ReCoM: Reinforcement Clustering of Multi-Type Interrelated Data Objects

Jidong Wang, Huajun Zeng, Zheng Chen, Hong-Jun Lu, Li Tao, Wei-Ying Ma, Hong-Jiang Zhang

2003.4.3

Technical Report
MSR-TR-2003-25

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

ReCoM: Reinforcement Clustering of Multi-Type Interrelated Data Objects

Jidong Wang, Huajun Zeng, Zheng Chen, Hong-Jun Lu, Li Tao, Wei-Ying Ma,
Hong-Jiang Zhang
Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA, USA 98052

Abstract

Most existing clustering algorithms cluster highly related data objects such as Web pages and Web users separately. The interrelation among different types of data objects is either not considered, or represented by a static feature space and treated in the same ways as other attributes of the objects. In this paper, we propose a novel clustering approach for clustering multi-type interrelated data objects, ReCoM (Reinforcement Clustering of Multi-type Interrelated data objects). Under this approach, relationships among data objects are used to improve the cluster quality of interrelated data objects through an iterative reinforcement clustering process. At the same time, the link structure derived from relationships of the interrelated data objects is used to differentiate the importance of objects and the learned importance is also used in the clustering process to further improve the clustering results. Experimental results show that the proposed approach not only effectively overcomes the problem of data sparseness caused by the high dimensional relationship space but also significantly improves the clustering accuracy.

1. INTRODUCTION

Clustering analysis is a process that partitions a set of objects into groups, or *clusters* in such a way that objects from the same cluster are similar and objects from different clusters are dissimilar. Traditional clustering approaches assume that data objects to be clustered are independent and identical, and are often modeled by a fixed-length vector of feature/attribute values. The similarities among objects are assessed based on the attribute values of involved objects. In the recent surge of data mining research, this classical problem was re-examined in the context of large databases. However, homogeneity of data objects to be clustered seems still the basic assumption.

Conference '00, Month 1-2, 2000, City, State.

Recently, some emerging applications, such as Web mining and collaborative filtering propose challenges to such an assumption. In such applications, data objects are of different types and are highly interrelated. In Figure 1 we depict three data objects in the Web environment: *user, web page*, and *query*. These three types of objects are interrelated: user *issues* queries; user *browses* web pages; and query *references* web pages. It is obvious that, when we cluster Web users, the pages they browse and the queries they issue should play an important role. Similarly, when clustering Web pages, it should also be taken into consideration how they are used by users and referenced by queries. Some recent results indeed indicate that relational links between objects provide a unique source of information that has been proved useful for both classification and clustering [9][14][11].

Most existing clustering approaches cluster the multi-types of objects individually even they are highly interrelated. Relationships are either used to by transforming attributes from one type of objects to their related objects, or directly represent the relationships as additional attributes of data objects. For example, when grouping the interrelated web-pages and web users, the features of a user can be represented by its own attributes and the keyword profile derived from the content of the visited pages. Another kind of representation method, such as the one used by collaborative filtering [18], is to form the feature vector by the presence/absence of the related objects. In this case, the relationship is considered as an additional feature. In both cases,

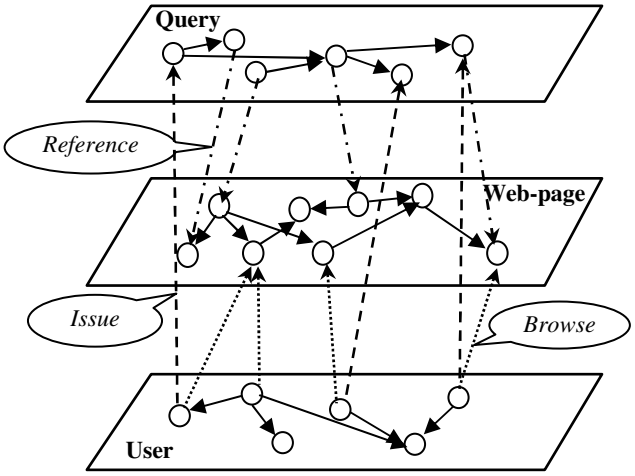


Figure 1: An example of interrelated multi-type data

the relationships or interactions between objects are only considered in feature representation step. That is, each data object is represented by two sets of features: one is extracted from data object itself, such as the attributes of the data; another is extracted from the relationship between objects. Those features are static in the sense that they remain unchanged during the clustering process.

The work reported in this paper is motivated by the following observations obtained from analyzing the existing approaches. First, relationships between interrelated data objects are often sparse in many cases. Clustering algorithms have to deal with high dimension feature vector. Second, for certain applications, data objects of the same type may play different role in the clustering process. For example, it is a well-known fact that some Web pages are more authoritative than others. Obviously the uniform treatment of all Web pages when clustering them may not be appropriate. Finally, some relationship among data objects may only be discovered during the clustering process that cannot be fully reflected in the features assigned to data objects.

To address these issues, we propose a novel framework ReCoM for clustering multi-type interrelated data objects, which fully explores the relationships between these objects for clustering analysis. Under this framework, we first identify both intra- and inter- type relationships among data objects, and represent such relationships as features of data objects. Thus, the similarity between objects includes not only the similarity of their own attributes but also the similarity of their relationships. Second, the inter-type relationships are used to reinforce the clustering process of the related objects. That is, the clustering results of one type of objects form a new feature space, which is projected and propagated to other types of objects. Then we perform clustering on those related types of objects with this updated feature space. This iterative reinforcement process is performed on each object types until converged clustering of all data types is achieved. Third, data objects are assigned different weights by analyzing the link structure derived from the inter- and intra-type relationships. A modified direct-k-means clustering algorithm is used to determine the cluster centroids by using the weighted sum of the objects in the cluster.

Our contribution can be summarized as follows.

1. We extended traditional clustering analysis to multi-type interrelated data objects. It groups interrelated data objects of different types simultaneously into clusters within each type. Objects within a cluster have high similarity with respect to both their own properties and their relationships with other objects. On the other hand, objects in different clusters are dissimilar not only in their own properties but also in their behavior when interact with other objects. Clustering of single type homogeneous data objects can be viewed as a special case for the proposed framework.
2. We proposed a reinforcement approach to cluster multi-type interrelated data objects. Under our approach, the feature spaces that represent the relationship among different types of objects are updated during the clustering process. Such changes are iteratively propagated among interrelated objects to improve the quality of clusters of all related data objects.
3. We also developed a method to differentiate importance of objects based on both the inter-and intra-type relationships among data objects and incorporate such importance into the clustering process.
4. We performed comprehensive experiments on both semi-synthetic data and real Web log data to evaluate the proposed approach. Experiments on semi-synthetic data show that the reinforcement clustering can achieve 31% and 53% entropy reduction relatively on the obtained web-page clusters and user clusters compared to traditional clustering algorithm. And the introduction of object importance can lead to better clustering results with reduced entropy under noise data. The experiment on real Web log data also shows that our algorithm outperforms the traditional algorithm by about 48% improvement in log likelihood of hits prediction.

The rest of this paper is organized as follows. Section 2 describes the problem of clustering multi-type interrelated data objects and presents our solutions. Experiments and their results are presented in Section 3. In Section 4, we present some related works. Section 5 concludes the paper.

2. CLUSTERING MULTI-TYPE INTERRELATED DATA OBJECTS

In this section, we first define the problem of clustering multi-type interrelated data objects, followed by our reinforcement clustering algorithm.

2.1 The Problem

We are given n different types of objects $X_1, X_2, \ldots, X_n$. Each type of data objects X_i is described by a set of features F_i. Data objects within the same type are interrelated with intra-type relationships $R_i \subseteq X_i \times X_i$. Data objects from two different types are related through inter-type relationships, $R_{ij} \subseteq X_i \times X_j \ (i \neq j, i = 1,2,\cdots n)$. To distinguish from the relationships, F_i is referred as *content feature* of data objects. For a specific object $x \in X_i$, we use $x.F_i$ to represent its content features, and use $x.R_i \subseteq X_i$ and $x.R_{ij} \subseteq X_j$ to denotes objects related to it in X_i and X_j, respectively.

The problem of clustering multi-type interrelated data objects is to partition each type of objects X_i into K_i clusters so that the data objects in each cluster have high similarity, and objects from different clusters are dissimilar.

Considering that an object has both content features and relationships with other objects, we define the similarity between two objects as

$$S = \alpha \bullet s_f + \beta \bullet s_{intra} + \gamma \bullet s_{inter} \qquad (1)$$

where s_f is content similarity, s_{intra} and s_{inter} are intra-type and inter-type similarities, respectively, α, β, and γ are weights for different similarities with $\alpha + \beta + \gamma = 1$.

From the above definition, we can see that the similarity between two objects is a linear combination of content similarity and relationship similarities. By assigning different values to α, β, and γ we can adjust the weights of different similarities in the overall similarity. For example, if α=1, $\beta = \gamma = 0$, we only consider the similarity between the content features. By setting $\beta = 0$, we ignore

ignore the effects of intra-type similarity.

Similarity in Equation 1 can be defined using different functions, usually determined by the types of objects and the applications. For example, content similarity between two web-pages could be defined as cosine function of the two keyword vectors derived from their contents.

In this paper, we represent relationship feature of a particular object by a vector whose entries correspond to its related objects. Each entry could be a numeric value corresponding to the weight of the relationship. For example, given two object types $X = \{x_1, x_2, \cdots x_m\}$, and $Y = \{y_1, y_2, \cdots y_n\}$, the inter-type relationship vector of object $x \in X$ is defined as $V_x = [v_1, v_2, \cdots, v_n]^T$ where $v_i \neq 0$ if $y_i \in x.R_Y$, and $v_i = 0$ otherwise. Then the similarity $S_{inter-XY}$ on inter-type relationship R_{XY} between the two objects in X could be also defined as the cosine function of the two vectors.

If objects in X_i have inter-type relationships with multiple types, the final inter-type relationship similarity could be the linear combination of all inter-type similarities.

2.2 Reinforcement Clustering

With the similarity functions defined above, the problem of clustering multi-type interrelated data objects can be solved using traditional clustering algorithms. That is, after mapping the relationships among data objects as relationship features, each type of data objects can be clustered individually. While this approach seems feasible, it will suffer when the number of objects becomes large, as the size of the feature vector for relationship will be very large. More importantly, objects within a type are treated uniformly in the feature vector. Intuitively, objects in a type have relationship with similar objects in another type should have high similarity.

To address the issues, we propose a mutual reinforcement clustering method to fully explore the relationships of different types of objects in the clustering process. The basic idea is to propagate the clustering results of one type to all its related types by updating their relationship features. That is, the relationship to individual objects is aggregated based on clusters. As such, the dimension of relationship feature is reduced relatively. Then, we can perform clustering on the updated feature. This process is iteratively performed until clustering results in all object types converge.

In the following, we use a special case of two object types $X = \{x_1, x_2, \cdots x_m\}$, and $Y = \{y_1, y_2, \cdots y_n\}$ to illustrate the process.

We first cluster the objects in *Y* into *k* clusters, denoted by $\{C_1, C_2, \ldots, C_k\}$ using any traditional clustering method. Recall that the inter-type relationship feature vector of $x \in X$ is originally defined as $V_x = [v_1, v_2, \cdots, v_n]^T$ with each component corresponding to one objects in *Y*. With the clusters in *Y*, we can replace the V_x by $V_x' = [v_1', v_2', \cdots, v_k']^T$ with each component corresponding to one cluster of Y and v_i' is non-zero if $x.R_Y \cap C_i \neq \Phi$. The numeric value of v_i' could be set to $|x.R_Y \cap C_i|$, which represent the number of relationships from object x to objects in cluster C_i , or other values such as the importance of the associated objects (will be described next subsection). Then the clustering of object in X is based on the new inter-type relationship feature. The process will continue by iteratively project the clustering results of one type to another by their inter-layer relationship until converge.

The advantage of the above algorithm is that it can solve the data sparseness problem to some extend. And the clustering results not only reflect the data distribution from the content, but also reflect the relationships with other data types. Furthermore, any traditional clustering algorithm could be easily embedded into our proposed framework to enhance the clustering performance.

2.3 Link Analysis and Importance of Objects

For some data objects and applications, objects in the same type may have different importance in the clustering process. Typical examples include Web-page/user clustering where certain Web pages are more important as they are authoritative pages, and item/user clustering for collaborative filtering, etc. where some users should be more authoritative in determining the belongingness of items. If we view objects as nodes, relationship between objects can then be viewed as links. That is, if two objects are related, a link is inserted between these two objects. With this notation, a simple approach is to apply the traditional link analysis method, such as HITS algorithm [11], to calculate the eigen-values of each data object. However, when multiple types of data objects are involved, this method will not work since the importance of different types of objects is not comparable.

To address this problem, we extend the HITS algorithm as follows. We not only consider the mutual reinforcement of object importance within a type but also the mutual reinforcement between types. Borrowing terminologies used in the HITS algorithm, each node is assigned a *hub* score and an *authority* score.

For simplicity, we continue to use the case which contains two types of interrelated objects as example to illustrate our proposed algorithm. Given two types of objects $X = \{x_1, x_2, \cdots x_m\}$, $Y = \{y_1, y_2, \cdots y_n\}$ and relationships of R_X, R_Y, R_{XY} and R_{YX} if directionality is considered. The adjacent matrixes are used to represent of the information of links. L_X and L_Y stands for the adjacent matrixes of link structures within set X and Y , respectively. L_{XY} and L_{YX} stand for the adjacent matrixes of links from X objects to Y objects respectively. For example, $L_{XY}(i, j) = 1$ if there is one link from node x_i to node y_j .

In our proposed algorithm, there are two levels of calculations: one is that the *hub* value and *authority* value of objects from same type reinforce each other by the intra-type relationships; and the other is that the *importance* of different types of nodes reinforces each other by inter-type relationships. The calculations in this approach are written as follows.

$$\begin{cases} a(X) = \beta L_X{}^T h(X) + (1-\beta) L_{XY} i(Y) \\ h(X) = \beta L_X a(X) + (1-\beta) L_{XY} i(Y) \\ i(X) = a(X) + h(X) \\ \\ a(Y) = \gamma L_Y{}^T h(Y) + (1-\gamma) L_{YX} i(X) \\ h(Y) = \gamma L_Y a(Y) + (1-\gamma) L_{YX} i(X) \\ i(Y) = a(Y) + h(Y) \end{cases} \quad (2)$$

where, $a(X)$ and $h(X)$ are the *authority* score and *hub* score of nodes within X , respectively. Similarly, $a(Y)$ and $h(Y)$ stands for the *authority* and *hub* score of nodes in Y, respectively; $i(X)$ and $i(Y)$ stands for the *importance* of the node in X and Y, respectively. β and γ are the weight parameters to adjust the influence of links derived from different relationships.

At the beginning of the calculation, all vectors, $a(X)$, $h(X)$, $a(Y)$ and $h(Y)$ are initialized to 1. The *hub* score *and* authority score are updated using Eq. (2) at each iteration. At the end of each iteration, the vectors will be normalized for the next iteration calculation.

The advantage of the above algorithm is that it provides a normalized and uniform importance within each object types and gets more reasonable result by considering the importance of the associated objects of other types through inter-type relationships.

Given the importance score of objects, the clustering process is modified to reflect the importance of objects. In our study, we modified the *k*-means clustering algorithm to weighted-*k*-means algorithm. That is, when calculating the cluster centroids, we use the weighted sum of cluster members as the new centroid. Thus, the cluster is biased to those important objects.

3. A Performance Study

In order to study the effectiveness of our proposed approach for clustering multi-type interrelated data objects, experiments are conducted on two kinds of dataset. The first is a semi-synthetic data which simulated a proxy log which recorded the users' browsing behaviors on web-pages. The second dataset is a real website traffic log data, which collected from the CS Division of UC Berkeley. Although the experiments focus on web-page/user clustering application, they could be generalized to other applications which have similar data models.

3.1 Semi-Synthetic Data

The semi-synthetic data is a simulated Web traffic log consisting of two types of objects, namely, the web-pages and the Web users. In the synthetic data, the web-pages are collected from real sources, while users and their browsing behaviors are automatically generated from a probabilistic model whose parameters are specified in advance. The real category labels of web-pages and users are concealed from the proposed algorithm and are taken as a ground-truth for the performance evaluation. In the following, after brief introduction on the data generation process, a number of comparative experiments are performed.

3.1.1 Data Generation

The data generation process is composed of three steps. Firstly, all the web-pages used in the experiment are collected from Open Directory Project [21]. About 300 categories and the associated 50,000 websites are selected from the second-level categories of this directory. Each category consists tens to hundreds of websites. We download the homepages of all the websites, and extract pure text of them

Secondly, we randomly generate 20,000 users which fall into 100 classes, each class having tens to hundreds of users. Those users have no content features defined. Thus the clustering of users fully relies on the link features. To simulate the different levels of Web experience, each user is assigned a importance value ranged from 1 to 10 randomly according to a Gaussian distribution. The value represents the importance of the user and has effect on the amount of the user's page hits. Furthermore, we assume each class of users has about 8~10 interests [4], each interest having a probability. The interests are mapped to the categories of web-pages.

Finally page hits are generated based on existing users, existing URLs, users' interests, and importance values of users and URLs. A page hit is generated by following process: (1) select a user according to probability distribution determined by user importance values; (2) get the class of this user and the corresponding interests; (3) randomly select an interest according to the probability; (4) randomly select a URL from corresponding Web category, according to URL importance values. At last, 200,000 page hits are generated in total (about 10 page hits per user) which forms a very sparse link structure.

A noise generation module is also applied to generate totally random page hits uniformly distributed on all possible pairs of users and URLs. In the experiment, we illustrate how clustering performance change along with the ratio of noise.

3.1.2 Evaluation Measure

We define a quantitative measure to evaluate the clustering accuracy. This measure is based on the entropy in information theory [7], which measures the uniformity or purity of a cluster. Specifically, given a cluster A and category labels of data objects inside it, the entropy of cluster A is defined by $H(A) = -\sum_j p_j \cdot \log_2 p_j$, where p_j should be the proportion of current class's data in the cluster. The traditional entropy measure is extended to a weighted one to consider objects' importance values. This is achieved by replacing the p_j in the above formula by a weighted proportion p_j':

$$p_j = \frac{\sum_{\forall x, label(x)=c_j} importance(x)}{\sum_{\forall x, x \in A} importance(x)}$$

where $c(x)$ denote the class label of each object $x \in A$, and $importance(x)$ denote its importance value. The total entropy is defined by:

$$H = \sum_{\text{All Clusters } A_k} \frac{\sum_{\forall x, x \in A_k} importance(x)}{\sum_{\forall x} importance(x)} H(A_k)$$

where A_k denotes the kth cluster.

In the followed series of experiments, when we do not consider the object's importance value, the importances are all set to 1, which back off to tradition entropy definition.

In the following experiments, we calculate the total entropy on fixed numbers of clusters for web-pages and users. A small value of entropy indicates a better clustering. When the data is perfectly clustered, which means objects with same class labels are clustered into one single group, the entropy is 0. For the purpose of comparison, we also calculated the maximal entropy (i.e. the expected entropy of clusters where objects of each class are uniformly assigned to all clusters) for web-pages and users: *max(H(web-pages))* = 8.7 and *max(H(users))* = 7.5.

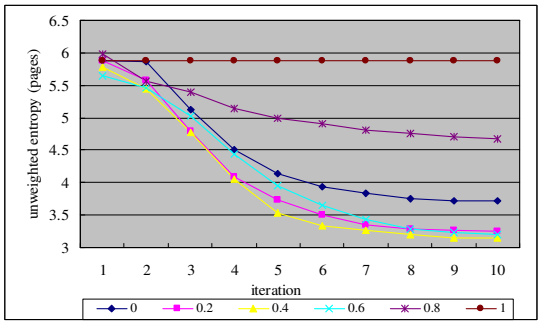
3.1.3 Impact of Relationship in Clustering

The following experiments illustrate the clustering performance of our algorithm on the generated semi-synthetic data. Given the data, the available features include content feature of web-pages and inter-type relationships between users and web-pages. Then according to Eq(1), the similarity function for web-pages is:

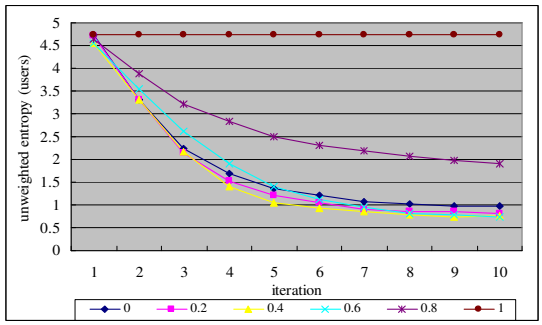
$$S = \alpha \bullet s_f + (1-\alpha) \bullet s_{inter}, \text{ where } 0 \le \alpha \le 1 \qquad (3)$$

Here α represents the weight of relationship and content feature, which has significant impact on web-page clustering accuracy. For the clustering of users, the similarity function consists only relationship feature. However it relies on the clustering result of web-pages, so its clustering accuracy also rely on the selection of α.

Since our proposed algorithm is an iterative process, in our experiment we let the algorithm to run 10 iterations (i.e. 10 iterations clustering for web-pages and 10 iterations for users). And we found that in most cases the clustering results converge after about 5~6 iterations. The clustering results for web-pages and users are represented as a curve of entropies along with iteration numbers, as shown in Figure 2(a) and 2(b) separately. Smaller entropy denotes a clustering which has higher accuracy.



(a)



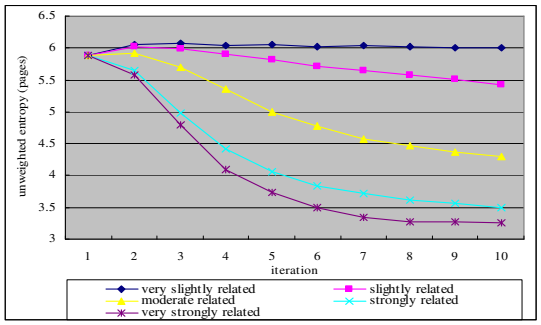
(b)

Figure 2. Impact of relationship feature and iterative reinforcement in clustering

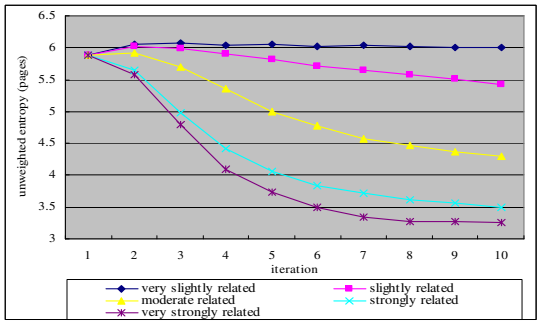
Figure 2 shows the entropy curves of our iterative clustering algorithm under different α of Eq(3). We first consider un-weighted version of this algorithm for analyzing the effect of iterative reinforcement. The web-page entropy curves are shown in plot (a), and the user entropy curves are shown in plot (b). In both plots, the best clustering accuracy is achieved when α is 0.4. When α is set to 1, which means only content feature is considered, the clustering couldn't benefit from the iteration process. Hence the entropy keeps constantly at a high value, because content feature is highly noisy and problematic. When α is set to 0, which means only relationship feature is used, the accuracy is also not satisfying because of the sparseness of link features. When α ranges from 0.2~0.6 clustering algorithm may achieve much better results. This is because relationship feature and content feature are complementary, where relationship feature provide more reliable while sparse description for data objects and content feature expand this description to similar objects.

In the two plots, the first iteration is in fact equivalent to the traditional clustering based on "flat" representations since there is no reinforcement between clustering of web-pages and users. The final entropy after iterations is the result of our algorithm, which drops significantly. Considering the curve where α=0.4, we achieve about 31% and 53% entropy reduction on the obtained clusters of web-pages and users relative to their maximal entropies. Furthermore, the experimental results also proved that our proposed iterative algorithm can converge to some static points after several iterations.

The density of relationships between the two types of objects has significant impact on the clustering accuracy. In Figure 3 we empirically analyze how clustering accuracy evolves when the two types of objects are interrelated from slightly to strongly. In this experiment, we randomly select 20%, 40%, 60%, 80% and 100% of the user/web-page relationships to represent different degree of how objects are interrelated. The parameter α is fixed to 0.4 here.



(a)



(b)

Figure 3. Clustering accuracy when dataset evolves from slightly interrelated to strongly interrelated

The results show that the degree of how tightly the objects are interrelated with each other has significant impact on the clustering accuracy. When objects become more strongly interrelated, our iterative the clustering results may improve in accuracy. But improvement will decrease, as we can see that when the relative density of relationships is larger than 80%, the clustering accuracy is very close to that of 100% relationships. More relationships will not help much compared to the cost of computational complexity.

Figure 4 illustrates the robustness of our algorithm by adding noises into those page hits. The result shows when the ratio of noise increase, the clustering accuracy increases accordingly. This means, the iterative reinforcement between object types may not generate significant improvement to accuracy when the ratio of noise relationship is exceeding some points.

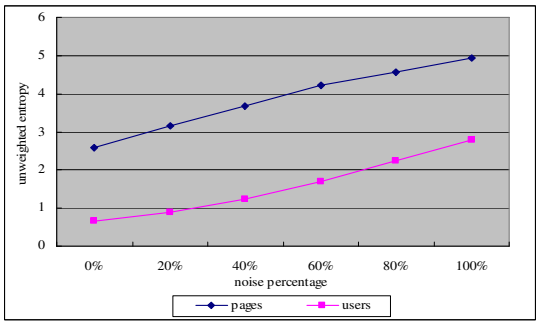


Figure 4. Clustering accuracy along with noise ratio

The next two comparative studies will show the effectiveness of weighted clustering and our link analysis algorithm under noise. Figure 5 shows that a clustering algorithm which incorporates the weights of objects will obtain better final clustering results, especially, in large noise ratios. By considering their importance value, important users or web-pages will have larger impact in forming the correct clusters. This may decrease the influence of the noise relationships.

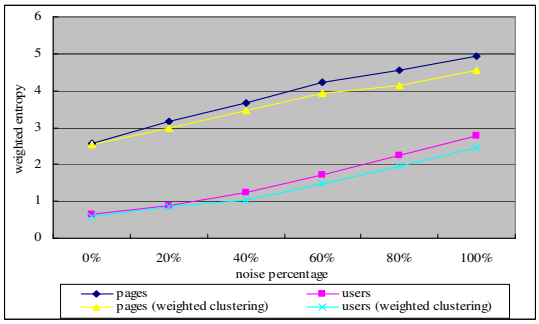


Figure 5. Comparison between importance weighted clustering and un-weighted clustering

From these experiments we can draw the conclusion that the weighted iterative clustering in our framework behaves well compared to traditional approaches. The introduction of relationship in defining object features and reinforcement clustering by relationship proves effective for the interrelated multi-type data objects.

3.2 Real Data

The algorithm is also carried on a real data, which is the Web log of CS Division of UC Berkeley in Nov. 2001. The log could be downloaded online from http://www.cs.berkeley.edu/logs/.

After preprocessed, the Berkeley log contains 19,057 URLs, 25,591 users and 103,794 visit records. Each user is identified by the IP address. This is not appropriate sometimes when multi-users share one IP address, or user accesses web by dynamic IPs. Thus in our experiment, the web-page clusters and user clusters exhibit visit pattern of a group of users. The noise is reduced by removing the web-pages which was visited no more than 3 times, and user IPs who visited no more than 3 pages. The cluster numbers of web-pages and users are 20 and 15 respectively.

To evaluate our algorithm, we partitioned the preprocessed Web logs into two parts, with 2/3 being the training data and 1/3 being test data. The partition ensured each URL and each IP occurred in

both training data and test data. The clustering algorithm is run on the training data, and the increment of log-likelihood of test data is calculated and depicted in Figure 6.

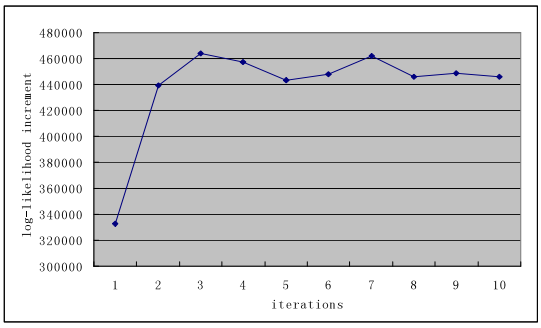


Figure 6. The precision of hits prediction

The vertical axis denotes the increment of log-likelihood of test data in each iteration of clustering from that without clustering. We may see that the improvement is about 48.7%. In particular, we assume a probability distribution identical to the frequencies of links among resulting clusters and calculate the product of likelihood of each web-page/user pair in the testing data. This result is then divided by the likelihood of the same test set when the web-pages and users have not been clustered, where we assume a uniform probability distribution in all the web-page/user pairs. To make the value convenient to be computed and displayed, we calculate all the likelihood in their logarithm values. From this figure, we could see the iterative clustering algorithm outperforms the traditional k-means clustering (the result in the first iteration).

3.3 Discussion of Time Complexity

Practically, our algorithm will converge and achieve satisfying clustering results in 5~6 iterations. So, intuitively, the time complexity of our algorithm should be 5~6 times that of traditional clustering algorithms. But in practice, since in iterations, the dimensionality of relationship feature is greatly reduced from the number of objects to the number of clusters, the real time spent of our algorithm is 3~4 times that of the traditional clustering algorithm, which is acceptable compared to the significant improvement in clustering accuracy.

4. RELATED WORK

Rooted in mathematics, clustering is an active research domain which aims at distributing data into different sets based on certain attributes. According to the divisions, we can learn some hidden relationships behind the raw data. So clustering is often utilized as a knowledge-discovered tool [12] or an efficient method to organize documents as related collections [8].

[1][15] provided a comprehensive review of different clustering techniques in data mining. Roughly they can be divided in two categories: hierarchical methods and partitioning methods. Agglomerative/Divisive algorithm belongs to the former and k-means is the latter. No matter in what category, a distance/similarity function must be pre-defined as the criterion of clustering.

Traditional clustering methods only consider content features of the objects to calculate the distance function. For the variety of objects on web, more attributes are gradually added into the clustering model. [16][19] demonstrated a correlation-based document clustering method which measures the similarity between web-pages only by users' usage logs. [4] described models for text, hypertext and semi-structured data, which can be used to represent the WWW environment. [10] extracted valuable information from user access by clustering multi-modal vectors that encompass various sources of information involving content, topology and URL.

The relationship in data set is ubiquitous in the real world which can offer lots of valuable information. Intuitively if two objects are related together and one of them has a property, it can be inferred that the other should also share that property. [13] presented an iterative classification procedure which implements this idea exactly. Based on the fully observed relational training set, the classifier labels the related test instances and accepts those with high confidence. Moreover, the idea of propagation has also been applied in the field of collaborative filtering successfully. [2] tries to recommend people with products he or she maybe likes. The estimation only comes from the user's preference patterns, which are usually represented as the connection between users and items, instead of the contents of items (e.g. word, author, description). By finding other people who have similar interests, collaborative filtering systems can return the items those similar users selected.

All clustering methods mentioned above are imposed on the single type data objects and accept an assumption that instances are independent and identically distributed which can be represented as "flat" data with a fixed-length vector of attribute values. Recent researches begin to consider the analysis of multi-type data and the relation structure together in some probabilistic forms. [6] combined two probabilistic models together and dealt with content and connectivity of documents in one formula. The theoretical foundation is that "the classification accuracy of the joint model is greater than that of either model in isolation, indicating that the content and link structure of a document collection do indeed corroborate each other". [20] proposed a two-layer model for clustering Web objects. [17] proposed a general class of models for classification and clustering which is powerful in explicating the multiple types of instances and complex relationships between them. Their work builds on the framework of Probabilistic Relational Models that extend Bayesian networks to a relational setting. Since the data cannot be treated as a set of independent instances corresponding to the objects in the model, they have to assume that the structure of the model is given and thus estimate parameters, such constraints limit the application of the model.

[8][11] etc. use the links between instances to identify clusters using some kind of spectral approaches. Besides those attributes used as the criterion in the calculating procedure, other issues may also influence the clustering results indirectly. Kleinberg's HITS algorithm [11] and Google's PageRank algorithm [3] enlightened us in this way. Both of them shared a common belief that different web-pages are not of equal importance and those which are cited more frequently by more important web-pages will get higher importance scores. So the importance of objects may offer helpful information for our clustering despite it is not a dominant factor.

5. Conclusions

In this paper, we defined the problem of clustering multi-type interrelated data objects, and proposed a reinforcement approach to cluster a set of interrelated objects with different types. Using this approach, multiple types of objects are clustered separately and inter-type links are used to iteratively project the clustering results of one type to another until reaching a converged clustering result of all interrelated types of objects. The relationship information is also used to determine the importance of data objects to achieve better clustering performance. We have shown in experiments that the relationship information allows us to achieve substantially better accuracies than a standard "flat" clustering scheme. We achieve 31% and 53% in entropy reduction in clustering the web-pages and Web user in a simulated web log data.

6. REFERENCES

[1] P. Berkhin, Survey of Clustering Data Mining Techniques, http://www.accrue.com/products/researchpapers.html, 2002

[2] J. S. Breese et al, Empirical Analysis of Predictive Algorithms for Collaborative Filtering, Technical report, Microsoft Research, 1998

[3] S. Brin and L. Page, The Anatomy of a Large-Scale Hypertextual Web Search Engine, in Proc. of the 7th international World Wide Web Conference Vol.7, 1998

[4] S. Chakrabarti, Data Mining for Hypertext: A Tutorial survey, In ACMSIGKDD Explorations, 2000

[5] L. Chen and K. Sycara, "Webmate: A personal agent for browsing and searching," In Proceedings 2nd Intl. Conf. Autonomous Agents, pp. 132--139, 1998

[6] D. Cohn & T. Hofman, The Missing Link – A Probabilistic Model of Document Content and Hypertext Connectivity, in Proc. Neural Information Processing Systems, 2001

[7] T. M. Cover and J. A. Thomas, Elements of Information Theory, Wiley, 1991

[8] I. Dhillon et al, Efficient Clustering of Very Large Document Collections, In Data Mining for Scientific and Engineering Applications, Kluwer Academic Publishers, 2001.

[9] D. Gibson, J. Kleinberg, and P Raghavan. Inferring Web communities from link topology, In Proc. 9th ACM Conference on Hypertext and Hypermedia, pages 225-234, 1998.

[10] J. Heer and E. H. Chi, Identification of Web User Traffic Composition Using Multi-Modal Clustering and Information Scent, in 1st SIAM ICDM, Workshop on Web Mining, Chicago, 2001

[11] J. Kleinberg, Authoritative Sources in a Hyperlinked Environment, in Proc. of the 9th ACM-SIAM Symposium on Discrete Algorithms, 1998.

[12] B. Liu et al, Clustering Through Decision Tree Construction, the 9th International Conference on Information and Knowledge Management (CIKM), 2000

[13] J. Neville and D. Jensen, Iterative Classification in Relational Data, In Proc. AAAI-2000 Workshop on Learning Statistical Models from Relational Data, AAAI Press, 2000

[14] S. Slattery and M. Craven, Combining statistical and relational methods in hypertext domains. In Proc.ILP, 1998.

[15] M. Steinbach et al, A Comparison of Document Clustering Techniques, in 6th ACM SIGKDD, World Text Mining Conference, Boston, 2000

[16] Z. Su et al, Correlation-based Document Clustering using Web Logs, In Proc. of the 34th Hawaii International Conference On System Sciences (HICSS-34), 2001.

[17] B. Taskar et al, Probabilistic Classification and Clustering in Relational Data, in Proc. of IJCAI-01, 17th International Joint Conference on Artificial Intelligence, 2001

[18] L. H. Ungar, D.P.Foster, Clustering Methods for Collaborative Filtering, In Workshop on Recommendation System at the 15th National Conference on Artificial Intelligence, 1998.

[19] J. Wen, J.Y. Nie, H. Zhang, "Query Clustering Using User Logs," ACM Transactions on Information Systems, 20 (1): 59-81, 2002.

[20] H. Zeng et al, A Unified Framework for Clustering Heterogeneous Web Objects, in Proc. of the 3rd International Conference on Web Information System Engineering, Singapore, 2002

[21] Open Directory Project, http://dmoz.org/