlight a course's line on the parallel coordinates chart by hovering over the course row in the result table, as shown in Figure 5.

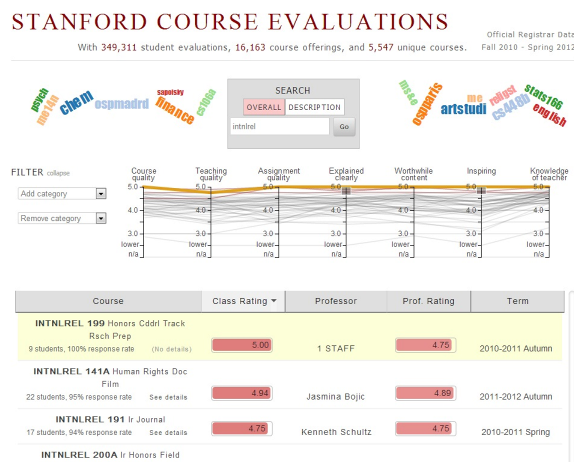

Figure 5. Course highlighting.

In Figures 6 and 7 we see the addition of an axis for logical content ordering.

The result table can be sorted by any of the columns. In Figure 8 we see an inverse re-sorting by term.

Figure 9 shows a more complicated query involving semi-colon separation.

Figure 10 shows a search in description mode.

The site generally requires about one second to retrieve the data and between one and two seconds to construct the visualization. Adding and removing axes is instantaneous. Filtering via brushing and linking occurs generally within

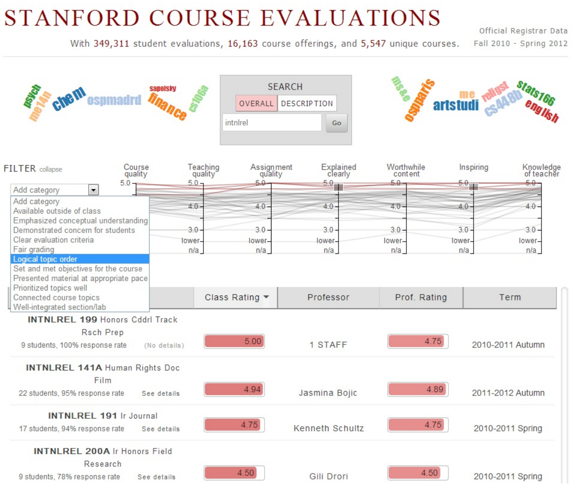

Figure 6. Selecting an additional axis from the drop menu.

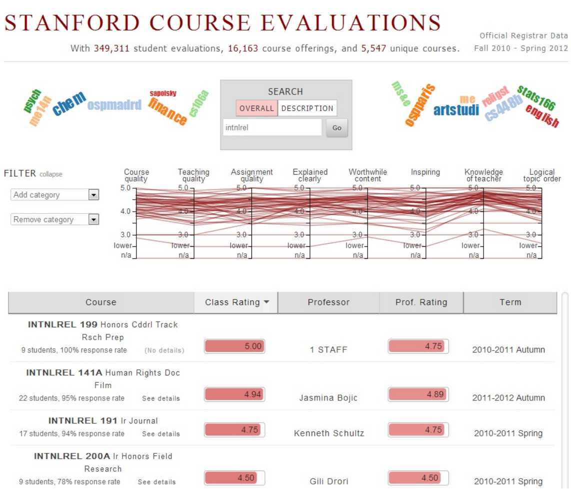

Figure 7. Axes are added to the rightmost portion of the chart.

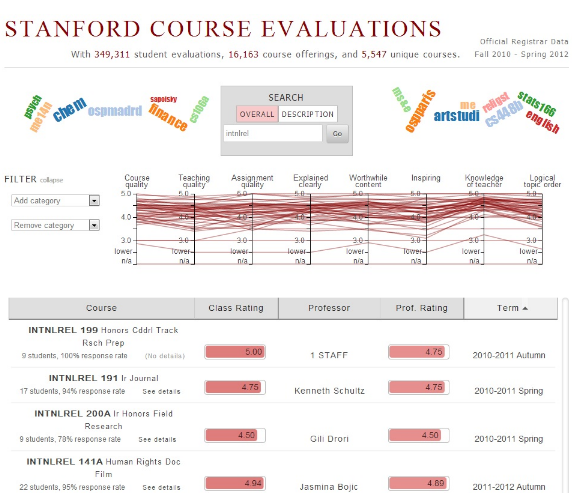

Figure 8. Results are re-sorted inversely by term.

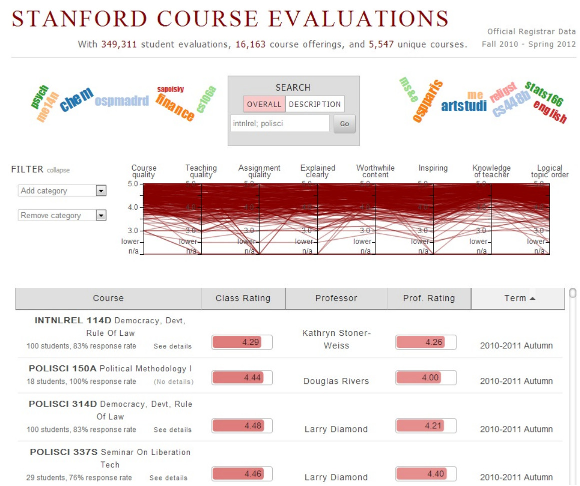

Figure 9. Displays results for both "intnlrel" and "polisci".

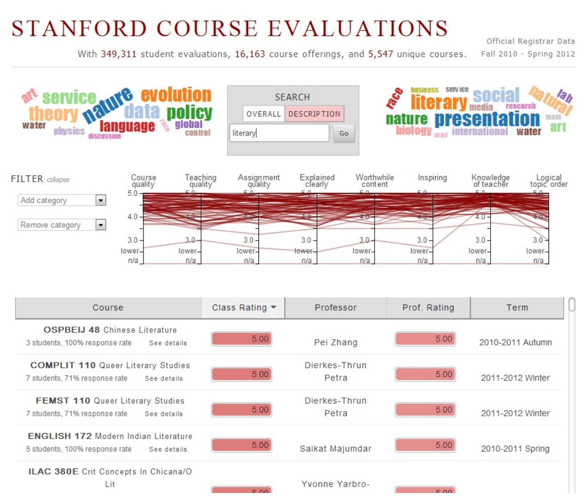

Figure 10. Search for "literary" in course descriptions.

a quarter of a second, although occasionally a user experiences lag when adding filters on large data sets.

DISCUSSION

With the help of our users, we made a number of interesting observations about course evaluation data using our application. We will enumerate them here.

Departmental Comparison: English vs. Math

Figures 11 and 12 show the results returned when the search query is "english" and "math" respectively. This is a prime example of the sort of high-level comparison that can be made using our visualization. Looking at the parallel coordinates chart for the English department, one can see that the vast majority of the line mass occurs above the 4.0 mark. This suggests that English generally receives fairly high course evaluation ratings. Looking at the chart for the Math department, one sees a shift in mass towards the area of the chart between the 3.0 mark and the 4.0 mark. This suggests that while most Math courses still receive fairly high ratings, a significant portion of them receive lower ratings.

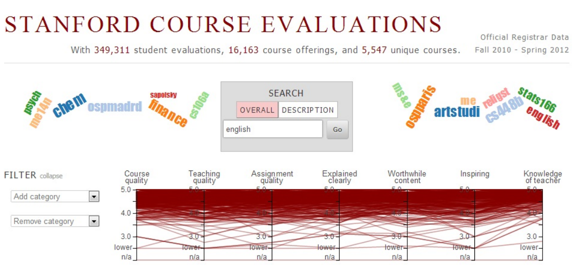

Figure 11. Visualization of evaluation data for the English department.

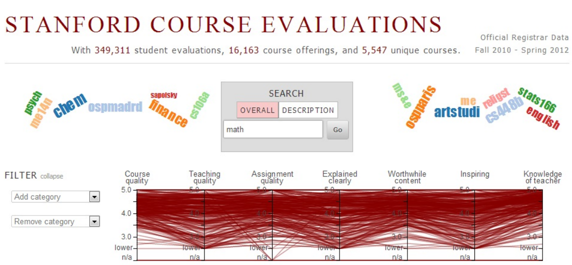

Figure 12. Visualization of evaluation data for the Math department.

Professor Ratings for Upper vs. Lower Division Courses

Figure 13 shows a search for the Biology core (Biology 41, 42, and 43) with an inverse sort on professor rating. A user might conclude that they should avoid these professors altogether. However, conducting a subsequent search on one of their names reveals the results shown in Figure 14. We see that this professor actually receives high ratings when they teach upper-division Biology courses. While one cannot ascertain why this might be based on the data, one could speculate that it has to do with the professor's particular interests, or with the fact that students tend to take upper-division courses out of interest rather than requirement.

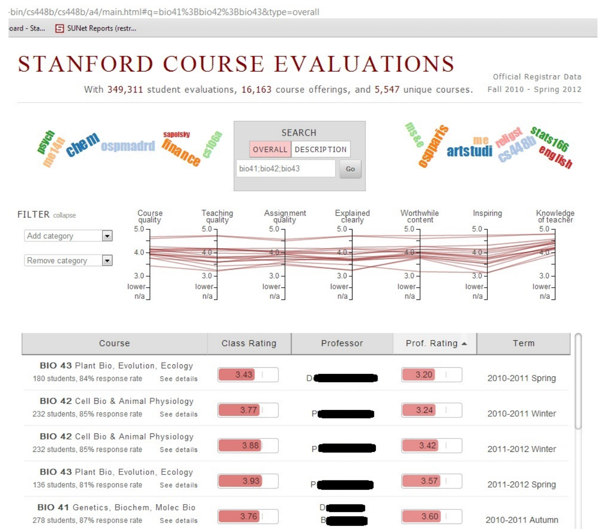

Figure 13. Result of sorting Biology core offerings inversely by professor rating.

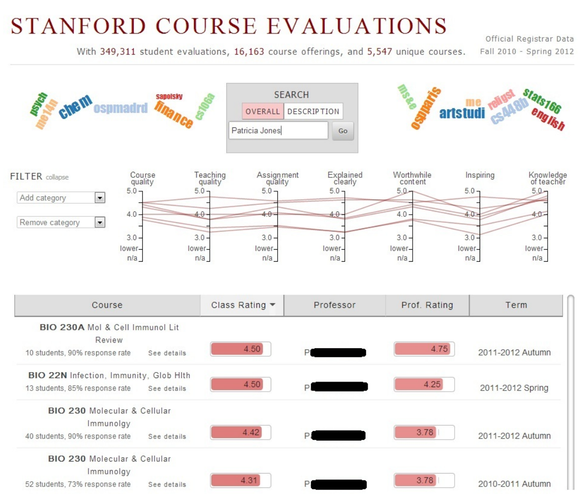

Figure 14. Result of searching for one of the professors from the result set in the last figure.

Impact of Public Offering on Evaluation Results

Finally, Figures 15, 16, 17, and 18 show interesting trends around the recent deployment of public offerings of certain courses. The figures show evaluation data for two such courses: CS 145 and CS 221. Looking at the change in ratings for each course from the 2010-2011 school year to the 2011-2012 school year, one sees a rise in ratings for CS 145. One also sees a drop in ratings for CS 221, particularly around assignment quality. Again, answers to questions of "why" given the data can only be speculative. However, perhaps this occurred because the CS 145 material is better suited to automated online exercises than the CS 221 material.

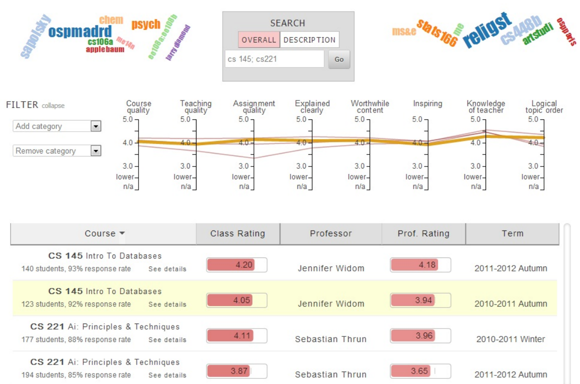

Figure 15. Data for CS 145 and CS 221 with the 2010-2011 offering of CS 145 highlighted.

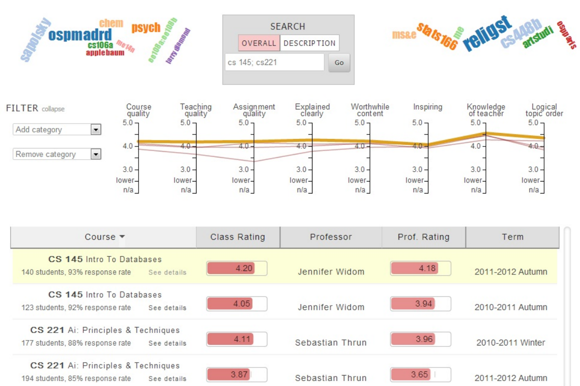

Figure 16. 2011-2012 offering of CS 145 highlighted.

Informal User Observation

We conducted an informal user study. We noticed the following common threads between the interactions each of our users had with our application:

1. Users generally began with questions about individual courses. Some wanted to look up courses that they planned on taking next quarter. Others wanted to look up courses they had already taken. Those who fell into the second category seemed eager to confirm their own experiences and opinions.

2. There were several moments where students saw search

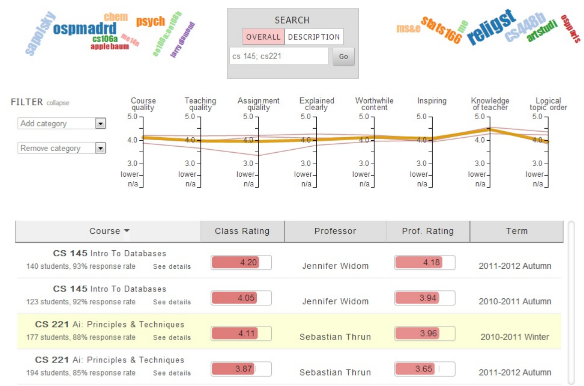

Figure 17. 2010-2011 offering of CS 221 highlighted.

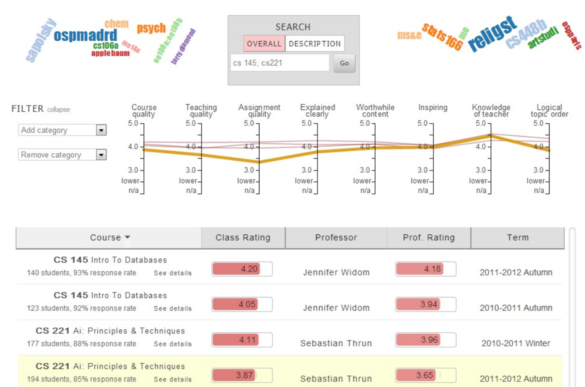

Figure 18. 2011-2012 offering of CS 221 highlighted.

results and reacted in a way that indicated that their expectations were borne out.

3. Low-level searches frequently prompted higher-level questions. After looking up a particular course, users often followed up with questions about the department as a whole.

4. Users who viewed the application in groups often encouraged each other to make queries. This drives home our notion that data applications are more engaging when they support social exploration.

We see the trend of moving from low-level analysis to high-level analysis as a good one. This allows users to see what an individual datum looks like before shifting to a view that shows data for a massive number of course offerings.

FUTURE WORK

Features we would like to add to our application include:

1. Adding tutorial features on how to use our parallel coordinates chart. We found that students had trouble overcoming the initial difficulty of working with such an unfamiliar interactive data visualization, although they had little trouble interacting with it once this initial difficulty was surmounted.

2. Parsing out prerequisite information from ExploreCourses and creating a directed graph visualization of courses and their prerequisites.

3. Migrating from SQLite to MySQL for faster lookup times and better performance under heavy load.

4. Integrating more aggregate information (department averages, professor averages, and course averages) into our visualization.

Soon, we would like to obtain official university support for our application and deploy it in a location accessible to all students.

REFERENCES

1. H.M. Anderson, J. Cain, and E. Bird. Online student course evaluations: Review of literature and a pilot study. *American Journal of Pharmaceutical Education*, 69(1):34–43, 2005.

2. Benjamin Bercovitz, Filip Kaliszan, Georgia Koutrika, Henry Liou, Zahra Mohammadi Zadeh, and Hector Garcia-Molina. Courserank: a social system for course planning. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, SIGMOD '09, pages 1107–1110, New York, NY, USA, 2009. ACM.

3. J. Coleman and WJ McKeachie. Effects of instructor/course evaluations on student course selection. *Journal of Educational Psychology*, 73(2):224, 1981.

4. M. Wattenberg and J. Kriss. Designing for social data analysis. *Visualization and Computer Graphics, IEEE Transactions on*, 12(4):549–557, 2006.