PageRank

Mikhail Goncharov

Optimization Class Project. MIPT

Introduction

At the end of the last century two Ph.D. candidates of Stanford University Sergey Brin and Larry Page proposed a breakthrough algorithm which ranked web-pages according to their importance. The idea of the algorithm was that the importance of each page should depend (linearly) on the importances of the pages which refer to it. The problem of finding such importances was named PageRank and it still attracts both academic community and industry. In this project we consider three radically different approaches to solving PageRank problem: first is so called power method, which was mentioned in the original Brin's and Page's paper, second is Monte Carlo method and last exploits optimizational technique.

Problem

Consider a weighted, directed graph with n nodes. The nodes correspond to the web-pages and the weight w_{ij} of edge from node i to node j is equal to number of links from page i to page j divided by number of all links from page i. Denote transposed graph's adjacency matrix as P. Then, the vector x^* of the importances should satisfy

$$Px^* = x^*, \quad x^* \in \mathbb{R}^n,\ x^* > 0,\ \|x^*\|_1 = 1, \tag{1}$$

One can see that x^* is eigenvector of P corresponding to an eigenvalue equal to 1. Such eigenvalue always exists for stohastic matrices (matrices with non-negative elements with sum of elements in each column equal to 1). Moreover, corresponding eigenvector with real non-negative components also exists (it can be proved in terms of the Brauer theorem), but it can be non-unique. In fact, when we start working with real graphs we face two problems. The first is that P could be non-stohastic! Indeed, there can be null columns, which correspond to nodes without outgoing edges, and in this case the solution of (1) doesn't exist. The second problem is nonuniqueness of solution. To deal with these problems we will just change a model:) But before it, we need to briefly introduce one result from Markov chains theory.

Marcov chain is called ergodic if it has limit distribution $x^{lim} \overset{def}{=} \lim P^k x_0$ with $k \to \infty$ for arbitary intitial distribution x_0. Ergodic theorem claims that ergodicity is equivalent to strong connection and non-periodicity, moreover, for ergodic chains x^{lim} coinsides with the unique stationary distribution x^* (s.t. $x^* = Px^*$) and also, if we consider random surfing on the graph, then

$$f_j \overset{a.s.}{\to} x_j^*,\ \forall j \tag{2}$$

where $f_j = \frac{1}{n}\sum_{i=1}^n I(\xi_i = j)$ is frequency of getting in j-th node, ξ_i is random variable, which is equal to index of node on i-th step.

Now one can see that ergodicity is very usefull property, so let's make our graph ergodic. At first, consider that if there is no link from current page we randomly go next to the arbitary one. Besides, on each step with some probability $1-\alpha$ we also do the same. It is easy to see that new matrix, corresponding to the new stohastic graph (which now is ergodic!) reads as follows

$$P' = \alpha(P + \frac{1}{n}ed^T) + (1-\alpha)\frac{1}{n}ee^T,$$

where e is a column of ones and d is column-indicator of null columns in P.

Power method

Since graph is ergodic, as was mentioned,

$$\lim_{k\to\infty} P'^k x_0 = x^*, \quad \forall x_0 \in \mathbb{R}^n,\ \|x_0\|_1 = 1.$$

So, the straightforward way to find x^* is iterational process $x_{k+1} = P'x_k$ with arbitary initial distribution x_0. Then (see [2])

$$\|x_k - x^*\|_2 \le C_1 \exp(-\gamma k / C_2),$$

where $\gamma = |\lambda_1 - \lambda_2|$ is so called spectral gap of P'. There λ_1, λ_2 – two largest eigenvalues. By the way, $\gamma \ge 1 - \alpha = 0.15$ (also see [2]).

Markov chain Monte Carlo

The idea of this approach is to exploit the claim (2) from section 2. Formulating this result in more constructive terms of convergence in probability, if $T > C\gamma^{-1}\ln(n/\varepsilon)$ – number of steps of random surfing on stohastic graph, then with probability not less then $1-\sigma$ it is fulfilled that (see [2])

$$\|f - x^*\|_2 \le C\sqrt{\frac{\ln n + \ln \sigma^{-1}}{\alpha T}}, \tag{3}$$

where f is vector of frequences, defined earlier.

Frank-Wolfe algorithm

Problem of finding a solution of $Ax = 0,\ x \in X$, where $A = P' - I$ and X is the probability simplex, can be considered as the convex optimization problem

$$f(x) = \frac{1}{2}\|Ax\|_2^2 \to \min_{x\in X}.$$

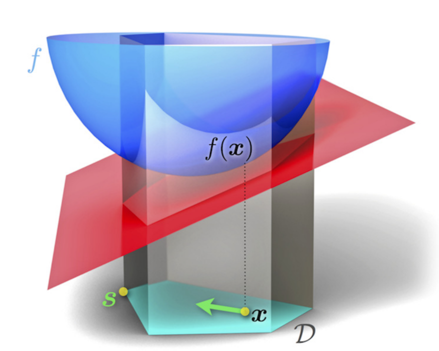
Let's apply Frank-Wolfe to solve it! Frank-Wolfe algorithm solves convex optimization problem iteratively with respect to limitations, doing steps

$$x_{k+1} = x_k + \gamma_k(y_k - x_k),$$

where $\gamma_k = \frac{2}{k+2}$ or can be chosen using line search and y_k is computed as follows

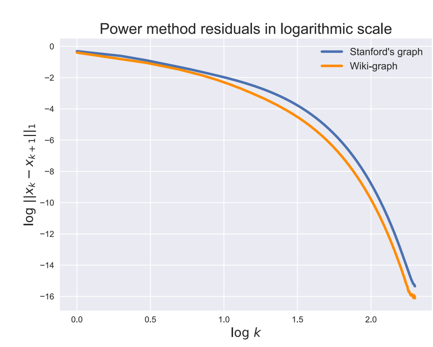
$$y_k = \operatorname*{argmin}_{x\in X}(\nabla f(x_k), x). \tag{4}$$

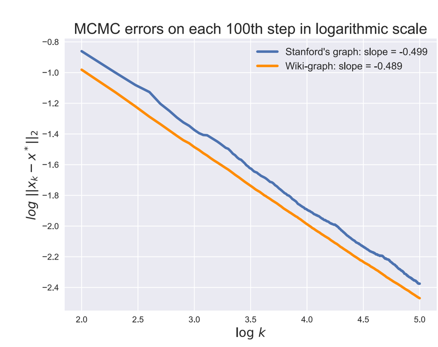
Since gradient on k-th step is a fixed vector $\nabla f(x_k) = A^T A x_k$ and x belongs to the probability simplex, problem (4) is a linear programming problem. As known, the solution of such problem is one of the vertices of the simplex, which are vectors e_i with one nonzero element equal to 1 on i-th position. So, the problem (4) reduces to finding the index of the minimal element in $\nabla f(x_k)$. It can be shown that $|f(x_k) - f(x^*)| = O(\frac{1}{k})$.

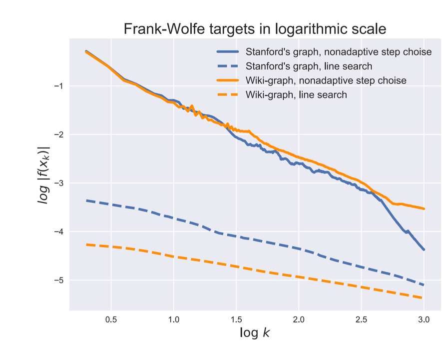


Experiments with real graphs

All of three methods were tested on two web-graphs. First graph was taken from http://snap.stanford.edu/data/#web and the second one, which nodes were papers from English wikipedia, was kindly given to me by my supervisor for this project. At first, power method was applied and it was confirmed that it ensured the exponential convergence to the solution (see figure 1). So, the final result after 200 iterations was adopted as the right solution x^*. Then MCMC procedure was carried out and order $-1/2$ of convergence from (3), section 4, was obtained (see figure 2). Finally, Frank-Wolfe algorithm was tested. But, as was mentioned, rate of convergence in terms of functional values is $O(\frac{1}{k})$, so to achieve 10^5 times better then initial accuracy (as we had using power method) algorithm needs to do 10^5 iterations. Such a number of iterations is justified in case if each iteration is cheap, and, actually, it could be implemented as cheap, but we decided to postpone this problem to the futher work. So, as for Frank-Wolfe algorithm, we can only see that it gradually (linearly in logarithmic scale) converges to the solution (see figure 3).







Below are top 5 articles in English Wikipedia, according to PageRank. All code can be found by https://github.com/migonch/page_rank.

Article	Power Method	MCMC	Frank-Wolfe
United States	0.002905	0.00327	0.004501
Multimedia	0.002068	0.00226	0.002472
Geographic coordinate system	0.001544	0.00170	0.002145
France	0.001356	0.00138	0.002190
Americans	0.001286	0.00139	0.002200

Conclusion an acknowledgements

In conclusion, we note that the problem is very mathematically meaningful. Therefore, it requires careful consideration and therefore there are so many different approaches to its solution. We succesfully examined three approaches and tested them. This project was released thanks to Daniil Merkulov, who motivated me to create it and to Alexander Gasnikov, who prompted the idea and provided me with all the necessary literature.

References

[1] Б. Поляк, А. Тремба, *Решение задачи PageRank для больших матриц с помощью регуляризации*, 2012, (pdf).

[2] А. Гасников, М. Жуковский, *Вокруг степенного закона распределения компонент вектора PageRank*, 2017, https://arxiv.org/abs/1701.02595.

[3] A. Anikin, A. Gasnikov, *Efficient Numerical Methods to Solve Sparse Linear Equations with Application to PageRank*, 2018, https://arxiv.org/abs/1508.07607.