

# Yale Daily News

---

Published: Monday, January 12, 2009

## Kosslyn: Elegance, not tech support

By Justin Kosslyn  
The World Beautiful

Computer science is tragically misunderstood. Popular opinion notwithstanding, my major is not preprofessional.

It does not teach how to fix a printer — though I sometimes wish it did — nor does it explain why my laptop keeps making disquieting noises. Despite three and a half years of classes, I have yet to learn how to create a slick Web site or upgrade a hard drive, hack into servers or configure a firewall. It's not my fault — none of those topics are in the computer science curriculum.

What, then, you may wonder, is computer science? It is the study of the structures and the construction of elegant systems.

Let's flesh this out. Take Google's search algorithm as an example. (Full disclosure: I interned at Google last summer, am currently the campus representative, and will be returning there next year.) Why did Google beat out AltaVista and all the other early search engines? Simple: by examining the structure of the World Wide Web.

There is no magic or mystery to the Web's basic structure, which consists of pages connected with hyperlinks. If you are reading this online, for example, on the top left of the page is a large hyperlink to the YDN's home page. Google's innovation was to realize that no page exists in a vacuum, and that by paying attention to the structure of the hyperlinks — to what was linking where — searches would yield more promising results.

The key to better results was attention to a Web site's salience — its prominence online — which is represented as a number from one to 10, known as PageRank. For example, as of the time of writing, huffingtonpost.com has a PageRank of eight (high salience), yaledailynews.com has a PageRank of seven, www.yale.edu/rumpus has a PageRank of five, and thecrimson.com has a PageRank of four (as if we needed Google to tell us that the YDN is far superior to the Crimson).

Sometimes hyperlinks change. For example, last spring huffingtonpost.com linked to a number of YDN articles about Aliza Shvarts '08. Huffingtonpost.com was thereby effectively endorsing yaledailynews.com. Hundreds of other high-profile external sites have also linked to the YDN, ranging from Wikipedia to Sports Illustrated. Each of these mini-endorsements boosts YDN's PageRank, resulting in the high ranking. And when the YDN links to another site, some of the YDN's high PageRank carries over.

You may be surprised by how straightforward this all is. This was a patent-worthy (number 6,285,999) computer science breakthrough with large practical implications, but the core of it is exceedingly simple: When important sites link to a new site, the new site is probably important too. Thus, the quality of Google's search results arises from the computer science perspective — understand structure, build elegant systems — applied to the task of Web searches.

But it is shopping period, and you have a more pressing question at hand: What courses should you shop? Computer science also has something to say about this kind of dilemma.

If you really wanted the optimal course list, you might proceed as follows. First, enumerate aspects of each course: class size, ratings from past years, classroom activities (discussions, movie screenings, lectures, visiting experts etc.), and so on. Then list out every course you have taken so far, recording every aspect of each course as well as whether you enjoyed it.

You — or a computer — are now in a position to build a model of your likes and dislikes, as well as to understand which dimensions matter more. You may learn, for example, that having movie screenings significantly boosts your enjoyment of a course while visiting experts make little difference.

Finally, you can use this understanding to make predictions for the future. By comparing the predictions against reality, the model can be further refined. In this way you are using the computer science approach: seeking to understand structure and build an elegant system to help solve an immediate problem.

This approach can be applied to a wide range of classification problems. For my senior project, for example, I used a large government database to examine the characteristics of people at risk of alcohol abuse.

Even if you have no interest in majoring in computer science, a bit of that perspective is valuable. History, political science, the harder sciences and even literature are replete with implicit structures that computer science can help frame.

So when you consider the structure of your curriculum this semester, consider computer science. If you already have a little programming experience — such as programming your calculator in high school and running analyses in MATLAB or Stata — your course is Introduction to Computer Science. If you are considering putting a toe in the water for the very first time, try Introduction to Programming.

And if you need to fix your printer, try calling tech support.

Justin Kosslyn is a senior in Ezra Stiles College.