Parallel Performance Analysis for CUDA-Based Co-rank Framework on Bipartite Graphs Heterogeneous Network

Fang Zheng
College of Informatics
Huazhong Agricultural University
Wuhan, China
zhengfang@mail.hzau.edu.cn

*Han Tan
Department of Mechanical Engineering
Wuhan Vocational College of Software and Engineering
823183817@qq.com

*Fang Tian
College of Informatics
Huazhong Agricultural University
Wuhan, China
Fangzhihai_2003@mail.hzau.edu.cn

***Abstract*—The Co-rank is a ranking algorithm based on bipartite graphs of heterogeneous networks, which has been extensively studied and employed in the past decades. The Co-rank algorithm can utilize all kinds of objects in heterogeneous networks. However, the computational complexity of Co-rank algorithm limits its application on large scale datasets. Considering the efficiency of computation, we parallelize the algorithm with CUDA. We demonstrate our method via large-scale experiments across drug-target datasets and obtain excellent speedup. The results show that the optimization effect of Co-rank on the GPU platform is obvious.**

Keywords-CUDA; parallel computing; Co-Rank Framework; Bipartite Network.

I. INTRODUCTION

The social nature of the network makes it contain a large number of heterogeneous network resources. There are at least two or more types of vertices and edges in a heterogeneous network. Compared to the network which has only one type of node, heterogeneous networks can bring more information. Heterogeneous network models are widely used in Web services [1-3], social networks [4-5] and biological networks [6-7], which are used to predict unknown information.

The Co-rank is a ranking algorithm based on bipartite graphs of heterogeneous networks, which has been extensively studied and employed in the past decades. The Co-rank algorithm can utilize all kinds of objects in heterogeneous networks. For example, [8] used Co-rank algorithm predicting authors and their publications ,and [9] propose a bi-type ranking algorithm to joint documents and authors in a bibliographic network , [10] is proposed to determine the importance of the objects and relationships of multiple relational data at the same time.

Due to the complexity of heterogeneous networks, the complexity of computation increases linearly with the number of network nodes, which brings more challenges to computation. In order to improve the efficiency of data processing, many parallel methods are proposed for heterogeneous networks. For example, [11-13] have proposed parallel computing model for PageRank and random walk. [14] provided an implementation of CUDA parallel to PageRank. [15] proposed parallel scheme of SimRank in Mapreduce.

Single core computing power is weaker than the CPU in the Many Integrated Cores, but the concurrency is much higher than the CPU. Some graphic processing chip manufacturers have found that the hardware features of GPU make it very suitable for parallel computing with high concurrency [16-17]. Therefore, GPGPU provides an important means for parallel computing in the way of coprocessor. GPU has been used in many fields of scientific computing. Such as computational fluid dynamics [18-19], bioinformatics [20-21], etc. Co-rank can be regarded as an iterative matrix vector multiplication, and CUDA is especially good at such calculations.

In this research, we design a Co-rank network to describe drug-target and predict the missing edges between drug and target. As the number of nodes in the network increases, the computational complexity of the algorithm increases proportionally. Considering the efficiency of computation, we parallelize the algorithm. Therefore, we do some parallelization on the Co-rank drug-target algorithm of hybrid heterogeneous network based on CUDA.

II. METHOD

A. Bipartite Heterogeneous Network Model of Drug-Target

A heterogeneous network contains two networks with different types of nodes and edges, as well as a bipartite graph containing bipartite associations between them [22]. Suppose the graph $G_D = (V_D, E_D)$ with $V_D = \{v_{d1}, v_{d2}, \dots, v_{dn}\}$, $G_T = (V_T, E_T)$ with $V_T = \{v_{t1}, v_{t2}, \dots, v_{tm}\}$. And the bipartite graph connecting G_D and G_T is $G_B = \{V_D \cup V_T, E_B\}$ with $E_B = V_D \times V_T$. The bipartite heterogeneous network can be defined of $G_B = (V_B, E_B)$ as: $V_B = \{V_D \cup V_T\}$, $E_B = \{E_D \cup E_T \cup E_{DT}\}$

B. Construction of Heterogeneous Network

This heterogeneous network was composed of a drug-drug similarity network, a target network, and a bipartite graph containing drug-target associations. The walk on the above three networks, four state transition probability matrices (P_D , P_T , $P_{DT(n\times m)}$ and $P_{TD(m\times n)}$) need to establish for jump between the two networks.

We define $A_{D(n\times n)}$, $A_{T(m\times m)}$, A_B as their respective adjacency matrix, then the transition probabilities can be computed based on adjacency matrix.

978-1-5386-7445-1/18/$31.00 ©2018 IEEE
DOI 10.1109/DCABES.2018.00014

(1) Drug-drug similarity network: P_D

Let $A_{D(n\times n)}$ be the adjacency matrix of a drug network with n number of nodes. $P_{D(i,j)}$ represent the probability of transition from node i to node j. The calculation of $P_{D(i,j)}$ is given as Equation (1).

$$P_{D(i,j)}=\begin{cases} A_{D(i,j)}/\sum_{k=1}^{n} A_{D(i,k)} & if\ e(i,j)\in E \\ 0, & otherwise \end{cases} \quad (1)$$

(2) Target-target interactive network: P_T

$A_{T(m\times m)}$ is the adjacency matrix of a target network with m number of nodes, if there is a interaction between t_i and t_j ,then $A_{T(i,j)}=1$. The probability of transition matrix of target-target interactive network is given by Equation (2).

$$P_{T(i,j)}=\begin{cases} A_{T(i,j)}/\sum_{k=1}^{m} A_{T(i,k)} & if\ e(i,j)\in E \\ 0, & otherwise \end{cases} \quad (2)$$

(3) Inter-network transfer matrix: P_{DT} and P_{TD}

A_B is the adjacency matrix of the drug-target bipartite network as Equation (3) , P_{DT} is the transition matrix from the drug network to the target network.

$$A_{B(i,j)}=\begin{cases} 1 & if\ d_i\ is\ related\ with\ t_j \\ 0, & otherwise \end{cases} \quad (3)$$

So the probability of transition matrix P_{DT} which can be given by the Equation (4).

$$P_{DT(i,j)}=\begin{cases} A_{B(i,j)}/\sum_j A_{B(j,i)} & if\ \sum_j A_{B(j,i)}\neq 0 \\ 0, & otherwise \end{cases} \quad (4)$$

Similarly, P_{TD} can be given by the Equation (5).

$$P_{TD(i,j)}=\begin{cases} A_{B(i,j)}/\sum_j A_{B(i,j)} & if\ \sum_j A_{B(i,j)}\neq 0 \\ 0, & otherwise \end{cases} \quad (5)$$

C. Co-rank on the drug-target network

Co-rank framework [10] based on the bipartite graph and random walk with restart algorithm. We take drug-target network as an example to describe the Co-rank framework. The Co-rank consists of two types of random walk, one is a random walk of the intra-network. A random walk in drug similarity network G_D and target interaction network G_T. For example, there is a link between node d_i and d_j, which are exactly from G_D. The random walker can move from d_i to its link neighbor d_j node with probability α (α ϵ(0 ,1)) or move back to its home node with the restart probability1-α. The other is a random walk between networks based on the bipartite graph G_B network. If a random walker on the G_D and moves to a bridging node, it can jump to the other network G_T with a probability λ (λ ϵ (0 ,1)) and takes *2k+1*inter-network steps or move back to the other nodes in its home subnet with the probability 1-λ take m intra-network steps walk on G_D. A transition probability matrix is denoted as P in the following Equation (6).

$$P=\begin{bmatrix} (\alpha)P_T^m & (1-\alpha)P_{TD}(P_{DT}P_{TD})^k \\ (1-\alpha)P_{DT}(P_{TD}P_{DT})^k & (\alpha)P_D^n \end{bmatrix} \quad (6)$$

Computational complexity of algorithm is $\mathbf{O}(t*(2k+1)*n_D*n_T)+\mathbf{O}(n*n_T^3+m*n_D^3)$. The computational time complexity of the algorithm is high, so we consider using CUDA to parallelize it.

III. CUDA PARALLEL OF CO-RANK

A. Parallel Scheme

The flow diagram of parallel Co-rank algorithm based on CUDA is shown in Fig.1.The calculation procedure is divided into two parts. The host needs to calculate the transition matrix P_D, P_T, P_{DT}, P_{TD} according to the adjacency matrix A_D , A_T , A_B . Cudamallochost is used to allocate the unified memory for host and device of the transition matrix. The device calculate matrix power for P_D and P_T, and then calculate matrix multiplication for P_{TD} and P_{DT}, so as to obtain the probability matrix P. Then calculate y^t iteratively, and finally copy y from the device back to the host.

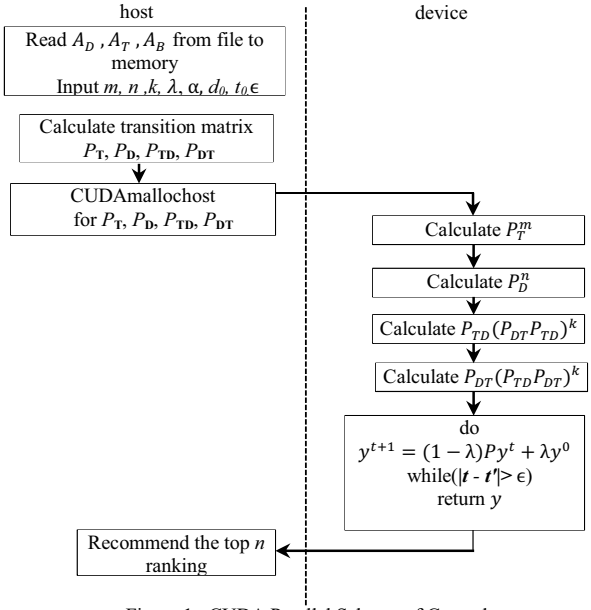

Figure 1. CUDA Parallel Scheme of Co-rank

B. Parallelization of Co-rank based on CUDA

The NVIDIA's CUDA CUBLAS library [23] provides matrix multiplication function. In addition, Cusparse_Dcsrmv function in CUSPARSE library [24] realizes the sparse matrix and vector multiplication function. The specific algorithm process is as follows.

Algorithm process : *Parallelization of Co-rank based on CUDA*

Procedure *Cu_Co-rank(P_D, P_T , P_{DT}, P_{TD}, m, n ,k, λ, α, d_0, t_0,ϵ)*
1: Allocate the variable memory space on GPU device and copy P_D, P_T , P_{DT}, P_{TD}, m, n ,k, λ, α, d_0, t_0 from CPU to GPU;
2: Initialization solution vector d_0, t_0 on GPU;
3: Call Cusparse_powermatrixmul($*P_T$, m);
4: Call Cusparse_powermatrixmul($*P_D$, n);
5: Call CuBiWalk($*P_{DT}$, $*P_{TD}$, k);
6: Call CuBiWalk($*P_{TD}$,$*P_{DT}$,k);
7: ***Repeat***
8: Call *cusparse_csrmv* function compute $(1-\lambda)Py^t+\lambda y^0$
9: **while**(|y - y'|> ϵ)
10: return *y*
11: copy *d, t* from GPU to CPU;
End Procedure

IV. Results and Discussion

In this paper, we used drug and target dataset to test our approach. And we compared our algorithm with RWR (Random Walk Restart) to evaluate accuracy and calculation performance.

A. Experiment environment

The computing platform of this paper is as following: CPU is Intel(R) Core(TM) i7-4790, 3.6GHz, 16GB host memory; GPU is GeForce GTX 850M with 4GB device memory. We use NVIDIA CUDA toolkit v9.0 libraries and GCC 5.4.0 compilation.

B. Dataset

Drug dataset and drug-target bipartite graph dataset were obtained from Drug Bank, DGIDB and TTD databases, which were descripted in document [25]. Drug dataset gives the structure similarity between the 606 drugs. Drug-target bipartite graph dataset contains drug and its corresponding target. We used PPI (Protein-Protein Interaction) as target-target interaction dataset, PPI dataset was obtained from the Human Protein Reference Database (http://www.hprd.org/) which included 39240 interactive information between 9673 proteins. So n of adjacency matrix $A_{D(n\times n)}$ is 606, m of adjacency matrix $A_{T(m\times m)}$ is 9673.

C. Evaluating accuracy

In this paper, we use the Leave- one-out cross validation to evaluate the accuracy of our algorithm.

In order to evaluate the effectiveness of the algorithm in drug target prediction as accurately as possible. We selected drugs which include targets greater than or equal to 2 in the bipartite network as test data. The dataset includes 450 drugs with 2960 corresponding drug-target relationships. In the test, we set the network parameters as follows: m=5, n=6, k=1, λ =0.5, α =0.6. And adopted the following test method[26]. For example, for drug d, its drug targets include target t.

(1) First, removing the corresponding relationship between d and t in the binary graph network G_D and D_G. Then initialing vector d_0 with the remaining known targets of d, running the Cu_Co-rank algorithm to predict. Finally, we obtained the ranking results of all the predicted targets corresponding to drug d.

(2) For each drug d, recording the prediction ranking of its removed target t.

(3) After 2960 groups of tests, the ROC and Recall curves were obtained through statistical calculation of all the test results.

By observing the ROC diagram shown in Fig.2, we find that with the increase of threshold, the prediction accuracy of Cu_Co-rank algorithm has been greatly improved compared with the RWR algorithm. From Fig.3 of the RECALL curve, we find that the results of the Cu_Co-rank algorithm is better than RWR algorithm. When the n is near to 800, the algorithm obtain more than 0.9 accuracy, and the RWR algorithm is only 0.65. The result show that the Cu_Co-rank algorithm was more effective in predicting.

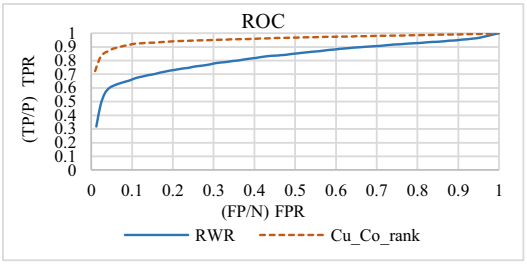

Figure 2. ROC Curve of Cu_Co_rank and RWR

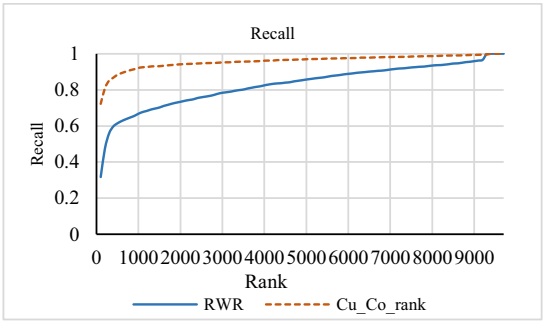

Figure 3. Recall Curve of Cu_Co_rank and RWR

D. Performance and Speedup

(1) Target interaction network calculation

Target interaction network has 9,673 nodes and 3,9240 relationships. This network structure diagram is a typical sparse matrix, so we use sparse matrix to calculate. When the test data set is 5000, the CPU computation time of the test data set is nearly 30 minutes, so the comparison between GPU and CPU acceleration ratio only calculates the data within 5000, and the results are as follows Table 1 and Fig.4.

Table 1. GPU/CPU Speedup and Sparse_GPU/CPU Speedup

Dataset	GPU/CPU Speedup	Sparse_GPU/CPU Speedup
50	0.00313699	0.001478835
100	0.026307628	0.009525391
500	2.643575467	2.937666715
1000	28.24542319	40.70232975
1500	68.32913795	78.91968159
5000	210.0562152	347.9921199

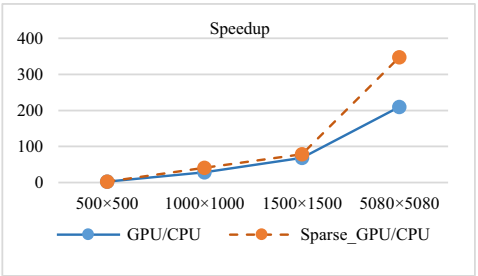

Figure 4. Speedup of GPU/CPU and Sparse_GPU/CPU

(2) The computation results of matrix multiple power

Since the transfer matrix P needs to calculate the multiple power multiplication of P_D matrix and P_T matrix. We test it with GPU and CPU, and the results are shown in below Fig.5.

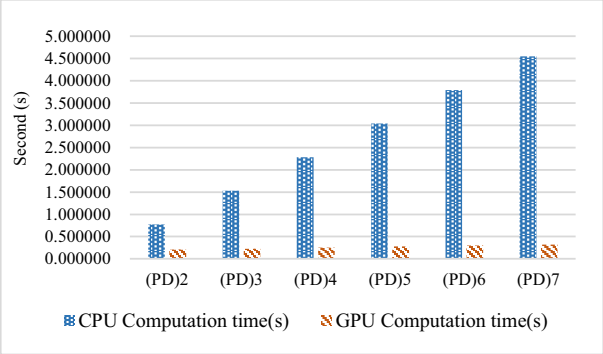

Figure 5. Calculation time of multiple power multiplication for $\boldsymbol{P_D}$ with GPU and CPU

V. Conclusions and Future Work

In this paper, we propose a GPU accelerated Co-rank algorithm based on bipartite graph heterogeneous network. First, we apply the Co-rank algorithm framework to the drug target heterogeneous network, and verify the effectiveness of the algorithm. The ROC and Recall curves of the algorithm are all superior to the RWR algorithm. Secondly, for the computational complexity of Co-rank algorithm on large data sets, we propose a parallel computing model based on CUDA. The test results show that the algorithm has a significant acceleration effect. This makes it possible to apply Co-rank algorithm to large scale datasets in desktop computer.

In future work, we hope to test the algorithm's parameters and performance on larger data sets. In addition, in order to further improve the computational efficiency, we need to optimize the Cu_Co-rank algorithm by data partitioning, sparse storage and texture memory.

Acknowledgment

This study is supported by Natural Science Foundation of Hubei Province of China (Program No. 2015CFB524), the Fundamental Research Funds for the Central Universities (Program No. 2015BQ023).

References

[1] Schattkowsky T, Loeser C. Peer-To-Peer Technology for Interconnecting Web Services in Heterogeneous Networks[C]// International Conference on Advanced Information NETWORKING and Applications. IEEE Computer Society, 2004:611.

[2] Skjervold E, Hafsøe T, Johnsen F T, et al. Delay and disruption tolerant Web services for heterogeneous networks[C]// Military Communications Conference, 2009. Milcom. IEEE, 2009:1-8.

[3] Zhu D, Zhang Y, Chen J, et al. Enhancing ESB Based Execution Platform to Support Flexible Communication Web Services over Heterogeneous Networks[C]// IEEE International Conference on Communications. IEEE Xplore, 2010:1-6.

[4] Dong Y, Tang J, Wu S, et al. Link Prediction and Recommendation across Heterogeneous Social Networks[C]// IEEE, International Conference on Data Mining. IEEE, 2013:181-190.

[5] Huang J, Nie F, Huang H, et al. Trust prediction via aggregating heterogeneous social networks[C]// ACM International Conference on Information and Knowledge Management. ACM, 2012:1774-1778.

[6] Blin G, Fertin G, Mohamed-Babou H, et al. Algorithmic Aspects of Heterogeneous Biological Networks Comparison[J]. 2011.

[7] Li J, Zhao P X. Mining Functional Modules in Heterogeneous Biological Networks Using Multiplex PageRank Approach:[J]. Frontiers in Plant Science, 2016, 7.

[8] Zhou D, Orshanskiy S A, Zha H, et al. Co-ranking Authors and Documents in a Heterogeneous Network[C]// IEEE International Conference on Data Mining. IEEE, 2007:739-744.

[9] Soulier L, Jabeur L B, Tamine L, et al. On ranking relevant entities in heterogeneous networks using a language ‐ based model[J]. Journal of the Association for Information Science & Technology, 2013, 64(3):500-515.

[10] Ng K P, Li X, Ye Y. MultiRank: co-ranking for objects and relations in multi-relational data[C]// ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, Ca, Usa, August. DBLP, 2011:1217-1225.

[11] Das Sarma A, Molla A R, Pandurangan G. Distributed computation in dynamic networks via random walks[J]. Theoretical Computer Science, 2015, 581(C):45-66.

[12] Sarma A D, Molla A R, Pandurangan G. Efficient random walk sampling in distributed networks[J]. Journal of Parallel & Distributed Computing, 2015, 77(C):84-94.

[13] Sarma A D, Molla A R, Pandurangan G, et al. Fast Distributed PageRank Computation[M]// Distributed Computing and Networking. Springer Berlin Heidelberg, 2013:113-121.

[14] Duong N T, Nguyen Q A P, Nguyen A T, et al. Parallel PageRank computation using GPUs[C]// 2012:223-230.

[15] He G, Feng H, Li C, et al. Parallel SimRank computation on large graphs with iterative aggregation[C]// ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2010:543-552.

[16] Programming Guide :: CUDA Toolkit Documentation[OL]. https://docs.nvidia.com/cuda/cuda-c -programming-guide, 2018

[17] Sanders J, Kandrot E. CUDA by Example: An Introduction to General-Purpose GPU Programming [M]. Addison-Wesley Professional, 2010.

[18] Riegel E, Indinger T, Adams N A. Implementation of a Lattice–Boltzmann method for numerical fluid mechanics using the nVIDIA CUDA technology[J]. Computer Science - Research and Development, 2009, 23(3-4):241-247.

[19] Agrawal S, Kumar M, Roy S. Demonstration of GPGPU-accelerated computational fluid dynamic calculations[M]// Intelligent Computing and Applications. Springer India, 2015:519-525.

[20] Bustamam A, Burrage K, Hamilton N A. Fast parallel Markov clustering in bioinformatics using massively parallel computing on GPU with CUDA and ELLPACK-R sparse format[J]. IEEE/ACM Transactions on Computational Biology & Bioinformatics, 2012, 9(3):679.

[21] George J, Gopal G N. Parallelizing Markov Clustering using CUDA and CSR Sparse Format[J]. 2015.

[22] Lee S, Park S, Kahng M, et al. PathRank: Ranking nodes on a heterogeneous graph for flexible hybrid recommender systems[J]. Expert Systems with Applications An International Journal, 2013, 40(2):684-697.

[23] cuBLAS :: CUDA Toolkit Documentation[OL].[2018-03-01]. https://docs.nvidia.com/cuda/cublas/

[24] cuSPARSE :: CUDA Toolkit Documentation[OL].[2018-03-01].https://docs.nvidia.com/cuda/ cu-sparse/

[25] Quan Y, Liu M Y, Liu Y M, et al. Facilitating Anti-Cancer Combinatorial Drug Discovery by Targeting Epistatic Disease Genes.[J]. Molecules, 2018, 23(4):736.

[26] Liu Xiaoyi. Heterogeneous Network Model Based Method for Disease Gene Prediction [D]. XiDian University, 2013.