Open Problem: Running time complexity of accelerated ℓ_1-regularized PageRank

Kimon Fountoulakis KIMON.FOUNTOULAKIS@UWATERLOO.CA
David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, Canada.

Shenghao Yang SHENGHAO.YANG@UWATERLOO.CA
David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, Canada.

Editors: Po-Ling Loh and Maxim Raginsky

Abstract

Who reads the search results beyond the first page in Google Search? An exaggerated answer to this question is nearly nobody. This motivates approximate personalized ranking of web-pages that are only immediately relevant to a user instead of ranking all web pages that exist. The results in Google Search, Twitter and other popular search engines traditionally utilize the Personalized PageRank (PPR) vector to rank the results in their search engines. Additionally, there is a plethora of applications beyond the web (Gleich, 2015) which are modelled using PPR. In recent work by Andersen et al. (2006); Gleich and Mahoney (2014); Fountoulakis et al. (2019), it was shown that small probabilities in PPR vector, e.g., web pages beyond the first page in Google Search, can be thresholded out automatically by utilizing ℓ_1-regularization or equivalently by early termination. Both versions result in approximate computation of PPR. The current fastest method for computing the ℓ_1-regularized PPR uses proximal gradient method and requires $\tilde{\mathcal{O}}((\alpha\rho)^{-1})$ total running time, where α is the teleportation parameter and ρ is a parameter which controls the level of sparsity in the ℓ_1-regularized PPR. It is important to note that the running time complexity does not depend on the size of the underlying graph (e.g. the length of the PPR vector). Such property has become a prerequisite to probe modern large scale networks. A seemingly natural way to build an even faster algorithm for computing the ℓ_1-regularized PPR is to accelerate the proximal gradient method and consequently reduce the running time complexity to $\tilde{\mathcal{O}}((\sqrt{\alpha}\rho)^{-1})$. This will lead to a speed-up by a factor of $1/\sqrt{\alpha}$ and improve the running time of various network analytic methods which build upon PPR. However, the original analysis of the proximal gradient method in Fountoulakis et al. (2019) does not apply to the accelerated version. While we have empirical evidence that indicates accelerated proximal gradient requires less total running time, it is not even clear if acceleration would not lead to a worse running time complexity in the worst case.

Keywords: PageRank, convex optimization, network analysis

1. Introduction

Consider an undirected and unweighted graph $G = (V, E)$ where V is a set of n nodes and E is a set of m edges. Also, consider a seed node $v \in V$. The Personalized PageRank (PPR) vector assigns values to the nodes such that nodes that are "close" to the seed node get higher values. The PPR vector can be computed by solving the following linear system

$$x = \alpha s + (1 - \alpha)Wx, \tag{1}$$

where $\alpha \in (0, 1]$ is the teleportation parameter, $s \in \mathbb{R}^n$ is an indicator vector that is equal to 1 for the seed node v and zero elsewhere. W is a random walk transition matrix. We will consider the

lazy random walk transition $W = \frac{1}{2}\left(I + AD^{-1}\right)$. Normally, W is defined as $W = AD^{-1}$, our choice of lazy random walk is done for convenience. It is known that lazy random walk version of PPR is equivalent to the standard random walk version of PPR up to a change in α (Andersen et al., 2006). In particular, let $x(\alpha)$ be the PPR vector in (1) with $W = \frac{1}{2}\left(I + AD^{-1}\right)$ and $\tilde{x}(\alpha)$ be the PPR vector in (1) with $W = AD^{-1}$, we have $x(\alpha) = \tilde{x}\left(\frac{2\alpha}{1+\alpha}\right)$. The linear equation in (1) can be also thought as the eigenvector with eigenvalue equal to 1 of the personalized transition matrix $\alpha s\mathbf{1}^T + (1-\alpha)W$ where $\mathbf{1}$ is the all-ones vector of length n.

1.1. Notations

We assume without loss of generality that nodes are ordered from 1 to n. We denote by d_i the degree of node i, which equals the number of neighbors of node i. We denote by $D \in \mathbb{R}^{n\times n}$ the diagonal matrix of degrees, i.e., $D_{ii} = d_i$. We define

$$Q := D^{-1/2}\left\{D - \frac{1-\alpha}{2}(D+A)\right\}D^{-1/2},$$

and

$$f(x) := \frac{1}{2}x^T Q x - \alpha x^T D^{-1/2} s. \tag{2}$$

The matrix Q is also equivalent to $Q = \alpha I + \frac{1-\alpha}{2}L$, where $L := I - D^{-1/2}AD^{-1/2}$ is the symmetric normalized Laplacian matrix. One can easily see that the gradient ∇f is 1-Lipschitz continuous w.r.t. the ℓ_2-norm. Moreover, f is α-strongly-convex with respect to the ℓ_2-norm.

2. ℓ_1-regularized Personalized PageRank

In the seminal work of Andersen et al. (2006) it was shown that one can compute the PPR vector (1) approximately and still recover approximately the values for nodes that are "close" to the seed node. Moreover, in Andersen et al. (2006) it was shown that small probabilities in the PPR vector can be thresholded to zero automatically, without even having to compute them. This property is crucial because it implies that the running time of the approximate PPR (APPR) algorithm finds an approximate PPR vector in $\tilde{\mathcal{O}}((\alpha\rho)^{-1})$ time and the output vector has at most $1/\rho$ non-zero entries, where ρ is a tolerance parameter of APPR.

In Fountoulakis et al. (2019) it was shown that APPR can be replaced with ℓ_1-regularization. In particular, instead of applying the APPR to solve (1) approximately, one can solve the ℓ_1-regularized PPR problem

$$\text{minimize} \quad \rho\alpha\|D^{1/2}x\|_1 + \frac{1}{2}x^T Q x - \alpha x^T D^{-1/2} s \tag{3}$$

using proximal gradient descent. It was shown in Fountoulakis et al. (2019) that the ℓ_1-regularized PPR problem also maintains the values of nodes that are "close" to the seed node and at the same it sets nodes with very small probability in the PPR vector (1) to zero. More importantly, it was shown in Fountoulakis et al. (2019) that the solution of ℓ_1-regularized PPR and the output of APPR have the same guarantees for the local graph clustering problem. Moreover, it was shown that a black-box proximal gradient descent algorithm can solve the ℓ_1-regularized PPR problem in $\tilde{\mathcal{O}}((\alpha\rho)^{-1})$ time. The proof of the running time result relies on showing that the standard proximal gradient method of (3) updates at most $1/\rho$ number of coordinates (**instead of** n **number of coordinates**) at

each iteration. Then the worst-case running time bound follows from standard analysis of proximal gradient descent (Beck and Teboulle, 2009) that the iteration complexity to obtain an ϵ-accurate solution is $\mathcal{O}\left(\frac{1}{\alpha}\log\frac{1}{\epsilon}\right)$.

3. Acceleration and Open Question

If we apply accelerated gradient methods, e.g., FISTA (Beck and Teboulle, 2009), Linear Coupling (Allen-Zhu and Orecchia, 2017), to solve (3), the iteration complexity is reduced to $\mathcal{O}\left(\sqrt{\frac{1}{\alpha}}\log\frac{1}{\epsilon}\right)$ for obtaining an ϵ-accurate solution. Unfortunately, since there is no known upper bound on the number of coordinates updated by the accelerated methods at each iteration, we only know of a pessimistic running time bound $\mathcal{O}\left(n\sqrt{\frac{1}{\alpha}}\log\frac{1}{\epsilon}\right)$ for accelerated methods. This bound easily becomes worse than the unaccelerated version if $1/\rho < n$. In fact, for most practical applications such as web-page ranking and local community detection, we usually have $1/\rho \ll n$ or even $1/\rho = O(1)$. On the other hand, a series of numerical experiments in Hu, Chufeng (2020) (see, e.g., Figure 5.1 and Figure 5.2 thereof) show that accelerated methods update at most δ/ρ number of coordinates at every iteration, where $\delta \leq 2$. These simulations suggest that while accelerated methods update more number of coordinates at every iteration, this number remains a constant times $1/\rho$ as opposed to growing with n. If this was true, then accelerated methods would enjoy a faster running time complexity $\tilde{\mathcal{O}}((\sqrt{\alpha}\rho)^{-1})$ and consequently speed up numerous graph analytic methods which build on PPR. Furthermore, an affirmative answer will not only improve the currently fastest algorithm for approximately computing PPR, but also likely inspire more algorithmic developments that apply numerical optimization techniques to solve graph-based learning problems scalably.

References

Zeyuan Allen-Zhu and Lorenzo Orecchia. Linear coupling: An ultimate unification of gradient and mirror descent. In *8th Innovations in Theoretical Computer Science Conference*, 2017.

R. Andersen, F. Chung, and K. Lang. Local graph partitioning using pagerank vectors. *FOCS '06 Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, pages 475–486, 2006.

A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sciences*, pages 183–202, 2009.

Kimon Fountoulakis, Farbod Roosta-Khorasani, Julian Shun, Xiang Cheng, and Michael W Mahoney. Variational perspective on local graph clustering. *Mathematical Programming*, 174(1-2): 553–573, 2019.

D. F. Gleich. Pagerank beyond the web. *SIAM Review*, 57(3):321–363, 2015.

D. F. Gleich and M. W. Mahoney. Anti-differentiating approximation algorithms: A case study with min-cuts, spectral, and flow. In *Proceedings of the 31st International Conference on Machine Learning*, pages 1018–1025, 2014.

Hu, Chufeng. Local graph clustering using l1-regularized pagerank algorithms. Master's thesis, 2020. URL http://hdl.handle.net/10012/15815.