Applications of Matrix Computations to Search Engines

With no doubt, Google is currently the most widely used search engine on the Web. Behind its success, there is a very effective, ingenious and at the same time simple algorithm that arranges Web Pages in order of certain notion of importance. The construction of such algorithm, known as PageRank, is an elegant application of a number of results in linear algebra and matrix computations.

The central idea is to compute a dominant left eigenvector (called the PageRank vector) of a huge stochastic irreducible matrix (the number of columns is in the order of billions). For practical and theoretical reasons the original matrix is modified or perturbed by introducing a so-called personalization vector, which allows to manipulate not only the ranks of particular WebPages, but also the speed of convergence of the algorithm [2, 6].

Approaches other than the power method are being investigated, including adaptive techniques [4], extrapolation methods [5], and partitions of the matrix according to groups of pages with no outlinks [7], all of them with the purpose of speeding up convergence by a certain factor. These and other new methods do not compete with one another and hence there is the possibility of combining two or more algorithms to obtain greater speeds of convergence.

A second and related problem deals with updating the PageRank vector. The computation of such eigenvector is very expensive, and the PageRank vector from a prior period is nearly useless for initializing the power method for the next period, so that new computations are restarted almost from scratch. There is currently active research on how to efficiently use previous computations to obtain a new PageRank vector. Worth mentioning are iterative aggregation (IAD) techniques [3, 8], Monte Carlo methods and asymptotic analysis [1]. Although these approaches yield some faster convergence, there is plenty of room for research in this direction. In particular, in [8] it is suggested that extrapolation techniques from [5] could possibly be combined with aggregation techniques to further accelerate the updating process.

Recently ([9]) the original power method has been combined with some matrix reorderings (with the goal of finding permutations of the link matrix that reduce the magnitude of its subdominant eigenvalue), giving faster rates of convergence, and the linear systems approach has been combined with iterative aggregation/disaggregation techniques to obtain more efficient algorithms. Among other things, it remains to obtain a sound theoretical proof on how matrix permutations can affect the magnitude of the subdominant eigenvalue of the link matrix, which partitions are optimal for IAD, and to experiment with several numerical algorithms for sparse linear systems. At the same time, extrapolation and asymptotic analysis may be combined with other techniques to achieve more efficient algorithms.

Prerequisites: A basic background on linear algebra, and ideally some programming experience.
References


