27-Jun-2011
kasper.peeters@durham.ac.uk

CM3 - The 150 Billion Dollar Eigenvector - 2011/12

Outline

The problem that internet search engines like Google try to solve is in principle easy to formulate: given a set of keywords, return those web pages which contain these keywords, and sort them by relevance. The key question is of course "How do you figure out the relevance of a page?". This is where some interesting mathematics comes in. (There is a whole host of other interesting problems related to search engines, such as how you handle a database of billions of pages, how you manage to keep it up to date, or how you deal with hundreds of thousands of search requests per second. Many of these are, however, of a more technical nature).

At the core of the solution to this problem you will often find graph theory and linear algebra ideas: the web is represented as graph of pages, and the key ingredient is a "transfer matrix" which estimates how likely it is that a user will navigate from one page to another one. This leads to concepts such as Markov Chains, sparse matrices, eigenvector computation, and information theory. Google's PageRank algorithm is just one of many different mathematical ways to construct a measure of relevance; there are various others, such as HITS, SALSA and BM25F, in use by other search engines. Comparing them is one good way to make your project more interesting.

Structure, supervision & meetings

At the beginning of term we will have regular meetings once every two weeks with all four students allocated to the project. During this period, we will go over some of the basic material and papers, to get you started. As time progresses, each of you will most likely pick a somewhat different direction, and it will make more sense to meet on an individual basis when actual questions arise. It is always useful (both for you and for me) if you can first try to ask a question over email, even if we end up discussing it on the blackboard later: it forces you to think carefully about what puzzles you, and you will have a written record of my answer that you can read again later.

If you want to prepare yourself over summer, it is useful to browse through some of the suggested literature, perhaps search the net for some of the other approaches and follow citation links in Google Scholar (see below). Try to figure out which aspects of the search engine problem you find most interesting (even if you have only a rough idea, that will save a lot of time later).

Literature pointers and summer reading/viewing

- The basic Google paper is

  > Sergey Brin and Lawrence Page,
  > "The anatomy of a large-scale hypertextual Web search engine",
  > Computer Networks and ISDN Systems **33** (1998) 107-117.

  Available online in PDF form; use Google Scholar. Nice to read, though there is better material out there for the maths.

- There is a lot of basic material in the book

  > Amy N. Langville & Carl D. Meyer,
  > "Google's PageRank and Beyond:
  > The Science of Search Engine Rankings"
  > Princeton University Press 2006.

  Ask me if you cannot locate a copy. This book discusses various other, non-Google ideas as well, and also contains a long list of original and related papers that can get you further into the topics. Highly recommended.

- You can find a lot of material using Google Scholar, the academic part of Google that searches through all journal papers (and very often gives you a link to a freely available PDF file so you do not have to go to the library). It also allows you to search for papers that cite a given paper. Try this!

Use of computers

Even though this is not a computer project, and you are not required to do any programming, you can potentially benefit tremendously from trying to implement (some of) the mathematical ideas in a concrete program, so you can 'play with them'. Experience with related projects in the previous years has shown that the topic becomes much more 'alive' if you are able to see things happen in front of your own eyes. Moreover, the mark for a CM3 project usually increases significantly if you show that you understand how the maths works in practise at the level of your own examples. Many ideas can be explored easily with general purpose programs such as Mathematica, Maple or Matlab, and you really do not need to be an expert of any kind to use these systems. Feel free to ask me for help to get started.