The Partition of Semantic Web Data

Yonglin Leng[1,a], Fuyu Lu [2,b]

[1]College of Information Science and Technology, Bohai University, Jinzhou, P.R. China

[2]Office of Academic Affairs, Bohai University, Jinzhou, P.R.China

[a]Lyl_dllg2013@mail.dlut.edu.cn, [b]lufuyu@qq.com

Keywords: semantic web; RDF directed graph; partition; shortest path; AP clustering

Abstract. With the rapid growth of the Semantic Web data, RDF data storage has become a hot research topic in the field of data storage. Distributed storage is an effective way to solve the scalability of RDF data, and data partition is the key to realize the distributed storage. In this paper we use graph clustering idea to realize the effective partition of RDF data. Since the properties of the RDF data model, we presents a similarity measure algorithm based on shortest paths to calculate the similarity between nodes of the RDF graph, then use the AP clustering algorithm to cluster similarity matrix, and realize the partition and distributed storage of RDF data. The experiment results show that, the algorithm can effectively complete the clustering partition of the RDF data, makes the high-similarity nodes fall into one cluster while low similarity nodes are distributed to different clusters.

Introduction

RDF (Resource Description Framework) is used to express a World Wide Web information resource, there are a lot of institutions and projects, such as Google, Wikipedia, DBLP expressed their metadata by RDF [1]. With the rapid development of the semantic web, the size of RDF data is growing fast. The storage and management of RDF data by single node has become a bottleneck in the development process of RDF data. While distributed storage model can solve the scalability problem of RDF data. The traditional relation and object data model can not meet the low data redundancy and high query performance at the same time. If the management of RDF data by graph model can not only avoid the conversion between RDF logical data and physical data model, but also can use the mature graph algorithm to optimize the RDF data query [2]. Graph clustering is the premise of distributed storage, according to certain standard the vertices of the graph are divided into different clusters, making the intra cluster similarity as large as possible, and the similarity between clusters is smaller.

A key problem of graph clustering is how to measure the similarity between nodes. There are many similarity metric algorithms, such as similarity metrics based on structural, similarity metrics based on attribute and structure/attribute similarity metric. The structural similarity metric does not require analyse the entity information and only consider the topology structure. The structure similarity metric is a common method for the similarity metric. SimRank [3], P-Rank (Penetrating-Rank) [4] are the universal structure of similarity model which is inspired by PageRank of Google ranking algorithm. But the SimRank and P-Rank algorithms as the iterative computation makes the algorithm's time complexity and space complexity is very high. Especially with the increasing amount of data, the efficiency of the algorithm has been unable to meet the requirements of real-time calculation. In this paper we presents a similarity measure method based on shortest paths for graph nodes similarity measure. Then the similarity matrix is clustered with AP clustering algorithm. The experimental results show that, the method proposed in this paper can realize the clustering partition of graph.

Related Theories and Technologies

RDF data use triple <subject, predicate, object> to express data, where predicate indicate the attribute of subject and object is the value of subject. RDF data can also be described by a directed graph, where the subject is the entity from which the (directed) edge emanated, the predicate is the label of the edge, and the object is the name of the entity or literal on the other side of the edge.

Given a RDF graph $G=(V,E)$, where V is the set of vertices and E is the set of edges, RDF graph clustering is to divide G into k disjoint segments $G_i=(V_i,E_i)$, where $V=\bigcup_{i=1}^{k}(V_i)$ and $V_i \bigcap V_j = \phi$ for any $i \neq j$.

Affinity Propagation (AP) clustering algorithm [5] was proposed by J.Frey and published in Science magazine. The algorithm cluster N data points according to the similarity between data points, which composed of $N \times N$ similarity matrix S.

AP algorithm does not need specify the cluster number in advance, all the data points will be the potential clustering center which is called exemplar. The value of $s(k,k)$ is the standard whether k point is the cluster center. The data points with larger values of $s(k,k)$ are more likely to be chosen as exemplars. These values are referred to as "preference". The number of clusters is affected by the values of the input preferences, this value can be varied to produce different numbers of clusters. AP algorithm transmited two types of message (responsibility) and (availability). $r(i,k)$ sent from data point i to candidate exemplar point k, reflects the accumulated evidence for how well-suited point k is to serve as the exemplar for point i, taking into account other potential exemplars for point i. $a(i,k)$ sent from candidate exemplar point k to point i, reflects the accumulated evidence for how appropriate it would be for point i to choose point k as its exemplar, taking into account the support from other points that point k should be an exemplar. $r(i,k)$ and $a(i,k)$ is stronger, the possibility of k point as the clustering center is bigger, and the possibility of i point belong to clustering which k as the cluster center is bigger. AP algorithm use iteration to update responsibility and availability of data points constantly, until produce m high quality exemplar. The responsibilities are computed using the formula(1), where s is the similarity matrix.

$$r(i,k)=s(i,k)-\max_{k' \neq k}\{a(i,k')+s(i,k')\} \tag{1}$$

The availability are computed according to the formula(2)

$$a(i,k)=\min\{0,r(k,k)+\sum_{i' \notin \{i,k\}}\max\{0,r(i',k)\}$$
$$a(k,k)=\sum_{i' \neq k}\max\{0,r(i',k)\} \tag{2}$$

Given a RDF graph $G=(V,E)$, $p(\tau)$ represent the transition probability matrix between any two vertexs of length τ of a graph G, $w(\tau)$ is the weight between any two vertexs of length τ [6].

$$w(\tau)=1/\tau \tag{3}$$

The longer path between two points, the lower similarity between two points.

The transition probability from v_i to v_j is defined as:

$$p^{\tau}_{(v_i,v_j)}=\frac{k_{\tau}(v_i,v_j)}{\sum_{\forall x \in V-\{v_i\}} k_{\tau}(v_i,x)} \tag{4}$$

$p^{\tau}_{(v_i,v_j)}$ is probability of going from v_i to v_j with length τ and is equal to number of τ-path from v_i to v_j divided by the number of τ-path starting from v_i.

The similarity between any two points is computed as follows:

$$\mathrm{d}(\mathrm{v}_{\mathrm{i}}, \mathrm{v}_{\mathrm{j}}) = \sum_{\tau=1}^{l} p(\tau)w(\tau) \tag{5}$$

Similarity Algorithm Based on Shortest Paths

Algorithm 1 describes the computation process in a given graph G.

Algorithm1:Shortest_similarity(G,len)

```
Input:a RDF graph G(V,E),the shortest path length len
Output:similarity matrix
Q=initialqueue()
    int P[][][] = new int[len-1][|V|][|V|]
    For each e($v_i$,$v_j$) in E
        P[0][$v_i$][$v_j$] = 1
        Q.insert(($v_i$,$v_j$))
    endfor
    Do
        tmppath = Q.remove()
        firstnode = tmppath.getfirstnode()
        for each e(lastnode,$v_j$) in E
            P(tmppath.length, firstnode, $v_j$) = P(tmppath.length, firstnode, $v_j$)+ 1
        endfor
    until tmppath.length<=len
    For each $v_i$ in G
        Do while i<=len
            For each $v_j$ in G
                scorepath($v_i$,$v_j$)=p(i,$v_i$,$v_j$)/allpath(i,$v_i$,{v}-{$v_j$})
                Similarity($v_i$,$v_j$)= Similarity($v_i$,$v_j$)+1/i*scorepath($v_i$,$v_j$)
            End for
        loop
    Endfor
```

Experiments

In this paper, we choose DBLP Computer Science Bibliogrpahy as the test data set, the data set consists of 2555 articles and 6101 citation relations, involving ten computer science domains. So we constructed 10 RDF subgraph and a RDF summary graph. The experimental environment: Inter I3 processor, 4GB memory, Windows XP operating system, C++ programming language.

To validate the algorithm proposed in this paper, we compared this algorithm with P-Rank and SimRank algorithm. The weight coefficients of P-Rank and SimRank is 0.5, the damping factor is 0.8, the length of shortest path is 5.

Experiment compared the value of the node pairs in three similarity matrices which generated by the three algorithms. If the similarity of node pairs equalled to 0, we say that there is no similarity between node pair, otherwise the node pair exist similarity.

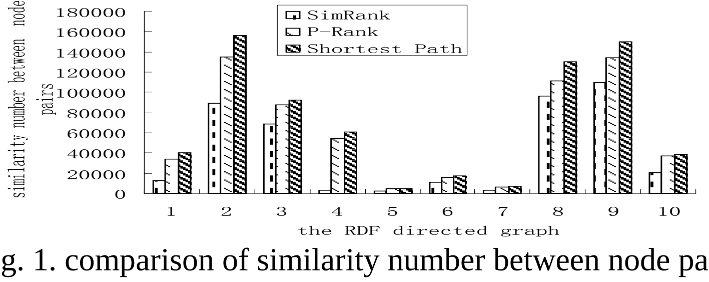

Fig. 1. comparison of similarity number between node pairs

Fig.1 described the comparison of similarity number between node pairs, P-Rank algorithm consider both the in degree and out degree, so the similarity number was higher than SimRank. We proposed the algorithm calculated all the length τ shortest path between nodes, so produce more similarity node pairs.

We define the structure distance between two nodes as formula (6).

$$d(v_i,v_j)=1-s_f(v_i,v_j) \qquad (6)$$

Where $s_f(v_i,v_j)$ is the similarity of between v_i and v_j. Clustering compression ratio is described as follow:

$$C_f=\frac{\sum_{i=1}^{k}\sum_{x\in C_i} d(x,m_i)}{\sum_{1\le i\le j\le k} d(m_i,m_j)} \qquad (7)$$

Where k is the number of clustering, C_i is the $i-th$ clustering, m_i,m_j represent the cluster center of C_i and C_j separately, numerator in formula (7) is the intra distance in clustering and denominator is the inter distance between clustering.

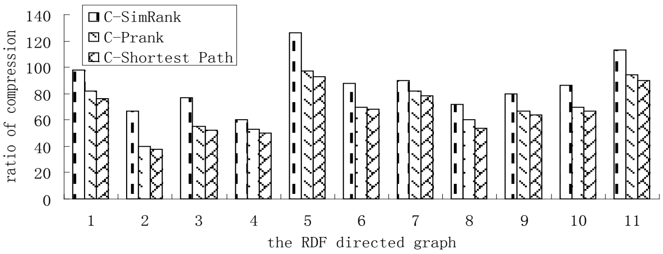

Fig. 2. Comparison of clustering compression ratio

Fig.2 indicate the shortest path algorithm with the highest compression ratio, mainly because the shortest path algorithm allows more node pairs connecting each other.

Conclusion

In this paper, we proposed a similarity measure method based on shortest path and used AP clustering algorithm to partition the RDF directed graph. We tested the similarity number between node pairs and compression ratio with SimRank, P-Rank, the Shortest path three different algorithm. Experiments show that the shortest path algorithm can measure the similarity between node pairs effectively and realize the directed graph partition.

References

[1] F. Du, Y. G. Chen, X. Y. Du, "Survey of RDF Query Processing Techniques," Journal of Software, vol. 24, no. 6, pp. 1222-1241, 2013.

[2] G. WU, "Research on Key Technologies of RDF Graph Data Management," Tsinghua University press, 2008.

[3] G.Jeh, J.Widom, "SimRank: a measure of structural-context similarity," In Proceedings of the eighth ACM SIGKDD conference(KDD'02), pp. 538-543, 2002.

[4] P. Zhao, J. Han, Y. Sun, "P-rank: A comprehensive structural similarity measure over information networks, " International Conference on Information and Knowledge Management, pp. 553-562, 2009.

[5] B. Frey, D. Duck, "Clustering by passing messages between data points," Science, vol. 315, no. 5814, pp. 972-976, 2007.

[6] H. Khosravi-Farsani, M. Nematbaksh, G. Lausen, "SRank: Shortest paths as distance between nodes of a graph with application to RDF clustering," Jouranl of Information Science, vol. 39, no. 2, pp. 198-210, 2013.