Multi-type Features Based Web Document Clustering

Shen Huang[1], Gui-Rong Xue[1], Ben-Yu Zhang[2], Zheng Chen[2], Yong Yu[1], and
Wei-Ying Ma[2]

[1] Department of Computer Science and Engineering, Shanghai Jiao Tong University,
1954 Huashan Ave., Shanghai, 200030, P.R.China
{shuang, yyu}@cs.sjtu.edu.cn, grxue@sjtu.edu.cn
[2] Microsoft Research Asia, 5F, Sigma Center
49 Zhichun Road, Beijing, 100080, P.R.China
{byzhang, zhengc, wyma}@microsoft.com

Abstract. Clustering has been demonstrated as a feasible way to explore the contents of document collection and organize search engine results. For this task, many features of Web page, such as content, anchor text, URL, hyperlink etc, can be exploited and different results can be obtained. We expect to provide a unified and even better result for end users. Some work have studied how to use several types of features together to perform clustering. Most of them focus on ensemble method or combination of similarity. In this paper, we propose a novel algorithm: Multi-type Features based Reinforcement Clustering (MFRC). This algorithm does not use a unique combine score for all feature spaces, but uses the intermediate clustering result in one feature space as additional information to gradually enhance clustering in other spaces. Finally a consensus can be achieved by such mutual reinforcement. And the experimental results show that MFRC also provides some performance improvement.

1 Introduction

World Wide Web grows with an unbelievable speed [12]. In such situation, autonomous or semi-autonomous methods for Web document clustering become more and more indispensable [21]. Clustering helps users tackle the information overload problem in several ways [7]: explore the contents of a document collection; organize and browse the result list returned by search engine; group duplicate and near duplicate documents. However, such unsupervised method can hardly achieve a good performance when evaluated using labeled data [7]. On the other hand, Web document has many different types of features including content, anchor text, URL, hyperlink etc. Using different kinds of feature, the clustering result will be somewhat different. We dedicate to find the optimal method which effectively exploits all kinds of features to get more consistent and better clustering results.

Many research focus on clustering using multi-type features. One intuitive way to combine results based on different features is called as *ensemble clustering* [4][10][15][16], which combines multiple partitionings of objects without accessing

X. Zhou et al. (Eds.): WISE 2004, LNCS 3306, pp. 253–265, 2004.

the original features. Such algorithms do not care how to get the different sets of partitions and can be smoothly exploited by clustering based on multi-type features. Ensemble clustering attempts to solve the problem that no original features available and only label information can be used to get a consensus result. Different with these work, what we try to solve is how to effectively combine multi-type features. Another method is to combine the similarity based on different features. For example, the similarity based on hyperlink feature has been integrated with content similarity in some work to improve clustering [5][18][19]. The main problem for similarity combination is the weights of different features are hard to determine. Linear regression is a choice, but as we show in experiment section, it does not work well in clustering task.

The ideas of Co-Training [1] and Co-EM [11] enlighten us to propose a new combination method. Co-Training is applied to learning problems that have a natural way to divide their features into subsets each of which are sufficient to learn the target concept. Co-EM is a semi-supervised, multi-view algorithm that use the hypothesis learned in one "view" to probabilistically label the examples in the other one. Similarly, we use the intermediate clustering result in one feature space as additional information to enhance clustering in other spaces. Thus different types of features are taken into account simultaneously and reinforce each other. We call it *mutual reinforcement*. Figure 1 shows an example for this idea: two Web pages are similar in URL feature space and help us find some implicitly related terms in their content. In step 1, this information can be exploited by the clustering in content space. Vice versa, in step 2, the newly found related URL terms will benefit the clustering in URL space.

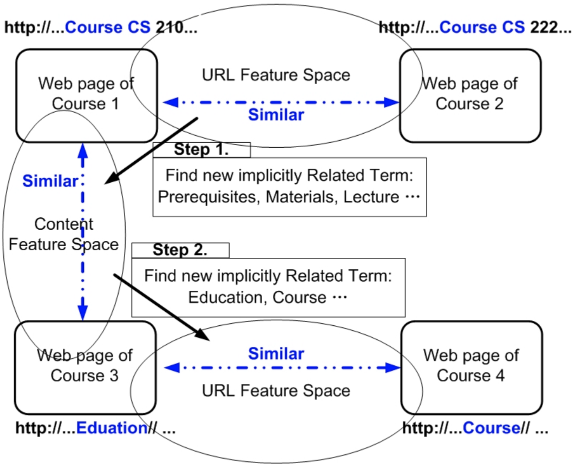

Fig. 1. An example for mutual reinforcement relation between content and URL feature.

We implement the mutual reinforcement idea in two steps. First, we transfer the similarity among different feature spaces by feature construction method. Then, we borrow the idea in [8] to generate *pseudo class* using the new feature vectors created in the first step. With such class label, supervised feature selection[1] can be done to improve performance. Feature construction and pseudo class based feature selection are both implemented in an iterative way. So they can be well integrated into an iterative clustering algorithm, such as K-means. We call such method *Multi-type Features based Reinforcement Clustering (MFRC).* The main contributions of this work are:

- A novel reinforcement method is proposed to combine multiple clustering results based on multi-type features. The intermediate clustering results are considered to be useful for the clustering in the next iteration.
- Feature construction and feature selection are performed during the mutual reinforcement process. The feature dimensions can be reduced largely and clustering is speed up while performance is improved.
- This method is evaluated using two WebKB benchmarks and one ODP dataset. Experimental results showed that our approach can work well in most cases.

The rest of the paper is organized as follows: Section 2 presents some related work. In Section 3, we propose the general idea of mutual reinforcement clustering. Section 4 introduces feature construction and pseudo class based feature selection in MFRC. Section 5 shows the experimental results on three datasets. Finally, we give the conclusion and the directions of future work in Section 6.

2 Related Work

Ensemble clustering attempts to combine multiple partitionings of a set of objects into a single consolidated clustering without accessing the features or algorithms that determined these partitionings. Minaei et al. [10] showed how consensus function operated on the co-association matrix. Topchy et al. [16] presented a consensus function in the space of standardized features could effectively maximize mutual information. Strehl et al. [15] reduced the problem of consensus clustering to finding the minimum cut of a hypergraph. Dudoit et al. [4] attempted to solve the correspondence problem and used a majority vote to determine the final consensus partition. Our idea in this paper differs in that it uses original features to achieve better performance.

In similarity-based clustering, similarity combination is a common method to exploit the different features of an object. Weiss et al. [19] proposed a new document similarity function based on both term similarity and hyperlink similarity factor. He et al. [5] also considered co-citation [14] relations and linearly combined co-citation similarity with text and hyperlink similarity. Wang et al. [18] used co-citation and co-coupling [6] to build combined feature vectors for K-means clustering. To our best knowledge, previous approaches get unique combined similarity for clustering, which is not used in our method. We take several different features into account simultaneously and let them reinforce each other in an iterative way.

[1] In this paper, feature selection is limited in single-type feature. Different types of features are selected separately.

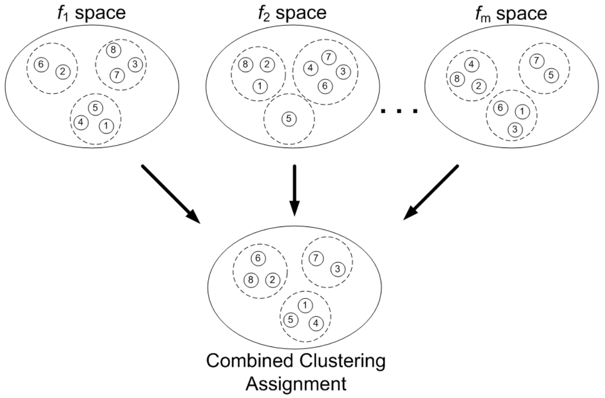

Fig. 2. Multi-type feature based clustering: combine clustering results in different feature spaces.

Blum and Mitchell [1] assumed that the description of each example could be partitioned into several distinct "views", each of which is sufficient for learning. All of the views can be used together to allow inexpensive bootstrapping. Co-EM [11] also explores the knowledge acquired in one view to train the other view. The major difference is that Co-EM uses a probabilistic model and does not commit to a label for the unlabeled examples. The "view" is similar to the different types of features discussed in this paper and we attempt to exploit the information acquired during the clustering procedure. We adopt a mutual reinforcement idea which has some relation to [17][22]. Zeng [22] and Wang [17] et al. introduced novel frameworks to cluster the heterogeneous data simultaneously. Under the frameworks, relationships among data objects are used to improve the cluster quality of interrelated data objects through an iterative reinforcement clustering process. Different from their work, our idea aims to clustering the same type of data objects. The mutual reinforcement is applied among multiple types of features, not heterogeneous types of data.

Traditional unsupervised feature selection method which does not need the class label information can be easily applied to clustering, such as Document Frequency (DF) and Term Strength (TS) [20]. Recently, there are some newly proposed methods, for example, entropy-based feature ranking method is proposed by Dash and Liu [3]. Martin et al. [9] introduced an Expectation-Maximization algorithm to select the feature subset and the number of clusters. For supervised feature selection, Liu et al. [8] used some methods to iteratively select features and perform text clustering simultaneously since no label information is available in clustering. Our idea differs with previous work in that we let feature selection in one feature space optimized by intermediate information generated in other spaces.

3 Problem Description and Mutual Reinforcement Clustering

Before introducing feature construction and feature selection in multi-type features based clustering, we first describe the multi-type features based combination problem and mutual reinforcement among different types of features. Suppose *m* feature spaces are available in the clustering task, f_k is the *kth* feature space. What we try to solve is how to combine the results from the *m* feature spaces into a unified one, as Figure 2 shows. And we expect the combined one will outperform the single ones.

Next, we'll introduce our mutual reinforcement mechanism to combine multi-type features. We assume that during an iterative clustering process, such as K-means clustering, additional information generated in one feature space will help the clustering in others. The general reinforcement algorithm is listed in Figure 3. In this algorithm, we first let clustering based on different features progress separately. Once some new information is achieved, it will be exploited by other features in the next iteration.

```
Loop for $n$ iterations
{
  Loop for $m$ features space
  {
    For $kth$ feature space
    If it's the first iteration then use original fea-
    ture vector $f_k orig$ to do clustering
    Else use both original vector $f_k orig$ and new vector
    $f_k new$ to do clustering
  }
  Construct or revise new feature vector $f_k new$ for each
  object
  Loop for $m$ features space
  {
    For $kth$ feature space
    Get combined pseudo class and select features (Op-
    tional)
    Calculate New Centroids using both $f_k orig$ and $f_k new$
  }
}
```

Fig. 3. Mutual reinforcement algorithm. Feature construction and selection are integrated into iterative clustering

4 Feature Construction and Feature Selection

4.1 Feature Construction for MFRC

In this section, we present how to exploit the intermediate information during the clustering in MFRC. We use Vector Space Model (VSM) to measure the similarity between document vectors constructed by TF×IDF [13]. After one iteration of clustering, each data object will be assigned to a cluster. The new feature is composed by the similarity between a data object and the centroids of different clusters. Suppose we should categorize data objects into l clusters in feature space k, then we can get a new feature vector for each object like formula (1).

$$f_k new = [f_k CSim_1, f_k CSim_2, ..., f_k CSim_l] \quad . \tag{1}$$

where $CSim_l$ is the similarity between a object and centroid l, f_k means the feature in space k. For clustering algorithm using geometric metric, this can be explained intuitively as Figure 4 shows: two data objects locate very near will have very similar $f_k new$ in formula (1). If the centroids are enough, the objects having similar vectors will also locate very near. For the task only few clusters should be formed, we choose some additional representative objects from each cluster and get new vector in formula (2):

$$f_k new = [f_k CSim_1, f_k CSim_2, ..., f_k CSim_l, f_k RSim_1, f_k RSim_2, ..., f_k RSim_r] \quad . \tag{2}$$

where $RSim_r$ is the similarity between a object and representative object r. The experiment shows that about 1% samples from the dataset are enough to assure the performance of clustering.

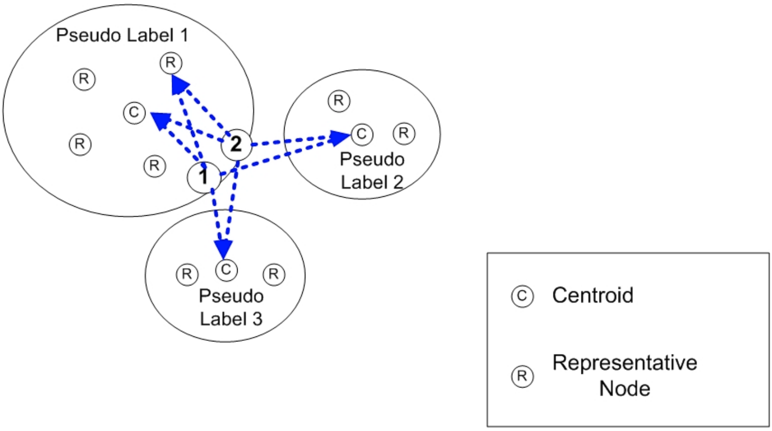

Fig. 4. The closeness of two points can be characterized using centroids and some other representative nodes.

After that, data objects in space k will get a combined vector like formula (3).

$$[...][f_{k-1} CSim_1, ..., f_{k-1} RSim_r] f_k orig [f_{k+1} CSim_1, ..., f_{k+1} RSim_r][...] \quad . \tag{3}$$

where $f_k orig = [f_k v_1, f_k v_2, \ldots, f_k v_n]$ is the original vector in space k. Finally each centroid should be updated using this combined vector. Using cosine metric, the combined similarity between an object o and centroid c in space k should be calculated using equation (4), where α and β are the weights of combination, m is available feature spaces, $ComSim_k(o,c)$ is the combined similarity in feature space k, $f_k Sim(o,c)$ is original vector similarity and $f_i Sim(o,c)$ ($i \quad k$) is the new vector similarity. With the combined similarity, objects will be reassigned in the next iteration, as algorithm in Figure 3 shows.

$$\begin{cases} ComSim_k(o,c) = \alpha \times f_k Sim(o,c) \; + \; \beta \times \dfrac{\sum_{i=1}^{m} f_i Sim(o,c) \qquad (i \neq k)}{m-1} (\alpha = 0.7, \beta = 0.3) \\ f_k Sim(o,c) = \dfrac{f_k orig(o) \bullet f_k orig(c)}{\sqrt{| f_k Orig(o) |^2 + | f_k Orig(c) |^2}} \\ f_i Sim(o,c) = \dfrac{f_i new(o) \bullet f_i new(c)}{\sqrt{| f_i New(o) |^2 + | f_i New(c) |^2}} \end{cases} . \qquad (4)$$

By equation (4), different feature space will generate different clusters in following iterations of clustering. However, similarity between any two objects in one feature space will be propagated to other spaces in low cost. The result sets will gradually converge to a unified one. The combination weights will control the speed of convergence. We'll show this in experiment section.

4.2 Feature Selection

Another potential method to exploit additional information is pseudo class based feature selection. First, it should be make clear that the selection of each type feature is limited in itself space. For supervised feature selection, one obstacle is the unavailability of label. Liu et al. [8] proved that this problem can be partially addressed by combining effective supervised feature selection method with iterative clustering. Here we expand this idea and integrated it with mutual reinforcement process.

After one iteration of clustering, each data object will be assigned to a cluster. We assume that although the undetermined assignments tend to be erroneous, such preliminary result still provide some valuable information for feature selection. Each cluster is corresponded to a real class and called *pseudo class*. Recall that each dimension of the new vector introduced in Section 4.1 is the similarity between an object and a centroid of a cluster. We normalized such vector so that the sum of each dimension is 1.0. Each dimension is treated as the confidence a data object belongs to corresponding cluster. For example, given a new vector [0.34, 0.23, 0.71], the normalized one will be [0.27, 0.18, 0.55]. That means the confidence that an object belong to cluster 1, 2 and 3 are 0.27, 0.18 and 0.55 respectively. If we match the clusters in different feature spaces, we can get a combined confidence that an object belong to a cluster, or pseudo class. The combined pseudo class is the one with max combined confidence. And it is used to conduct feature selection.

To combine pseudo class, the correspondence of clusters should be solved. We use a F1-Measure to estimate the degree of match between two clusters. For any two clusters C_1 and C_2, let N_1 be the number of objects belong to both cluster C_1 and C_2, N_2 be the number of objects belong to cluster C_1 and N_3 be the number of objects belong to cluster C_2:

Precision(C_1, C_2) = N_1 / N_2, Recall(C_1, C_2) = N_1 / N_3

$$F(C_1, C_2) = \frac{2PR}{P+R} \quad . \tag{5}$$

where *F* is F1-Measure, *P* is Precision and *R* is Recall. Actually, *Precision* and *Recall* are not appropriate names here. But we don't use new ones to avoid confusion. This measurement is also used to evaluate the clustering performance in our experiments.

Again, suppose we should categorize data objects into *l* clusters using *m* feature spaces. We get combined pseudo class using following equation:

$$\begin{cases} Conf_k(C_j \mid o) = \dfrac{f_k Sim(o, c_j)}{\sum_{i=1}^{l} f_k Sim(o, c_i)} \\ ComConf_k(C_j \mid o) = \alpha \times Conf_k(C_j \mid o) + \beta \times \sum_{i=1}^{m} Conf_i(C_j \mid o) \quad (i \neq k) \qquad (\alpha = 0.7, \beta = 0.3) \\ C_k(o) = \arg\max_j (CombConf_k(C_j \mid o)) \end{cases} \quad . \tag{6}$$

where c_j is the centroid of cluster C_j, $Conf_k(C_j \mid o)$ is the confidence that object *o* belong to cluster C_j in space *k*, $ComConf_k(C_j \mid o)$ is the combined confidence that *o* belong to cluster C_j in space *k*, $C_k(o)$ is combined pseudo class for *o* in space *k*.

Having the label information, we do feature selection using Information Gain (IG) and χ^2 statistic (CHI) [20]. Information gain measures the number of bits of information obtained for category prediction by the presence or absence of a feature in a document. Let *l* be the number of clusters. Given vector $[f_k v_1, f_k v_2, \ldots, f_k v_n]$, the information gain of a feature fv_n is defined as:

$$\begin{aligned} IG(fv_n) = & -\sum_{i=1}^{l} p(C_i) \log p(C_i) \\ & + p(fv_n) \sum_{i=1}^{l} p(C_i \mid fv_n) \log p(C_i \mid fv_n) \\ & + p(\overline{fv_n}) \sum_{i=1}^{l} p(C_i \mid \overline{fv_n}) \log p(C_i \mid \overline{fv_n}) \end{aligned} \quad . \tag{7}$$

χ^2 statistic measures the association between the term and the category. It is defined to be:

$$\begin{cases} \chi^2(fv_n, C_i) = \dfrac{N \times (p(fv_n, C_i) \times p(\overline{fv_n}, \overline{C_i}) - p(fv_n, \overline{C_i}) \times p(\overline{fv_n}, C_i))^2}{p(fv_n) \times p(\overline{fv_n}) \times p(C_i) \times p(\overline{C_i})} \\ \chi^2(fv_n) = \underset{i=1}{\overset{m}{avg}} \{\chi^2(fv_n, C_i)\} \end{cases} \quad . \tag{8}$$

After the feature selection, objects will be reassigned, features will be re-selected and the pseudo class information will be re-combined in the next iteration. Finally, the iterative clustering, feature selection and mutual reinforcement are well integrated.

5 Experiment

We conducted experiments to demonstrate that MFRC can improve the clustering performance when evaluated by entropy and F-Measure. K-means was chosen as our basic clustering algorithm. For this algorithm tends to influenced by selection of initial centroids, we randomly selected 10 sets of initial centroids and averaged the performances in the 10 times as the final result. TF×IDF [13] with "ltc" scheme was used to calculate the weight of each vector dimension.

5.1 Data Sets

Our evaluation approach measures the overall quality of generated clusters by comparing them with a set of categories created manually. We use three test sets:

- Co-Training (CT): A subset of the 4 Universities dataset containing web pages and hyperlink data[2]. It's used for the Co-Training experiments by Blum et al. [1].
- WebKB (WKB): A data set consisting classified Web pages[3] for Web->KB project[4].
- Open Directory Project (ODP): A data set in Open Directory Project[5], including user access log of it from MSN search engine. We use the user access as one feature for Web document clustering.

The information about these data sets is shown in Table 1:

Table 1. The test collections and some statistics. "Feature Type Num" means the number of different feature types.

Test Set	Class Num.	Doc Num.	Terms Num.	Average Terms Per Doc	Feature Type Num.
CT	2	1,051	38,991	37.1	3 (content, URL, anchor text)
WKB	4	5,396	205,683	38.1	2 (content, URL)
ODP	15	8,071	109,569	13.6	3 (content, URL, user access)

5.2 Performance Measures

Two kinds of measurements, entropy and F-Measure were used to evaluate the clustering performance. Entropy is based on the entropy in information theory [2], 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

[2] http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-51/www/co-training/data/

[3] http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/

[4] http://www-2.cs.cmu.edu/~webkb/

[5] http://www.dmoz.org

$$H(A) = -\sum_{j} p_j \bullet \log_2 p_j \quad . \tag{9}$$

where p_j should be the proportion of *jth* class's data in the cluster.

F-Measure has been used in [7][17]. Since each cluster always consists of documents from several classes, we use an overall F-Measure as follows:

$$OverallFMeasure = \frac{\sum_{j=1}^{l} (|C_j| \arg\max_i F(C_j, Class_i))}{\sum_{j=1}^{l} |C_j|} \quad . \tag{10}$$

where $F(C_j,Class_i)$ is the F-Measure of cluster C_j when class i is used as the correct categorization to evaluate C_j. $|C_j|$ is the number of documents in cluster C_j. F-Measure can also be used to measure the degree of match between two clusters, as mentioned in Section 4.2.

5.3 Results and Analysis

To combine clustering based on different types of feature, we expect to get a unified result set. First, let's have look at the convergence of feature construction. We Use micro F1-Measure to compare two sets of clustering result, 1.0 means the two sets are totally equal. In most of the tests (>90%), F1-Measure becomes larger than 0.99 within 10 iterations, which means the different features get a consensus quickly. Figure 5 shows the convergence on three data sets.

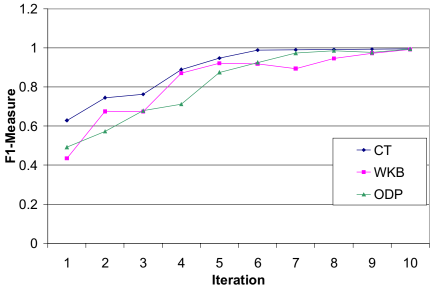

Fig. 5. The convergence of feature construction. The weights of combination are α = 0.7, β = 0.3. Given n types of features, n(n-1)/2 F-Measure values exist. We use the average one.

As to feature selection method, the consensus can't be achieved. As Liu et al. showed in [8], 10 iterations are enough for the iterative feature selection to get a good performance. We try similar test and use the combined pseudo class in the 10th iteration as the final result. Different percentages of selected features are tested and Figure 6 is obtained. In most cases, 2-percentage selection will get best performance.

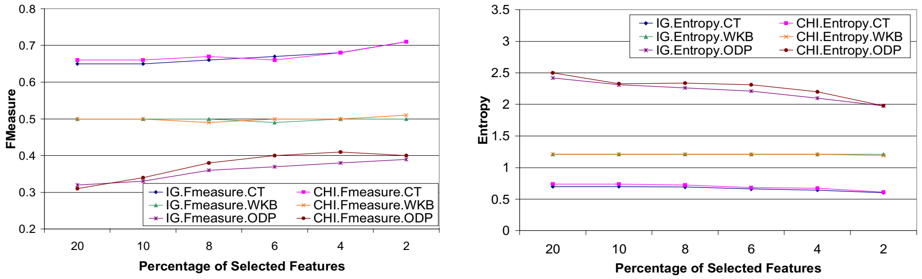

Fig. 6. FMeasure and entropy comparison on three data sets with different percentage of selected features.

In Table 2, we compare the performance of our approach with other methods, including the ones using single feature and linear combination of similarity. Besides averagely assigning weights to different types of feature, we use *linear regression* to choose the weights for similarity combination. However, we find it's not easy for clustering task to get training samples. With manually labeled data, i.e. the answer for evaluation, we use two methods to build samples. The first one uses the distance between a data object and class centroid as input, 1 (object belongs to this class) or 0 (object doest not belong to this class) as output. The second one uses the distance between any two objects as input, 1 (the two objects belong to the same class) and 0 (the two objects belong to different class) as output. We call the two methods LRC and LRP respectively. For both methods, the residues in linear regression are not small enough. Using the weight chosen by such methods, the performance may even worse than the best one based on single type of feature.

Table 2. Performance Comparsion. Due to space limit, we show only the best performance for single type of feature. AVG means assign average weights to each type of feature. The represenstive nodes in feature concstruction is 1%. Percentage of seleted features in feature selection is 2.

Test Sets	Measure	Best with Single Feature	Linear Combination			MFCMR		
			AVG	LRC	LRP	FC (1%)	FS (2%)	
							IG	CHI
CT	FMeasure	0.69	0.67	0.66	0.65	0.72	0.71	0.71
	Entropy	0.65	0.72	0.73	0.73	0.63	0.60	0.61
WKB	FMeasure	0.46	0.42	0.44	0.44	0.53	0.50	0.51
	Entropy	1.28	1.30	1.29	1.32	1.30	1.21	1.19
ODP	FMeasure	0.27	0.20	0.23	0.24	0.36	0.39	0.38
	Entropy	2.68	2.65	2.77	2.67	2.32	1.98	2.19

6 Conclusion and Future Work

We proposed Multi-type Feature based Reinforcement Clustering (MFRC), a novel algorithm to combine different types of features to do Web document clustering. It contains two main parts: feature construction and pseudo class based feature operations. We use the intermediate clustering result in one feature space as additional information to enhance clustering in other spaces. Besides, pseudo class is used for feature selection to improve performance. The two parts are all implemented in an iterative way and can be well integrated into an iterative clustering algorithm.

In future, we need to prove the convergence of feature construction and test MFRC on more data sets. Besides, the reinforcement idea will be tested using some clustering algorithms other than K-means, e.g. soft clustering, hierarchical clustering and density-based clustering.

References

1. Blum, A. and Mitchell, T.: Combining Labeled and Unlabeled Data with Co-Training. In Proceeding of the Conference on Computational Learning Theory, 1998.
2. Cover, T. M. and Thomas, J. A.: *Elements of Information Theory*, Wiley, 1991.
3. Dash, M. and Liu, H.: Feature Selection for Clustering. In Proceeding of 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2000.
4. Dudoit, S. and Fridlyand, J.: Bagging to Improve the Accuracy of a Clustering Procedure. Bioinformatics, 2003.
5. He, X., Zha, H., Ding, C. and Simon, H. D.: Web Document Clustering Using Hyperlink Structures. *Computational Statistics and Data Analysis*, 45:19-45, 2002.
6. Kessler, M. M.: Bibliographic coupling between scientific papers. American Documentation, 14:10-25, 1963.
7. Larsen, B., Aone, C.: Fast and Effective Text Mining Using Linear-time Document Clustering. In Proceedings of the 5th ACM SIGKDD International Conference, 1999.
8. Liu, T., Liu, S., Chen, Z. and Ma, W.-Y.: An Evaluation on Feature Selection for Text Clustering. In Proc. of the 20th International Conference on Machine Learning, 2003.
9. Martin, H. C. L., Mario, A. T. F. and Jain, A.K.: Feature Saliency in unsupervised learning, Technical Report, Michigan Sate University, 2002.
10. Minaei, B., Topchy, A., Punch, W. F.: Ensembles of Partitions via Data Resampling. In Proceeding of the International Conference on Information Technology, 2004.
11. Nigam, K. and Ghani, R.: Analyzing the Effectiveness and Applicability of Co-Training. In Proceeding of Information and Knowledge Management, 2000.
12. Ntoulas, A., Cho, J. and Olston, C.: What's New on the Web? The Evolution of the Web from a Search Engine Perspective. To appear: the 13th International WWW, 2004.
13. Salton, G.: Automatic Text Processing: The transformation, analysis, and retrieval of information by computer. Addison-Wesley, 1989.
14. Small, H.: Co-citation in Scientific Literature: A new measure of the relationship between two documents. *Journal of the American Society for Information*, 1973.
15. Strehl, A. and Ghosh, J.: Cluster Ensembles – A Knowledge Reuse Framework for Combining Multiple Partitions. *Journal on Machine Learning Research*, 2002.

16. Topchy, A., Jain, A. K. and Punch, W.: A Mixture Model of Clustering Ensembles. To appear in Proceedings of the SIAM International Conference on Data Mining, 2004.
17. Wang, J., Zeng, H.-J., Chen, Z., Lu, H., Li, T. and Ma, W.-Y.: ReCoM: Reinforcement Clustering of Multi-Type Interrelated Data Objects. In Proc. of the 26th SIGIR, 2003.
18. Wang, Y. and Kitsuregawa, M.: Clustering of Web Search Results with Link Analysis. Technique report, 1999.
19. Weiss, R., Velez, B., Sheldon, M. A., Namprempre, C., Szilagyi, P., Duda, A. and Gifford, D. K.: HyPursuit: A Hierarchical Network Search Engine that Exploits Content-Link Hypertext Clustering. In 7th ACM Conference on Hypertext, pages 180-193, 1996.
20. Yang, Y. and Pedersen, J. O.: A Comparative Study on Feature Selection in Text Categorization. In Proceedings of 14th International Conference on Machine Learning, 1997.
21. Zamir, O., Etzioni, O.: Web Document Clustering: A Feasibility Demonstration. In Proceeding of the 21st Annual International ACM SIGIR Conference, 1998.
22. Zeng, H.-J., Chen, Z. and Ma, W.-Y.: A Unified Framework for Clustering Heterogeneous Web Objects. In Proc. of the 3rd International Conference on WISE, 2002.