Shifted power-GMRES method accelerated by extrapolation for solving PageRank with multiple damping factors

Luca Lombardo

Abstract

In the years following its publication in 1998, the PageRank model has been studied deeply to be extended in fields such as chemistry, biology and social network analysis. The aim of this project is the implementation of a modified version of the Power method to solve the PageRank problem with multiple damping factors. The proposed method is based on the combination of the Power method with the shifted `GMRES` method.

Contents

1 Introduction **2**
1.1 Overview of the classical PageRank problem 3

2 The shifted power method for PageRank computations **4**
2.1 The implementation of the shifted power method 4

1 Introduction

The PageRank model was proposed by Google in a series of papers to evaluate accurately the most important web-pages from the World Wide Web matching a set of keywords entered by a user. For search engine rankings, the importance of web-pages is computed from the stationary probability vector of the random process of a web surfer who keeps visiting a large set of web-pages connected by hyperlinks. The link structure of the World Wide Web is represented by a directed graph, the so-called web link graph, and its corresponding adjacency matrix $G \in \mathbb{N}^{n\times n}$ where n denotes the number of pages and G_{ij} is nonzero (being 1) only if the *jth* page has a hyperlink pointing to the *ith* page. The transition probability matrix $P \in \mathbb{R}^{n\times n}$ of the random process has entries as described in 1.

$$P(i,j) = \begin{cases} \dfrac{1}{\sum_{k=1}^{n} G_{kj}} & \text{if } G_{i,j} = 0 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

The entire random process needs a unique stationary distribution. To ensure this propriety is satisfied , the transition matrix P is usually modified to be an irreducible stochastic matrix A (called the Google matrix) as follows:

$$A = \alpha\tilde{P} + (1-\alpha)\,ve^T \tag{2}$$

In 2 we have defines a new matrix called $\tilde{P} = P + vd^T$ where $d \in N^{n\times 1}$ is a binary vector tracing the indices of the damping web pages with no hyperlinks, i.e., $d(i) = 1$ if the *i-th* page ha no hyperlink, $v \in \mathbb{R}^{n\times n}$ is a probability vector, $e = [1,1,...,1]^T$ and $0 < \alpha < 1$, the so-called damping factor that represents the probability in the model that the surfer transfer by clicking a hyperlink rather than other ways. Mathematically, the PageRank model can be formulated as the problem of finding the positive unit eigenvector x (the so-called PageRank vector) such that

$$Ax = x, \quad \|x\| = 1, \quad x > 0 \tag{3}$$

or, equivalently, as the solution of the linear system

$$(I - \alpha\tilde{P})x = (1-\alpha)\,v \tag{4}$$

The authors of the paper [1] emphasize how in the in the past decade or so, considerable research attention has been devoted to the efficient solution of problems 3 4, especially when n is very large. For moderate values of the damping factor, e.g. for $\alpha = 0.85$ as initially suggested by Google for search engine rankings, solution strategies based on the simple Power method have proved to be very effective. However, when α approaches 1, as is required in some applications, the convergence rates of classical stationary iterative methods including the Power method tend to deteriorate sharply, and more robust algorithms need to be used.

In the reference paper that we are using for this project, the authors focus their attention in the area of PageRank computations with the same network structure but multiple damping factors. For example, in the Random Alpha PageRank model used in the design of anti-spam mechanism [2], the rankings corresponding to many different damping factors close to 1 need to be

computed simultaneously. They explain that the problem can be expressed mathematically as solving a sequence of linear systems

$$(I - \alpha_i \tilde{P}) x_i = (1 - \alpha_i) v \quad \alpha_i \in (0,1) \quad \forall i \in \{1,2,...,s\} \tag{5}$$

As we know, standard PageRank algorithms applied to 5 would solve the s linear systems independently. Although these solutions can be performed in parallel, the process would still demand large computational resources for high dimension problems. This consideration motived the authors to search novel methods with reduced algorithmic and memory complexity, to afford the solution of larger problems on moderate computing resources. They suggest to write the PageRank problem with multiple damping factors given at once 5 as a sequence of shifted linear systems of the form:

$$(\frac{1}{\alpha_i} I - \tilde{P}) x^{(i)} = \frac{1 - \alpha_i}{\alpha_i} v \quad \forall i \in \{1,2,...,s\} \quad 0 < \alpha_i < 1 \tag{6}$$

We know from literature that the Shifted Krylov methods may still suffer from slow convergence when the damping factor approaches 1, requiring larger search spaces to converge with satisfactory speed. In [1] is suggest that, to overcome this problem, we can combine stationary iterative methods and shifted Krylov subspace methods. They derive an implementation of the Power method that solves the PageRank problem with multiple dumpling factors at almost the same computational time of the standard Power method for solving one single system. They also demonstrate that this shifted Power method generates collinear residual vectors. Based on this result, they use the shifted Power iterations to provide smooth initial solutions for running shifted Krylov subspace methods such as `GMRES`. Besides, they discuss how to apply seed system choosing strategy and extrapolation techniques to further speed up the iterative process.

1.1 Overview of the classical PageRank problem

The Power method is considered one of the algorithms of choice for solving either the eigenvalue 3 or the linear system 4 formulation of the PageRank problem, as it was originally used by Google. Power iterations write as

$$x_{(k+1)} = A x_k = \alpha \tilde{P} x_{(k)} + (1 - \alpha) v \tag{7}$$

The convergence behavior is determined mainly by the ratio between the two largest eigenvalues of A. When α gets closer to 1, though, the convergence can slow down significantly.

As stated in [1] The number of iterations required to reduce the initial residual down to a tolerance τ, measured as $\tau = \|Ax_k - x_k\| = \|x_{k+1} - x_k\|$ can be estimated as $\frac{\log_{10} \tau}{\log_{10} \alpha}$. The authors provide an example: when $\tau = 10^{-8}$ the Power method requires about 175 steps to converge for $\alpha = 0.9$ but the iteration count rapidly grows to 1833 for $\alpha = 0.99$. Therefore, for values of the damping parameter very close to 1 more robust alternatives to the simple Power algorithm should be used.

2 The shifted power method for PageRank computations

In this section presents the extensions of stationary iterative methods for the solution of PageRank problems with multiple damping factors, as presented in [1]. We are interested in knowing if, for each method, there exists an implementation such that the computational cost of solving the PageRank problem with multiple damping factor is comparable to that of solving the ordinary PageRank problem with single damping factor.

2.1 The implementation of the shifted power method

Inspired by the reason why shifted Krylov subspaces can save computational cost, the authors of [1] investigate whether there are duplications in the calculations of multiple linear systems in this problem class by the stationary iterative methods, so that the duplications in the computation can be deleted and used for all systems. It's some sort of dynamic programming approach. Firstly, they analyze the Power method applied to the sequence of linear systems in 4. It computes at the *i-th* iteration approximate solutions $x_k(i)(1 \le i \le s)$ of the form

$$\alpha_i^k \tilde{P}^k x_k^{(i)} + (1-\alpha_i^k) \sum_{j=0}^{k-1} \alpha_i^j \tilde{P}^j v \tag{8}$$

If the s systems in 4 are solved synchronously, that is all $x_k^{(i)}$ are computed only after all previous approximations $x_{k-1}^{(j)}$ are available, then the computation can be rearranged efficiently as follows:

- at the first iterations
  - compute and store $\mu_1 = \tilde{P}x_0$ and $\mu_2 = v$;
  - compute and store $x_1^{(i)} = \alpha_i \mu_1 + (1-\alpha_i)\mu_2$;
- at any other subsequent iteration $k > 1$
  - compute and store $x_k^(i) := (1-\alpha_i)\sum_{j=0}^{k-2} \alpha_i^j \tilde{P}^j v = x_{k-1}^{(i)} - \alpha_i^{k-1}\mu_1$;
  - compute and store $\mu_1 = \tilde{P}\mu_1$ and $\mu_2 = \tilde{P}\mu_2$;
  - compute and store $x_k^{(i)} = \alpha_i \mu_1 + x_k^{(i)} + (1-\alpha_i)\alpha_i^{k-1}\mu_2$.

This implementation requires at most 2 matrix-vector products at each step, which is a significant gain compared to the s matrix-vector products required by the standard Power method to compute $x_{k+1}^{(i)}$, especially when $s \gg 2$.

An efficient implementation can compute and store $\mu = \tilde{P}v - v$ at the first iteration and store

$$\mu = \tilde{P}^{k-1}(\tilde{P}v - v) = \tilde{P} \cdot (\tilde{P}^{k-2}(\tilde{P}v - v))$$

At each *k-th* iteration ($k > 1$), and finally from each approximate solution as $x_k^{(i)} = \alpha_i^k \mu + x_{k-1}^{(i)}$. The residual vector $r_k^{(i)}$ associated with the approximate solution $x_k^{(i)}$ has the following expression

$$r_k^{(i)} = Ax_k^{(i)} - x_k^{(i)} = x_{k+1}^{(i)} - x_k^{(i)} = \alpha_i^{k+1}\tilde{P}^k(\tilde{P}v - v) \tag{9}$$

Since in general each of the s linear systems may require a different number of Power iterations to converge, the s residual norms have to be monitored separately to test the convergence.

Now we can summarize the efficient implementation of the Power method presented in this section for solving problem 4 in Algorithm 1, and we call it the shifted Power method hereafter.

Algorithm 1 Shifted-Power method for PageRank with multiple damping factors

Require: $\tilde{P}$, v, τ, $\max_{mv}$, α_i $(1 \le i \le s)$
Ensure: mv, $x^{(i)}$, $r^{(i)}$ $(1 \le i \le s)$
  Compute $\mu = \tilde{P}v - v$
  Set $mv = 1$
  for $i = 1 : s$ **do**
    Compute $r^{(i)} = \alpha_i \mu$
    Compute $Res(i) = \|r^{(i)}\|$
    if $Res(i) \ge \tau$ **then**
      Compute $x^{(i)} = r^{(i)} + v$
    end if
  end for
  while $\max(Res \ge \tau)$ and $mv \le \max_{mv}$ **do**
    compute $\mu = \tilde{P}\mu$
    $mv = mv + 1$
    for $i = 1 : s$ **do**
      if $Res(i) \ge \tau$ **then**
        Compute $r^{(i)} = \alpha_i^{k+1} \mu$
        Compute $Res(i) = \|r^{(i)}\|$
        if $Res(i) \ge \tau$ **then**
          Compute $x^{(i)} = r^{(i)} + x^{(i)}$
        end if
      end if
    end for
  end while

Where mv is an integer that counts the number of matrix-vector products performed by the algorithm. The algorithm stops when either all the residual norms are smaller than the tolerance τ or the maximum number of matrix-vector products is reached. An implementation of this algorithm written in Python is available in the github repository [3] of this project.

References

[1] Zhao-Li Shen, Meng Su, Bruno Carpentieri, and Chun Wen. Shifted power-gmres method accelerated by extrapolation for solving pagerank with multiple damping factors. *Applied Mathematics and Computation*, 420:126799, 2022.

[2] Paul G. Constantine and David F. Gleich. Random alpha pagerank. *Internet Mathematics*, 6(2), 1 2009.

[3] Luca Lombardo. Shifted power method for solving pagerank with multiple damping factors. `https://github.com/lukefleed/ShfitedPowGMRES`, 2022.