Efficient and Accurate PageRank Approximation on Large Graphs

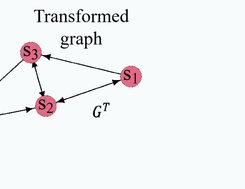
Siyue Wu, Dingming Wu, Junyi Quan, Tsz Nam Chan, Kezhong Lu.
College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China.

Introduction

PageRank was originally proposed to evaluate the importance of the web pages in search engines, and the PageRank algorithm is the key component of why Google has been so successful. Nowadays, PageRank is used to measure the importance of vertices not only for web search engines (e.g.,Google) now, but also other graphs, such as social networks, recommendation systems.

However, there are some challenges of computing PageRank, which make it hard for the PageRank to be widely used.

- **Inefficient:** Exact computation is consuming even on small graphs. The estimations based on iteration or Monte Carlo are not efficient enough to cope with the rapid growth of the real-world graph size.
- **Inaccurate:** The estimations based on sampling can deal with the inefficiency, but they perform poorly in terms of accuracy or have a loose error bound.
- **Partial Estimation:** Part of estimations based on graph or matrix sampling to accelerate the estimation, which will only return the result of sampled vertices.



Novelty

- Summarize PageRank estimation methods based on sampling (matrix sampling or graph sampling) for the first time.
- Propose a generalized low-rank matrix approximation PageRank estimation framework, called **Transformation model**, which combines matrix low-rank approximation technique with random walks and allows any low-rank approximation algorithm to be incorporated to estimate PageRank efficiently (e.g,CUR, constant SVD, truncated SVD).

- Form **CUR-Trans** PageRank estimation algorithm by applied CUR low-rank approximation algorithm to transformation model, because CUR low-rank approximation is currently the most effective method in terms of both efficiency and accuracy.
- Propose a method using matrix multiplication approximation to accelerate PageRank estimation based on iteration, which is named **T^2-Approx** PageRank estimation algorithm. CUR-Trans and T^2-Approx can effectively balance the efficiency, accuracy, and can output full-set estimation result, other algorithms have not been able to achieved satisfactorily.

Solutions

Supposed that T is the transition matrix of the graph, the formulation of PageRank is

$$\pi = (1-\alpha)\cdot\left(\sum_{k=0}^{\infty}\alpha^k T^k\right)\cdot\pi_0, \tag{1}$$

where $\pi_0 = [1/n]_{n\times 1}$ is the initial PageRank vector. From the Equation 1, the most consuming part of the PageRank computation is T^k, the continuous multiplication of high-dimensional matrices. Therefore, the main idea of our work is converting, "how to convert the PageRank computation on large graph to that on the smaller graph" or " how to convert the multiplication of large matrix to that of smaller matrix". Based on the idea about converting, we proposed Transformation model to estimate PageRank, which is shown in Figure 1.

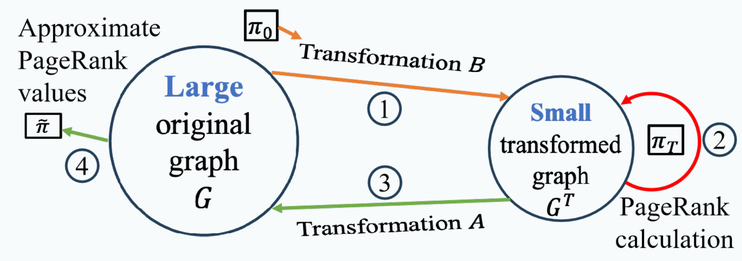


Figure 1. Transformation model.

Figure 2. An example of CUR-Trans.

(1) CUR-Trans: approximation of T^1, accurate and efficient.
If matrices $A_{n\times c}$ and $B_{c\times n}$ ($c \ll n$) exist, such that $T_{n\times n} \approx A_{n\times c}\cdot B_{c\times n}$. We have

$$\underline{T^k_{n\times n}} \approx (A\cdot B)^k \approx A\cdot(B\cdot A)^{k-1}\cdot B \approx A\cdot \underline{D^{k-1}_{c\times c}}\cdot B$$
$$\Longrightarrow \pi \approx (1-\alpha)\cdot\left(\sum_{k=0}^{\infty}\alpha^k(A\cdot B)^k\right)\cdot\pi_0$$
$$\Longrightarrow \pi \approx (1-\alpha)\cdot\pi_0 + A\cdot\alpha(1-\alpha)\cdot\left(\sum_{k=1}^{\infty}\alpha^{k-1}D^{k-1}\right)\cdot B\cdot\pi_0.$$

- **Operation:** Using CUR low-rank approximation, we can obtain matrix $C_{n\times c}$, $U_{c\times r}$ and $R_{r\times n}$, such that $T \approx C\cdot U\cdot R$. Let $A = C\cdot U$, $B = R$ and $D = B\cdot A$, and an example of the algorithm has been shown in Figure 2.
- **Error bound:** $\|T - C\cdot U\cdot R\|_F \le (1+\phi)\|T - T_\rho\|_F$, where ϕ is a constant and T_ρ is the best rank-ρ low rank approximation of T.
- **Time complexity:** Because $c \ll n$, the computation cost will be reduced. Finding A, B and D costs $O\left((c\cdot r + c^2)\cdot n + r^2c + r^3 + c^2r + c^3\right) = O(n)$, and the iteration cost is $O(k\cdot c^2)$, $k = O(\log c + \log_\alpha \tau)$, where τ, c and r is constants which are much smaller than n, so it is $O(1)$.

(2) T^2-Approx: approximation of T^2, more efficient.
If matrices $X_{n\times c}$ and $Y_{c\times n}$ ($c \ll n$) exist, such that $T_{n\times n}\cdot T_{n\times n} \approx X_{n\times c}\cdot Y_{c\times n}$. We have

$$T^k_{n\times n} \approx (X\cdot Y)^k \approx X\cdot(Y\cdot X)^{k-1}\cdot Y \approx X\cdot Z^{k/2-1}_{c\times c}\cdot Y$$
$$\Longrightarrow \pi \approx (1-\alpha)\cdot\left(\sum_{k=0}^{\infty}\alpha^k(X\cdot Y)^{\frac{k}{2}}\right)\cdot\pi_0$$
$$\Longrightarrow \pi \approx (1-\alpha)\pi_0 + \frac{\alpha}{1+\alpha}\cdot X\cdot\left((1-\alpha^2)\cdot\sum_{k=1}^{\infty}(\alpha^2)^{\frac{k-1}{2}}Z^{\frac{k-1}{2}}\right)Y\cdot\pi_0.$$

- **Operation:** The Monte-Carlo matrix multiplication conversion method can be used to find X and Y, then let $Z = Y\cdot X$, and an example of the algorithm has been shown in Figure 3.
- **Error bound:** $\|T^2 - X\cdot Y\|_F \le \sqrt{(n-c)}\|T\|_F^2$
- **Time complexity:** Because $c \ll n$, the computation cost will be reduced. Finding X, Y and Z costs $O\left(c^2\cdot n\right) = O(n)$, and the iteration cost is $O(k\cdot c^2)$, $k = O(\log c + \log_\alpha \tau)$, where τ, c and r can be seen as constants which are much smaller than n, so it is $O(1)$.

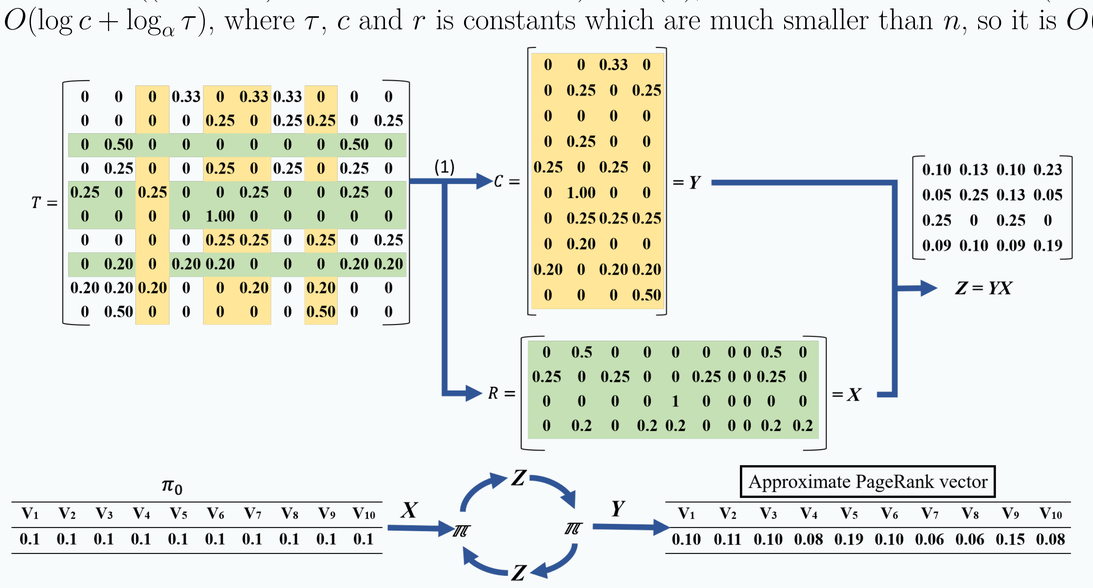


Figure 3. An example of T^2-Approx.

Experimental results

Efficiency: Our algorithms are much more efficient then open source systems on the PageRank computation shown in Table 1. Compared with other sampling-based estimations shown in Figure 4, T^2-Approx is the fastest one. CUR-Trans and ApproxRank are both the second fastest, but CUR-Trans is much accurate than ApproxRank. Because of space limitation, we on present results on the dataset Orkut, and other results can be found in our paper.

Table 1. Time of our algorithms and open source systems.

	Orkut	Friendster	UKDomain
Networkit	11 seconds	535 seconds	275 seconds
GraphX	46 minutes	3+ hours	3+ hours
Giraph	11 minutes	3+ hours	3+ hours
CUR-Trans(0.1%)	4 seconds	198 seconds	129 seconds
T^2-Approx (0.1%)	5 seconds	215 seconds	93 seconds

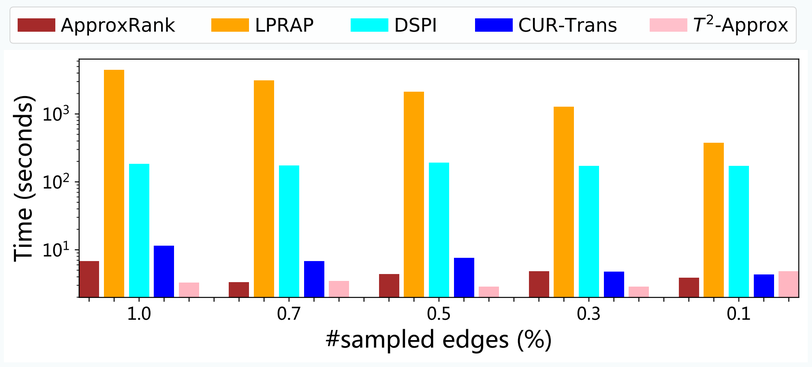


Figure 4. Time of different sampling-based PageRank estimations on Orkut.

Accuracy: Compared with other sampling-based estimations in Figure 5, CUR-Trans is the most accurate, and T^2-Approx is comparable with others. NDCG is a measurement of the accuracy of ranking, and CUR-Trans is the second most accurate, while DSPI is much less efficient than CUR-Trans, and T^2-Approx is comparable with others.

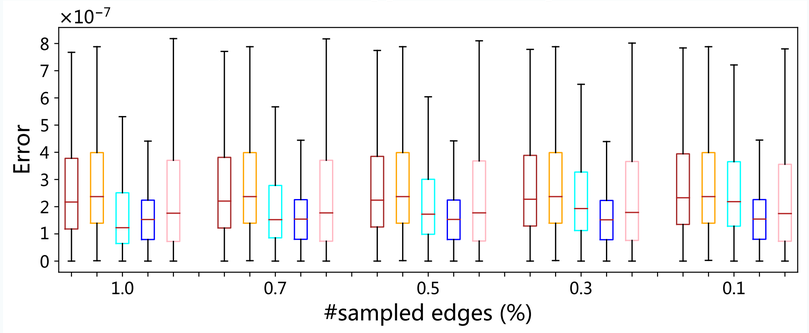


Figure 5. Errors of each vertices under different sampling-based PageRank estimations on Orkut.

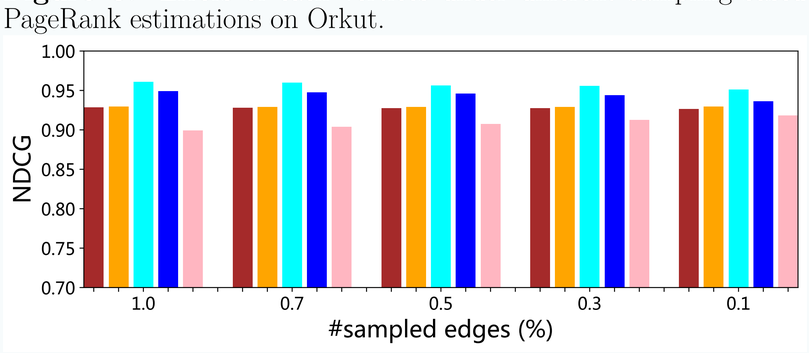


Figure 6. NDCG of different sampling-based PageRank estimations on Orkut.

Full-set Estimation: As we mention above, sampling-based estimations perform graph sampling to accelerate the estimation, which will only return the result of sampled vertices. Our proposed methods can return estimation of all vertices.

Table 2. The amount of returned vertices with PageRank values.

	Orkut	Friendster	UKDomain
ApproxRank	2.00%–9.00%	2.00%–9.00%	2.00%–9.00%
LPRAP	0.03%	0.002%	0.001%
DSPI	8.90%–59.19%	12.68%–64.91%	13.29%–55.32%
CUR-Trans	100%	100%	100%
T^2-Approx	100%	100%	100%

Summary: As the experimental results shown above, T^2-Approx is much faster than others with comparable accuracy, while CUR-Trans is the second most efficient with the best accuracy. Moreover, only our two methods can return estimations of all vertices. More details can be found in our paper and codes, and please scan the QR codes to obtain.

Paper

Code