Journal of Theoretical and Applied Information Technology
10th February 2016. Vol.84. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AVGSIM: RELEVANCE MEASUREMENT ON MASSIVE DATA IN HETEROGENEOUS NETWORKS

[1] DING XIAO, [2] XIAOFENG MENG, [3] YITONG LI, [4]CHUAN SHI, [5]BIN WU

[1]Lecturer, Beijing Key Lab of Intelligent Telecommunication Software and Multimedia, Beijing University of Posts and Telecommunications, China
2Engineer, International Business Corporation, China
3Master Student, Beijing Key Lab of Intelligent Telecommunication Software and Multimedia, Beijing University of Posts and Telecommunications, China
[4]Professor, Beijing Key Lab of Intelligent Telecommunication Software and Multimedia, Beijing University of Posts and Telecommunications, China
[5]Professor, Beijing Key Lab of Intelligent Telecommunication Software and Multimedia, Beijing University of Posts and Telecommunications, China
E-mail: [1] dxiao@bupt.edu.cn, [2] mengxiaofeng0126@gmail.com, [3] yitongglee@gmail.com, [4] shichuan@bupt.edu.cn, [5] wubin@bupt.edu.cn

ABSTRACT

Heterogeneous information network includes multiple types of objects and multiple types of links. Compared with Homogeneous information network which only contains objects of the same type, heterogeneous information network has more abundant semantic information. Heterogeneous information network is very common in our daily life, such as social networks. Similarity search in heterogeneous information network can mine more precise and accurate knowledge. However, real social networks such as Sina Microblog and Facebook have a huge amount of data, which significantly increases the difficulty of similarity search. Unfortunately, many existing methods can only measure similarities between objects of the same type, moreover, the limitation of computing memory size results in quite limited measurable data amount, thus they can't be actually applied to real relation networks. In this paper, we propose a novel measure, called AvgSim, which can measure similarity between objects at the ends of any searching path in heterogeneous information networks. In addition, we apply parallel computing method in the realization of AvgSim in order to enable the handle of massive data and the application in real networks. Experiments on real datasets verify the effectiveness and efficiency of this novel algorithm.

Keywords: *Heterogeneous Information Network, Similarity Search, Random Walk, MapReduce*

1. INTRODUCTION

In recent years, heterogeneous information network analysis has become a hot research topic in data mining field. Different from widely used homogeneous networks which include only same-typed objects or links, Heterogeneous Information Network (HIN) organizes the networked data as a network including different-typed objects and links. It is clear that this kind of networks is ubiquitous and forms a critical component of modern information infrastructure. For example, in the case of bibliographic network, the object types include authors, papers, venues; and links between objects correspond to different relations, such as write relation between authors and papers, and citation relation between papers. Combination of different-typed objects and links results in more comprehensive structure information and richer semantics information. Thus, heterogeneous information network analysis will mine more interesting patterns

Many data mining tasks have been exploited in HIN, such as clustering [1], classification [2]. Among these data mining tasks, similarity measure is a basic and important function, which evaluates the similarity of object pairs in networks. Although similarity measure in homogeneous networks has been extensively studied in the past decades, such as PageRank [3] and SimRank [4], it is just beginning in heterogeneous networks and several measures have been proposed. PathSim [5] is proposed to measure the similarity of same-typed objects based on symmetric paths, and PCRW [6] evaluates the reachable probability along the give path. Recently, Shi et al. proposed the HeteSim [7], which can measure the relatedness of objects with

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the same or different types in a uniform framework. HeteSim has some good properties (e.g., self-maximum and symmetric), and has shown its potential in several data mining tasks.

However, we can also find that it has several disadvantages. (1) HeteSim has relatively high computational complexity, in particular, the adoption of path decomposition approach while measuring the relevance on odd-length path further increases complexity of calculation. (2) Besides, HeteSim cannot be extended to large-scale network with massive data, since its calculation process is based on memory computing. Therefore, it is desired to design a new similarity measure, which not only contains some good properties of HeteSim (e.g., symmetric and uniform similarity framework for heterogeneous objects), but also overcomes the disadvantages of computation.

In this paper, we propose a new relevance measure method - **AvgSim**, which is a symmetric and uniform measure to evaluate the relevance of same or different-typed objects. Since AvgSim can also measure the relevance of different-typed objects, we use the relevance measure instead of similarity measure in the following sections. AvgSim value of two objects is the average of reachable probability under the given path and the reverse path. It guarantees that AvgSim can measure relevance of same or different-typed objects as well as satisfying the symmetric property. In addition, compared with HeteSim which takes a pair-wise random walk, AvgSim doesn't need to consider the length of path and there is no path decomposition involved. Thus, it's more simple and efficient. Furthermore, we take parallelization of this new algorithm on MapReduce in order to eliminate restriction of memory size and deal with massive data more efficiently in practical applications. Experiments on real dataset show that AvgSim can achieve comparative performances with high efficiency and effectiveness, compared with other methods including HeteSim, PathSim and PCRW. Moreover, experiments on large-scale dataset also validate the effectiveness of parallelized AvgSim.

The rest of this paper is organized as follows: Section 2 reviews some concepts and related works briefly, and detailed description of AvgSim is presented in Section 3. The parallelization of AvgSim is explained in Section 4. Section 5 analyzes the experiment results of AvgSim to validate its effectiveness and efficiency. Matrix parallelization experiments are also in this section. Finally, we conclude this paper in Section 6.

2. RELATED WORK

In this section, we will review some concepts and notable methods. We mentioned several path-based similarity measure methods in the previous section, such as PathSim, PCRW and HeteSim. All of these methods are based on meta path which describes semantic relations, and measure similarity in HIN. Meta path is an effective tool of mining semantic relationship, and different meta paths will produce different data mining results, such as two meta paths connecting authors: "Author-Paper-Author" represents coauthors of the same paper, while "Author-Conference-Author" represents that authors publish papers in the same conference. A **Meta Path** P is defined as the form of $A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \cdots \xrightarrow{R_l} A_{l+1}$,where A_1 to A_{l+1} are heterogeneous or homogeneous objects and R_1 to R_l are multiple-typed relations between objects. We define $R = R_1 \circ R_2 \circ \cdots R_l$ as a composite relation where $\circ$ is the composition operator on relations and R^{-1} is the inverse relation of R.

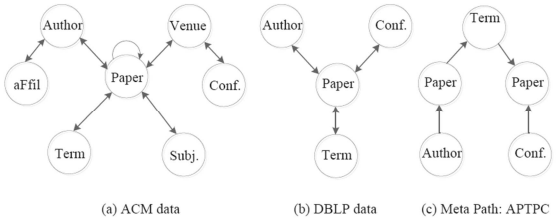


Figure 1: Bibliographic Network Schema And Meta Path Example.

The bibliographic network schemas of ACM and DBLP dataset are shown in Fig.1(a) and Fig.1(b). We take meta path APTPC on DBLP dataset which contains four types of objects: papers (P), conferences (C), authors (A), and terms (T) as an example. In Fig.1(c), authors and conferences can be connected via "Author-Paper-Term-Paper-Conference" (APTPC) path, which means that conferences publishing the papers which have some same terms as the authors' papers. And the semantic relation can be described as $A \xrightarrow{write} P \xrightarrow{contain} T \xrightarrow{contained} P \xrightarrow{published} C$. Thus, we can measure the relevance between authors and conferences based on APTPC.

PathSim is a method to measure similarity of same-typed objects based on symmetric paths in HIN proposed by Sun et al. However, the property of measuring on symmetric paths restricts its application range because lots of meaningful paths in HIN are asymmetric. What's worse, PathSim can only measure similarity between same-typed objects, which also brings constraints since

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

similarity measure for different-typed objects is more valuable. PCRW (Path Constrained Random Walk) is a path-based method to measure reachable probability along the given path in directed graph proposed by Lao and Cohen. The method can measure similarity of different-typed objects without restriction on path type. But the asymmetric property of it restricts its application. Shi Chuan et al. proposed a pair-wise random walk method named HeteSim which has symmetric property to measure similarity of arbitrary node pairs in heterogeneous network. However, HeteSim has relatively high complexity. Especially when dealing with odd-length meta paths, it applies path decomposition which further increases complexity of calculation. Besides, the memory computing implementation of HeteSim restricts its application in large-scale network. Considering the disadvantages of HeteSim and other similarity measures, the new relevance measure we desired should contain the following properties: (1) Ability to measure relevance between arbitrary same or different-typed objects based on meta path in HIN. (2) Symmetric property. (3) Ability to be applied in large-scale networks with high efficiency and effectiveness.

3. AVGSIM: A NOVEL RELEVANCE MEASURE

Through deeply study and analysis of similarity measures in heterogeneous information network showed in Section 2, we are inspired to design a new meta path based method which not only contains good properties (e.g., symmetric and uniform similarity framework for heterogeneous objects), but also has the ability to be extended to large-scale networks. The new relevance measure we proposed is called **AvgSim** and the definition of it is as follows.

Definition 1: AvgSim. Given a meta path P which is defined on the composite relation R=$R_1 \circ R_2 \circ \cdots R_l$, AvgSim between two objects s and t (s is the source object and t is the target object) is:

$$AvgSim(s,t|P) = \frac{1}{2}[RW(s,t|P) + RW(t,s|P^{-1})] \quad (1)$$

$$RW(s,t|R_1 \circ R_2 \circ \cdots R_l) = \frac{1}{|O(s|R_1)|}\sum_{i=1}^{|O(s|R_1)|}(O_i(s|R_1),t|R_2 \circ \cdots R_l) \quad (2)$$

Equation (1) shows that the relevance of source object and target object based on meta path P is the arithmetic mean value of random walk result from s to t along P and reversed random walk result from t to s along P^{-1}. Equation (2) shows the decomposed step of AvgSim, namely the measure of random walk. The measure takes a random walk step by step from starting point s to end point t along path P using iterative method, where $|O(s|R_1)|$ is the out-neighbors of s based on relation R_1. If there is no out-neighbor of s on R_1, then the relevance value of s and t is 0 because s cannot reach t. We need to calculate random walk probabilities for each out-neighbor of s to t iteratively, and then sums them up. Finally, the summation should be normalized by the number of out-neighbors to get average relatedness. The stop sign of iteration is that s meets t at t node along P. In contrast to simple random work method, AvgSim shows its comprehensiveness and effectiveness in later experiments.

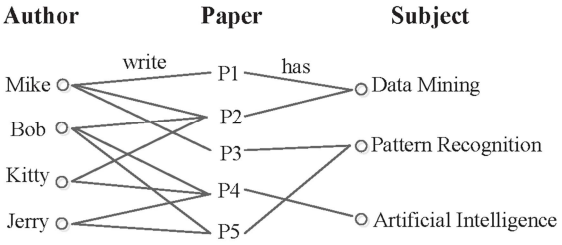


Figure 2: Heterogeneous Relation Network Example.

We take the simple network showed in Fig.2 as an example to calculate the relevance between *Mike* and the subject *DataMining* (*DM* for short) based on path *APS* ("Author-Paper-Subject").

$$AvgSim(Mike,DM|APS = \frac{1}{2}[RW(Mike,DM|APS) + RW(DM,Mike|SPA)]) \quad (3)$$

$$RW(Mike,DM|APS) = \frac{1}{|O(Mike|AP)|}\sum_{i=1}^{|O(Mike|AP)|} RW(O_i(Mike|AP),DM|PS) \quad (4)$$

We notice from Fig.2 that O(Mike|AP) = {P1; P2; P3}, thus we need to calculate relatedness between each out-neighbor of Mike and DM, like $RW(P_1,DM|PS)$.

$$RW(P_1,DM|PS) = \frac{1}{|O(P_1|PS)|}\sum_{i=1}^{|O(P_1|PS)|} RW(O_i(P_1|PS),DM) \quad (5)$$

Since $O(P_1|PS) = \{DM\}$, out-neighbors of P_1 based on relation PS will meet with DM, thus $RW(P_1,DM|PS) = 1$. Here we believe that $RW(DM,DM) = 1$ since the relatedness between an object and itself is 1. Therefore, we can define equation (6) as follows to measure relatedness value of objects which meet during random walk.

Journal of Theoretical and Applied Information Technology
10th February 2016. Vol.84. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$RW(s,t) = \mathrm{f(x)} = \begin{cases} 1, & \text{s and t are same} \\ 0, & \text{else} \end{cases} \quad (6)$$

Finally, we can easily calculate that the relatedness value between Mike and DM along path APS is 2/3. Likewise, the relatedness value of reverse random walk along path SPA is 2/3. Thus the relevance value (i.e. AvgSim) between author Mike and subject DataMining is 0.67 (2/3).

The example above shows the operation process of AvgSim measuring relevance of two arbitrary objects along a meta path. Next we will study on how to calculate AvgSim generally using matrices.

Given a simple directed meta path $A \xrightarrow{R} B$, where object A and B are linked though relation R. The relationship between A and B can be expressed by adjacent matrix, denoted as M_{AB}. Two normalized matrix R_{AB} and C_{AB} are generated by normalizing MAB according to row and column vector respectively. R_{AB} and C_{AB} are transition probability matrix which represent $A \xrightarrow{R} B$ and $B \xrightarrow{R^{-1}} A$ respectively. According to properties of matrix, we can derive relations $R_{AB} = C'_{AB}$ and $C_{AB} = R'_{AB}$ where R'_{AB} is the transpose of R_{AB}.

If we extend the simple meta path to $\mathrm{P} = A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \dots \xrightarrow{R_l} A_{l+1}$ where R is a composite relation $\mathrm{R} = R_1 \circ R_2 \circ \cdots R_l$, then the relationship between A_1 and A_{l+1} is expressed as **reachable probability matrix** which is obtained by computation on the basis of transition probability matrix. The reachable probability matrix of P is defined as $RW_P = R_{A_1A_2}R_{A_2A_3} \dots R_{A_lA_{l+1}}$, where RW_P is the random walk relatedness matrix from object A_1 to A_{l+1} along path P.

Then we can rewrite AvgSim using reachable probability matrix according to equation (1) and (2) as follows.

$$\begin{aligned} &AvgSim(A_1, A_{l+1}|P) \\ &= \frac{1}{2}[RW(A_1, A_{l+1}|P) + RW(A_{l+1}, A_1|P^{-1})] \\ &= \frac{1}{2}[RW_P + RW'_{P^{-1}}] \end{aligned} \quad (7)$$

In equation (7), two reachable probability matrices should have same dimensions since we will take arithmetic mean for them. Thus we transpose the second probability matrix as $RW'_{P^{-1}}$ to satisfy our calculation need.

$$\begin{aligned} &AvgSim(A_1, A_{l+1}|P) \\ &= \frac{1}{2}[R_{A_1A_2}R_{A_2A_3} \dots R_{A_lA_{l+1}} \\ &\qquad + (R_{A_{l+1}A_l}R_{A_lA_{l-1}} \dots R_{A_2A_1})'] \\ &= \frac{1}{2}[R_{A_1A_2}R_{A_2A_3} \dots R_{A_lA_{l+1}} + (R'_{A_2A_1}R'_{A_2A_3} \dots R'_{A_{l+1}A_l})] \\ &= \frac{1}{2}[R_{A_1A_2}R_{A_2A_3} \dots R_{A_lA_{l+1}} + C_{A_1A_2}C_{A_2A_3} \dots C_{A_lA_{l+1}}] \end{aligned} \quad (8)$$

Applying relation $C_{AB} = R'_{BA}$, equation (8) is derived above. We notice that the calculation of AvgSim is unified as two chain multiplications of transition probability matrices. The only difference between two chains is the normalization form of original adjacent matrix.

AvgSim can measure relevance of any heterogeneous or homogeneous objects based on symmetric path (e.g. APCPA) or asymmetric path (e.g. APS). Besides, the method has symmetric property, which can be verified easily from the definition equation of AvgSim and the symmetric property has a positive effect on clustering. However, the calculation of AvgSim, or rather the chain matrix multiplication is time-consuming and restricted by memory size. In order to apply our algorithm in real large-scale heterogeneous information networks, we have to consider how to improve the efficiency of AvgSim.

Algorithm 1 (AvgSim-MatrixChain) Dynamic Programming for Matrix Chain Multiplication

Input: p[n + 1]
Output: s
Set n = p. Length − 1 and Let m and s be new tables
for i ← 1 to n **do**
  m[i, j] ← 0
end for
for r ← 2 to n **do**
  for i ← 1 to n − r + 1 **do**
    j ← i + r − 1
    m[i, j] ← m[i, i] + m[i + 1, j] + $p_{i-1}p_ip_j$
    s[i, j] ← i
    for k ← i to j − 1 **do**
      q ← m[i, k] + m[k + 1, j] + $p_{i-1}p_kp_j$
      if q < m[i, j] **then**
        m[i, j] ← q, s[i, j] ← k
      end if
    end for
  end for
end for
return s

4. PARALLELIZATION OF AVGSIM

Parallelism [8] is an effective method for processing of massive data and improving algorithm's efficiency. According to the features and application scenarios of AvgSim, we will realize it using parallelization method and the specific steps are as follows.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

1. Since the core calculation of AvgSim is the chain matrix multiplication, we firstly change the order of matrix multiplication operations by applying Dynamic Programming strategy.

2. After step 1, we turn to focus on single large-scale matrix multiplication and it can be parallelized on Hadoop distributed system using MapReduce programming model.

4.1 Dynamic Programming

Different orders of operations in chain matrix multiplication leads to different computation time. There exists an optimal order of chain matrix multiplication using Dynamic Programming, which consumes the shortest computation time.

Let m[i,j] be the shortest operation time of chain matrix multiplication $A_i \dots A_j$, thus m[1,n] is the optimal solution we need. The following recursive expression is used to construct the optimal solution, where the dimension of matrix A_i is $p_{i-1} \times p_i$.

$$m[i,j] = \begin{cases} 0, & if\ i = j \\ min_{i \le k \le j}\{m[i,k] + m[k+1,j] + p_{i-1}p_k p_j\}, & if\ i < j \end{cases} \quad (9)$$

We define a one-dimensional array p[n+1] which contains dimensions of each matrix in the matrix chain where n is the quantity of matrices, a two-dimensional array m to record values of cost, and another two-dimensional array s to record the segmentation of point k corresponding to the optimal value. Algorithm for deriving s is shown in **Algorithm 1**.

Optimal solution can be constructed recursively according to S and then we can do chain matrix multiplication according to the new order of operations. The single matrix multiplication will be parallelized on MapReduce in the next section.

4.2 Parallelization On MapReduce

Parallelization of AvgSim is mainly the parallelization of matrix multiplication after Dynamic Programming process. Dealing with large-scale matrices, we use parallelized "block matrix multiplication" method on MapReduce to transform multiplication of two large matrices into several multiplications of smaller matrices. This method is flexible with selecting dimensions of block matrix according to the configuration of Hadoop cluster and avoids exceeding the memory size. The selection of block matrices determines operating efficiency to a great extent and will be experimented in Section 5.

Given a $M \times N$ matrix A and a $N \times P$ matrix B, we partition them into block matrices with dimensions $s \times t$ and $t \times v$ respectively. We can derive a new left matrix and a new right matrix after partition (A is $m \times n$ and B is $n \times p$), and the corresponding partition expression is as follow.

$$\begin{cases} m = \frac{M-1}{s} + 1 \\ n = \frac{N-1}{t} + 1 \\ p = \frac{P-1}{v} + 1 \end{cases} \quad (10)$$

The parallelization of block matrix multiplication will be implemented by two-round MapReduce computing. Assume that the two matrices for multiplication is A and B mentioned above, and M_{xy} represents the value of x-th row and y-th column of matrix M. The algorithms for the two-round MapReduce are shown in **Algorithm 2** and **Algorithm 3**.

Applying two-round MapReduce algorithm above iteratively to the chain matrix multiplication which is re-ordered by Dynamic Programming, we can get one of the two reachable probability matrices of AvgSim (e.g., RW_P, which is measured in the given meta path P). And the other probability matrix ($RW'_{P^{-1}}$) can be obtained through exactly the same procedure. Finally, the relevance matrix is derived by taking arithmetic mean of these two reachable probability matrices.

Algorithm 2 The First Round MapReduce of Matrix Multiplication

Map: Value is $< \mathrm{M, x, y}, M_{xy} >$
if M is "A" **then**
  for k_per_s ← 0 to p − 1 **do**
    emit $\left((\mathrm{i/s, j/t, k\_per\_v}), (\text{"A"}, \mathrm{i\%s, j\%t}, A_{ij})\right)$
  end for
else
  for i_per_s ← 0 to m − 1 **do**
    emit $\left((i\_per\_s, \mathrm{j/t, k/v}), (\text{"B"}, \mathrm{j\%t, k\%v}, B_{jk})\right)$
  end for
end if

Reduce: Key is $<$ i_per_s, j_per_t, k_per_v $>$ and Value is a list of $< \mathrm{M, x\_mod, y\_mod}, M_{xy} >$
for value in values **do**
  if M is "A" **then**
    add $(\mathrm{i\_mod\_s, j\_mod\_t}, A_{ij})$ to List_A
  else
    add $(\mathrm{j\_mod\_t, k\_mod\_v}, B_{jk})$ to List_B
  end if
end for
Define: hash={}
for a in List_A **do**
  for b in List_B **do**
    if a[1] == b[0] **then**
      hash[(a[0], b[1])]+= a[2] ∗ b[2]
    end if
  end for

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

end for
for {(i_mod_s, k_mod_v): v} in hash **do**
emit($(key[0] * s + i\_mod\_s, key[2] * v + k\_mod\_v), v$)
end for

Algorithm 3 The Second Round MapReduce of Matrix Multiplication

Map: emit (key, value)

Reduce: Key is < x, y > and Value is a list of < value >
Set: sum = 0
for value in values **do**
sum+= value
end for
emit (key, sum)

5. EXPERIMENTS

5.1 Datasets

ACM dataset, DBLP dataset and Matrix dataset are used in experiments and the first two network schemas are shown in Fig. 1(a) and (b). In detail, the ACM dataset is downloaded from ACM digital library in June 2010 and contains 17K authors, 1.8K author affiliations, 12K papers and 14 computer science conferences including 196 corresponding venue proceedings. We also extract 1.5K terms and 73 subjects from these papers. The DBLP dataset collected from DBLP website contains 14K papers, 14K authors, 20 conferences and 8.9K terms. And we label 20 conferences, 100 papers, and 4057 authors in the dataset with four research areas including database, data mining, information retrieval and artificial intelligence for experiments use.

Large-scale dataset is needed in parallelization experiments, where ACM dataset or DBLP dataset cannot satisfy this requirement. Thus we artificially generate several large-scale sparse square matrices, whose dimensions are 1000×1000, 5000×5000, 10000×10000, 20000×20000, 40000×40000, 80000×80000, 100000×100000 and 150000×150000, respectively. And the sparsity of each matrix includes 0.0001, 0.0003, 0.0005, 0.0007 and 0.001. Then we form a large-scale matrix dataset named Matrix dataset including the 40 matrices in total mentioned above.

5.2 Performance of AvgSim

In this section, we design several experiments to validate the effectiveness and efficiency of AvgSim.

5.2.1 Performance on query task and clustering task

We design two tasks to verify the effectiveness of AvgSim, which are query task and clustering task respectively.

In the query task, we compare the performance of AvgSim with both HeteSim and PCRW though measuring the relevance of heterogeneous objects on DBLP dataset. Based on labels of the dataset, we calculate the AUC (Area Under ROC Curve) score to evaluate the performance of the results which are the related authors ranked by relevance scores for each conference on meta path CPA. We evaluated 9 out of 20 marked conferences, whose AUC values are shown in Table 1. We notice that AvgSim gets the highest value on 8 conferences, which means AvgSim performs better than other two methods in the query task. In the clustering task, we compare the performance of AvgSim with both HeteSim and PathSim though measuring the relevance of homogeneous objects on DBLP dataset. We firstly apply three algorithms respectively to derive the relevance matrices on three meta paths including CPAPC, APCPA and PAPCPAP. Applying Normalized Cut to the result matrices, we perform clustering task and then evaluate the performances on conferences, authors, and papers using NMI criterion (Normalized Mutual Information). The clustering accuracy result is shown in Table 2 and AvgSim gets the highest NMI values in all the three tasks. The results of query task and clustering task validate the effectiveness of AvgSim.

Table 2: Clustering accuracy results for path-based relevance measures on DBLP dataset

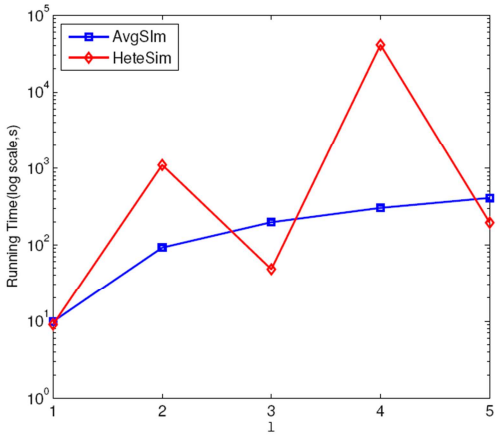
	Venue NMI	Author NMI	Paper NMI
PathSim	0.8162	0.6725	0.3833
HeteSim	0.7683	0.7288	0.4989
AvgSIm	**0.8977**	**0.7556**	**0.5101**

5.2.2 Efficiency of AvgSim compared with HeteSim

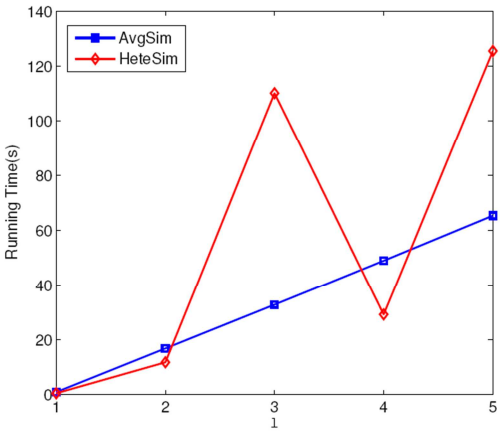
In this section, we will verify the efficiency of AvgSim on ACM dataset. We take relevance measure experiments of AvgSim and HeteSim respectively based on meta paths $(APCPA)^l$ and $(TPT)^l$, where l is the number of path repetitions with a range from 1 to 5.

Journal of Theoretical and Applied Information Technology
10th February 2016. Vol.84. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195



(a) $(APCPA)^l$



(b) $(TPT)^l$

Figure 3: Running time of AvgSim and HeteSim based on different meta paths.

Fig.3(a) and (b) show the relationship between running time and different meta paths for each method. We notice that running time of HeteSim exhibits great fluctuations with the change of path length, while AvgSim is relatively more stable. According to the definition of AvgSim, the longer paths (value of l) it measures, the more matrices should be multiplied, thus the time increases persistently.

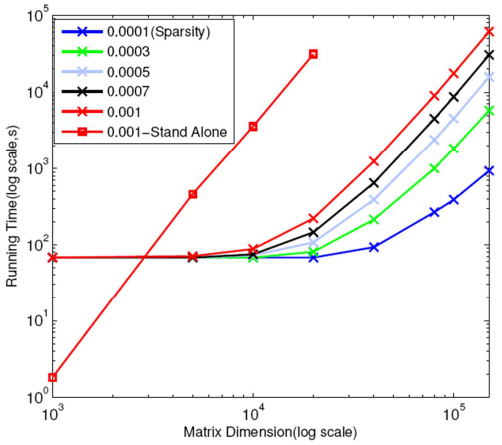
In contrast, the calculation of HeteSim needs two steps including matrix multiplication and relevance computation. In matrix multiplication step, HeteSim calculates reachable probability matrices from source and target nodes to the middle node respectively. The longer paths it measures, the more time it needs. In relevance computation step, the relevance matrix is the multiplication result of two probability matrices in previous step. We take two data points with l value of 2 and 3 in Fig.3(a) as an example. Time of matrix multiplication at l = 3 is longer than that at l = 2 because more matrices need to be multiplied. However, the total time it takes is much shorter as shown in the figure, which means the relevance computation at l = 2 costs more time. In fact, the relevance computation at l = 2 is $RW_{AA} \times RW_{AA}$ with middle node A, while at l = 3 is $RW_{AC} \times RW_{CA}$ with middle node C. Since the dimension of C is much smaller than that of A, computation at l = 2 will take longer time. The reason for data points with l value of 3 and 5 in Fig.3(b) is the same.

In conclusion, relevance computation of HeteSim affects its performance to a great extent and it will be relatively poor for large-scale matrices. Conversely, AvgSim performs much more stable, and its efficiency is only related to matrix dimension and meta path length, which can be improved by parallelized matrix multiplication on MapReduce.

5.2.3 Performance of parallelized matrix multiplication

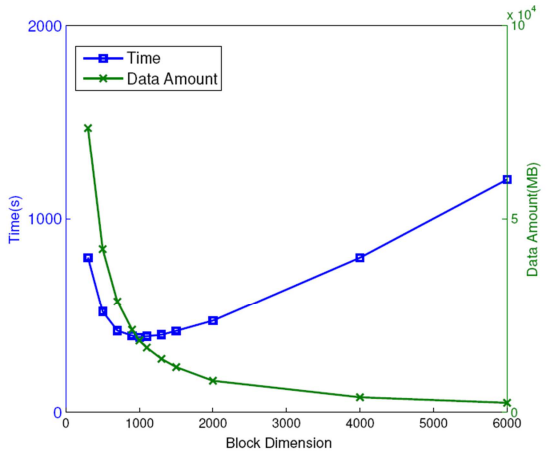
All parallelized matrix multiplication experiments are conducted in a cluster composed of 7 machines with 4-cores E3-1220 V2 CPUs of 3.10GHz and 32 GB RAM running on RedHat 4 operating system. The cluster uses Hadoop v0.20 and consists of 1 NameNode and 7 DataNodes. The experiments will measure several factors affecting block matrix multiplication, including matrix dimensions, matrix sparsity and partition strategy (i.e., dimensions of blocks). Results will reflect the performance of parallelized AvgSim algorithm.

Fig.4(a) shows the relationship among matrix dimensions, matrix sparsity and running time of parallelized block matrix multiplication together with the comparison between stand-alone and parallelized matrix multiplication. All the matrix multiplications are experimented on dataset M and apply partition strategy of 1000×1000 block matrix.



(a)Matrix Dimension And Sparsity

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195



(B) Partition Strategy
Figure 4: Factors Affecting Parallelized Block Matrix Multiplication.

From Fig.4(a), we notice that the larger the dimension or sparsity of matrix is, the more time is required in matrix multiplication. And the comparison results between stand-alone and parallelized matrix multiplication with sparsity of 0.001 show that stand-alone algorithm costs shorter time for quite small matrix dimension. This is because that parallelized algorithm spends lots of time in starting task nodes of Hadoop cluster and resources of cluster are not fully utilized for small amount of calculations. However, the efficiency of parallelized algorithm is getting better as the matrix dimension increases. Besides, stand-alone algorithm is restricted by memory size because there are no results derived in the last three large-scale matrix multiplications shown in Fig.4(a).

Fig.4(b) shows the relationship among running time, intermediate data amount and partition strategy of block matrix multiplication. There are 11 kinds of partition strategies with square block matrix dimensions of 300×300, 500×500, 700×700, 900×900, 1000×1000, 1100×1100, 1300×1300, 1500×1500, 2000×2000, 4000×4000 and 6000×6000 respectively applying in the square matrix with dimension of 100000×100000 and sparsity of 0.0001 in the experiment.

We notice from Fig.4(b) that intermediate data amount of matrix multiplication decrease gradually with the increase of block dimension. In contrast, running time reaches its minimum value at 5-th data point shown in the figure. Smaller intermediate data amount results in less disk IO operations and data amount transmitted by shuffle, which also means shorter time and better performance to a certain extent as front several data points reflected. However, excessive large block dimension will reduce the concurrent granularity and increase the amount of calculations for single node, which conversely results in longer time of computation as several data points behind reflected.

In conclusion, appropriate partition strategy and sufficient sizes of cluster affect the efficiency in parallelized block matrix multiplications greatly. Applying parallelization method, AvgSim gains the ability to measure relevance in larger-scale networks with massive data efficiently.

6. CONCLUSIONS

In this paper, we introduced a novel algorithm with symmetric features named AvgSim for measuring relevance of arbitrary objects in heterogeneous information network. In addition, using Dynamic Programming and parallelization methods, AvgSim is implemented to be able to deal with massive data and is applied to actual networks. Experiments given in the paper verified the effectiveness and efficiency of AvgSim in measuring the relevance of heterogeneous or homogeneous objects based on meta paths.

This work is supported in part by and the National Natural Science Foundation of China (No. 71231002), and the Co-construction Project of Beijing Municipal Commission of Education.

REFERENCES:

[1] Y. Sun, J. Han, P. Zhao, Z. Yin, H. Cheng, and T. Wu.: RankClus: integrating clustering with ranking for heterogeneous information network analysis. In: EDBT, pp.565{576. (2009)

[2] X. Kong, P. S. Yu, Y. Ding, and D. J. Wild.: Meta path-based collective classification in heterogeneous information networks. In: CIKM, pp.1567{1571. (2012)

[3] L. Page, S. Brin, R. Motwani, T. Winograd.: The pagerank citation ranking: bringing order to the web. Stanford University Database Group. Technical report, (1998)

[4] Glen Jeh, Jennifer Widom.: SimRank: a measure of structural-context similarity. In: KDD, pp.538{543. (2002)

[5] Y. Sun, J. Han, X. Yan, P. Yu, T. Wu.: Pathsim: meta path-based top-k similarity search in heterogeneous information networks. In: VLDB, pp.992{1003. (2011)

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[6] N. Lao, W. Cohen.: Relational retrieval using a combination of path-constrained random walks. In: Machine Learning, 81(1): pp.53{67. (2010)

[7] Chuan Shi, Xiangnan Kong, Yue Huang, Philip S. Yu, Bin Wu.: HeteSim: A General Framework for Relevance Measure in Heterogeneous Networks. In: IEEE Trans. Knowl. Data Eng. 26(10): 2479{2492 (2014)

[8] Liangliang Cao, Brian Cho, Hyun Duk Kim, Zhen Li, Min-Hsuan Tsai, Indranil Gupta.: Delta-SimRank computing on MapReduce. In: BigMine, pp.28{35. (2012)

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 1: AUC Values For Relevance Search Of Conferences And Authors Based On CPA Path On DBLP Dataset

	KDD	ICDM	SDM	SIGMOD	VLDB	ICDE	AAAI	IJCAI	SIGIR
HeteSim	0.8111	0.6752	**0.6132**	0.7662	0.8262	0.7322	0.8110	0.8754	0.9504
PCRW	0.8030	0.6731	0.6068	0.7588	0.8200	0.7263	0.8067	0.8712	0.9390
AvgSim	**0.8117**	**0.6753**	0.6072	**0.7668**	**0.8274**	**0.7286**	**0.8114**	**0.8764**	**0.9525**