

Content-based Clustered P2P Search Model Depending on Set Distance

This paper presents P2P search model based on set distance. It makes use of the sparse feature vectors of set to define documents, and weights the document similarity by the difference between these vectors. This can effectively reduce the complexity of computing the document similarity. At the same time, clustering methods based on superpeers can reduce the redundant messages brought by flooding.

VSM is a kind of model that has been widely used these years to define the document characteristic. In this model, the document space is looked as a vector space made up of a group of quadrature term vectors and every document is defined as one of the term vectors. The similarity two different documents can be indicated by the cosine between them. But when there are too many characteristics or keywords extracted from the whole document space, while the number of characteristics to define every document is too small, there will be many characteristics valued 0 in the vectors of these documents. Then, the document space becomes high dimension sparse matrix, which brings out much redundant information and makes cosine calculation more complex. This paper proposes define a document similarity calculation based on set distance which can compress data effectively and decreases the calculations. First, the definition of set distance is given here.

Definition 1: Given a corpus which contains n objects and m keywords, each object is described by m attributes and the keywords are valued 0 or 1 (sparse value). X is a subset of this corpus, in which the number of object is denoted as $|X|$. The number of attributes that equal 1 for all objects is indicated by a , and the number of attributes that equal 1 for some objects and equal 0 for other objects is indicated by b . Distance of set X , $SD(X)$, is denoted as,

$$SD(X) = \frac{b}{|X| * a} \quad (1)$$

The sparse distance of a document set can indicates the distance degree among the documents in this set. By using the distance of this set, definition 2 in this paper presents a new measurement to calculate the document similarity.

Definition 2: There is a set D . Select two documents named d_1 and d_2 from D to compose subset D' . In these two documents, the number of keywords valued 1 in both is Num1, and the number of keywords that are not equal is Num2. By the set distance of D' , we can get $Sim(d_1, d_2)$, the document similarity between d_1 and d_2 is,

$$Sim(d_1, d_2) = \frac{1}{SD(D')} = \frac{2 * Num1}{Num2} \quad (2)$$

For example, the keywords set to define a corpus D is (A_1, A_2, \dots, A_6) . Assume d_1 is described $(1,0,1,1,1,1)$, d_2 is described $(1,0,0,0,0,1)$, and d_3 is $(1,0,1,1,0,0)$. Then, according to equation (2), we get $Sim(d_1, d_2)$ is 1.3333, and $Sim(d_1, d_3)$ is 3. That means d_1 and d_3 are more similar. Thus, the correctness of definition 2 is proved.

VSM compares the similarity among the documents by the cosine of two documents vectors.

But to the document vector in high dimension semantic vector space, the calculation of cosine has high complexity. This paper calculates the documents similarity depending on set distance. This can restrain the computing complexity in linear time. Observing definition 1, we can found that making use of set distance can not only describe the documents similarity, but also can implement content-based cluster. Besides, the effect of cluster can also be got from set distance. The definition to cluster feature vector is followed.

First, this paper views every peer in the system as a cluster. Thus, every peer has a cluster feature vector. The peers are clustered according to their feature vector distance and connected by superpeers. The peers ready to be clustered are in underlayer and in upperlayer are the superpeers representative k different clusters.

Each peer in the system maintains a inverted table of local documents to record the appearance of keywords in itself. At the same time, every superpeer maintains a vocabulary to record the keywords that have been extracted from its cluster and their distribution among peers. As shown in fig.3, when a peer p initiate a query q with the vecor is (t_1, t_2, \dots, t_m) , the searching algorithm is followed.

Step1: First, p executes q in its own inverted table. If DocID of some related documents can be got, calculate the similarity between query vector q and these documents (given in definition 2), to return some documents with the similarity greater than $Similarity_{document}$. If the number of documents arrives at the given number, then finish the query. Otherwise, q is forwarded to superpeer SN;

Step2: When SN received query q , it first searches locally. Then, it will search related documents in its inverted table according to cluster feature vector to return K peers, p_1, p_2, \dots, p_k , that are responsible for these documents. Query q is forwarded to these peers.

Step3: p_1, p_2, \dots, p_k , execute the query in their local documents respectively and return the result to SN. SN sorts the documents according to the similarity.

Step4: If the number of documents that have been returned doesn't arrive at the given number, SN will flood the query to other superpeers, to execute step 2 and step 3 iteratively until finished.

Content-based clustering method introduced in section 3.1 has clustered the peers with high similarity. Thus, the algorithm here has selected peers containing related documents with high probability to avoid redundant messages brought by flooding and improved the query efficiency.

In dynamic P2P system, the free left and join of peers will induce the augment and deletion of documents, which will accordingly make the vocabulary change. When a document with new words or teams joins, the peer will send the message to superpeer. Superpeer will subsequently update its vocabulary and flood the messages to other superpeers to maintain the consistency of keyword information in the whole system.