A Clustering Framework Based on Adaptive Space Mapping and Rescaling

Yiling Zeng[1], Hongbo Xu[1], Jiafeng Guo[1], Yu Wang[1], and Shuo Bai[1,2]

[1] Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China
[2] Shanghai Stock Exchange, Shanghai, 200120, China
{zengyiling,hbxu,guojiafeng,wangyu2005}@software.ict.ac.cn,
sbai@sse.com.cn

Abstract. Traditional clustering algorithms often suffer from model misfit problem when the distribution of real data does not fit the model assumptions. To address this problem, we propose a novel clustering framework based on adaptive space mapping and rescaling, referred as M-R framework. The basic idea of our approach is to adjust the data representation to make the data distribution fit the model assumptions better. Specifically, documents are first mapped into a low dimensional space with respect to the cluster centers so that the distribution statistics of each cluster could be analyzed on the corresponding dimension. With the statistics obtained in hand, a rescaling operation is then applied to regularize the data distribution based on the model assumptions. These two steps are conducted iteratively along with the clustering algorithm to constantly improve the clustering performance. In our work, we apply the M-R framework on the most widely used clustering algorithm, i.e. k-means, as an example. Experiments on well known datasets show that our M-R framework can obtain comparable performance with state-of-the-art methods.

Keywords: Document Clustering; Space Mapping; Data Representation.

1 Introduction

With the explosion of documents on the Web, there has been increasing need for efficient and effective analysis methods to manage massive text collections. Document clustering, as one of the primary analysis techniques in text mining area, has then been applied in different kinds of IR tasks [2], e.g. speeding up the information retrieval procedure [1], improving the precision or recall in information retrieval systems [3], browsing a collection of documents [4], and organizing the search results for a given query [5].

However, the performance of clustering algorithms often suffers from the model misfit problem [7]. Most clustering algorithms are based on some underlying model assumptions. When real data fits the assumptions well, the performance of the clustering algorithm would be reasonably good, otherwise not. Typically, there are two kinds of approaches to addressing the model misfit problem: adjusting algorithms' strategies to deal with the real data distributions [7, 8], or applying space transformation (e.g., kernels) to alter the data representation (or distributions) [9, 10]. The first kind

G.G. Lee et al. (Eds.): AIRS 2009, LNCS 5839, pp. 363–374, 2009.

of approaches usually makes refinement on local regions where training errors occur, while the global characteristics of data distribution are often ignored. However, such global characteristics should be reasonably considered because they may be directly related to the model misfit problem. The second kind of approaches tries to apply space transformation to alleviate the problem. However, since space transformation is usually proposed based on some assumptions without the consideration of algorithm's model, the performance improvement may be limited. In this paper, we present a framework in the way of space transformation. However, unlike the previous approaches, our transformation is proposed with respect to the clustering algorithm's model assumptions.

The framework we propose is referred as M-R framework. It is a clustering framework based on adaptive space mapping and rescaling. Considering that real data distribution usually may not fit the model assumptions of clustering algorithms very well, our solution is to regularize the data distribution such that it is more consistent with the model assumptions. Specifically, the M-R framework consists of two important steps as follows.

Step 1: Space Mapping. Since the distribution features of data in high dimensional space are usually complicated and hard to analyze, we choose to map all documents into a low dimensional coordinate which is constructed with respect to the cluster centers. In this way, the distribution statistics of each cluster could be analyzed on the corresponding dimension.

Step 2: Rescaling. With those distribution statistics obtained in hand, we apply a rescaling operation to regularize the data distribution based on the model assumptions. In this way, we are able to make the data distribution more consistent with the model assumptions to help make better clustering decisions.

By conducting these two steps iteratively along with the clustering algorithm, we are able to constantly improve the clustering performance. In our paper, we apply our M-R framework on the most popular clustering algorithm, i.e. k-means, to verify the effectiveness of our framework.

The rest of this paper is organized as follows. Section 2 discusses related work and Section 3 proposes the M-R framework in detail. We apply our framework on k-means in Section 4 and present the experimental results in Section 5, which is followed by some concluding remarks in the last section.

2 Related Work

Different clustering algorithms [6] take different point of views of data spaces. Hierarchical clustering algorithms hold the assumption that any cluster is composed of smaller sub clusters that semantically related to each other, while partitioning algorithms (e.g., k-means) believe that clusters obey isotropic Gaussian distributions that are distributed in spherical regions with the same radius. Density-based algorithms and grid-based algorithms, however, focus on local attributes that restricted in an ε-neighborhood or in a small unit named grid, and make clustering decisions with these attributes. Actually, such ideal models more or less misfit real data distribution. Therefore, model misfit becomes a common problem in text mining. To address this problem, researchers make their efforts to figure out solutions in both algorithm layer and data presentation layer.

In algorithm layer, most research approaches require training errors to refine the model, and thus are mainly proposed in supervised learning area. Wu et al. [7] retrain a sub-classifier using the training errors of each predicted class with the same learning method to refine the algorithm models. Tan et al. [8] propose an effective and yet efficient refinement strategy to enhance the performance of text classifiers by means of on-line modification of the base classifier models. However, as in the clustering field, little work has been done for solving the model misfit problem due to the lack of labels. Generally, this kind of approaches improves performance by achieving local refinements, but the global feature of real data distribution is usually ignored.

As for approaches in the data representation layer, the basic idea is to apply space transformation such that certain problems in the old feature space could be properly solved. Dumais [1] proposes LSI decomposition to rectify the deficiency in VSM model that takes correlated terms as independent dimensions. Kernel method [9] aims to find a proper implicit mapping φ via kernel functions such that in the new space, problem solving is much easier. Another novel and important approach in feature space transformation for unsupervised learning is spectral clustering [10]. The basic idea of spectral clustering is to model the whole dataset as a weighted graph, and aims to optimize some cut value (e.g. Normalized Cut [11], Ratio Cut [12], Min-Max Cut [13]). Since those criterions could be guaranteed global optimums via certain eigen-decompositions, spectral clustering algorithms often achieve better results than traditional clustering algorithms. On the whole, this kind of approaches concentrates on solving existing problems in the current feature space. Most of them seldom account in the algorithm's model assumptions during the mapping procedure. And the computational and memory requirements are usually high.

3 M-R Framework

3.1 Model Misfit in Clustering

A simple but direct example about model misfit problem of clustering algorithm (e.g., k-means) is shown in Figure 1. There are two intrinsic clusters in the dataset that are marked out with dashed lines (labeled with "+" and "○" respectively). Points with label "+" distribute in a narrow elliptic region while points with label "○" distribute in a circular region. Obviously, the distribution of the dataset violates the underlying assumption of k-means. Therefore, without knowledge of the distribution characteristics, k-means probably organizes all the points into two clusters which are marked with the two solid circles, i.e. Cluster1 and Cluster2. As a result, part of the points with label "+" is assigned to the wrong cluster.

How can we avoid such kind of mistakes caused by model misfit? According to the decision criterion of k-means, a point should be assigned to the nearest cluster. Take a point **x** in Figure 1 as an example. Assume the distance from **x** to the centroid of Cluster1 is *d1*, and the distance to the centroid of Cluster2 is *d2*. Since *d2<d1*, document *x* will be assigned to Cluster2 improperly. To avoid the mistake, we can make a transformation according to the distribution characteristics, such that in the new scenario *d1<d2*, and thus point *x* will be assigned to Cluster1 more reasonably.

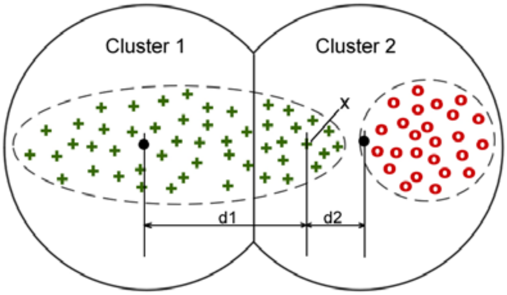

Fig. 1. An example of model misfit in K-means. Point x which should belong to Cluster1 is assigned to Cluster2 as a result of improper distance comparison.

An intuitive solution is to use Gaussian Mixture Model together with Expectation-Maximization [14] which would estimate the parameters of different distributions. However, the disadvantage is that GMM could not be directly applied in a sparse and high dimensional feature space and therefore high computational operation (i.e., LSI) is introduced to reduce the dimensionality. While in this paper, the M-R framework, as our proposed solution to address the model misfit problem, would not bring in any time consuming operations.

3.2 Rationale

In our solution of M-R framework, we analyze the data distribution and transform it to better fit the model assumptions. However, the analysis work in a high dimensional space is complicated and the sparsity feature of data distribution makes the analysis impractical. To solve this problem, we first map all documents into a lower dimensional space which is suitable for distribution analysis.

Suppose we have a dataset of n documents which contains k classes/clusters marked as $C_1,\ldots,C_i,\ldots,C_k$ $(1\leq i\leq k)$. The corresponding document numbers of these clusters are $n_1,\ldots,n_i,\ldots,n_k$. Let $\mathbf{m}_i = \frac{1}{n_i}\sum_{\mathbf{x}\in C_i}\mathbf{x}$ be the centroid of C_i, and $\mathbf{m} = \frac{1}{n}\sum_{\mathbf{x}}\mathbf{x}$ be the mass center of the dataset. According to the Fisher Linear Discriminant [15] for multiple classes, a matrix **W** constructed with the best set of discriminant vectors can be obtained by maximizing the *between-class scatter* $\left|\mathbf{W}^T\mathbf{S}_B\mathbf{W}\right|$ and minimizing the *within-class scatter* $\left|\mathbf{W}^T\mathbf{S}_W\mathbf{W}\right|$. This is equivalent to solve the criterion function defined as $J(\mathbf{W}) = \left|\mathbf{W}^T\mathbf{S}_B\mathbf{W}\right| / \left|\mathbf{W}^T\mathbf{S}_W\mathbf{W}\right|$, where $\mathbf{S}_B = \sum_{i=1}^{k} n_i(\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^T$ is the *between-class scatter matrix* and $\mathbf{S}_W = \sum_{i=1}^{k}\sum_{\mathbf{x}\in C_i}(\mathbf{x}-\mathbf{m}_i)(\mathbf{x}-\mathbf{m}_i)^T$ is the *within-class scatter matrix.* In our framework, we assume that the rescaling operation would constrain the within-class scatter properly. Therefore, to simplify the criterion function, only the maximization of the between-class scatter will be considered. Thus, our criterion function is defined as

$$J_{M-R}(\mathbf{W}) = \left|\mathbf{W}^T\mathbf{S}_B\mathbf{W}\right| = \left|\mathbf{W}^T \sum_{i=1}^{k} n_i(\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^T\mathbf{W}\right| \ . \tag{1}$$

The columns of **W** could be obtained by solving the eigenvectors of $\mathbf{S}_B$. Specifically, the space spanned by the eigenvectors is the same spanned by $\mathbf{m}_1 - \mathbf{m}, ...\mathbf{m}_i - \mathbf{m}, ..., \mathbf{m}_k - \mathbf{m}$ *(1≤i≤k)*. Therefore, we may directly use this set of directions to construct a new coordinate as well.

M-R Coordinate: For a dataset containing *k* clusters, a coordinate system could be constructed by taking the mass center **m** of the dataset as the origin and $\mathbf{m}_1 - \mathbf{m}, ...\mathbf{m}_i - \mathbf{m}, ..., \mathbf{m}_k - \mathbf{m}$ *(1≤i≤k)* as the directions of coordinate axes, This coordinate system is called M-R coordinate. The coordinate value of point x_j in M-R coordinate is marked as $(\mathbf{x}^c_{j,1}, ..., \mathbf{x}^c_{j,i}, ..., \mathbf{x}^c_{j,k})$, where

$$\mathbf{x}^c_{j,i} = (\mathbf{x}_j - \mathbf{m})^T \frac{\mathbf{m}_i - \mathbf{m}}{\left\|\mathbf{m}_i - \mathbf{m}\right\|_2} \ . \tag{2}$$

Although the choice of axes in M-R coordinate may not be optimal (better axes could be obtained by applying orthogonalization to those directions), there are proper features of these directions that may facilitate the rescaling operation. While the whole set of directions $\mathbf{m}_1 - \mathbf{m}, ...\mathbf{m}_i - \mathbf{m}, ..., \mathbf{m}_k - \mathbf{m}$ *(1≤i≤k)* optimizes the criterion function (1), any single direction $\mathbf{m}_i - \mathbf{m}$ *(1≤i≤k)* in this set is also a discriminative direction to distinguish corresponding cluster C_i and other clusters. Suppose a partition containing two clusters, one of the clusters consists of documents in C_i and the other one consists of documents out of C_i, referred as $C_{i'}$, and the centroid of $C_{i'}$ is marked as $\mathbf{m}_{i'}$. Significantly, the best direction to maximize the between-class scatter of C_i, and $C_{i'}$ is $\widehat{\mathbf{d}_i} = \mathbf{m}_i - \mathbf{m}_{i'}$. And this direction is identical with $\mathbf{m}_i - \mathbf{m}$ because

$$\widehat{\mathbf{d}_i} = \mathbf{m}_i - \mathbf{m}_{i'} = \mathbf{m}_i - \frac{1}{n - n_i}\sum_{\mathbf{x}\notin C_i}\mathbf{x} = \mathbf{m}_i - \frac{n\mathbf{m} - n_i\mathbf{m}_i}{n - n_i} = \frac{n}{n - n_i}\left(\mathbf{m}_i - \mathbf{m}\right).$$

Therefore, for every cluster in the dataset, there is a corresponding axis which gives a discriminative direction to distinguish the current cluster and other clusters by maximizing the between-class scatter. The benefit is that the rescaling operation on this axis could be applied according to the distribution statistics of the current cluster.

We apply the rescaling operation by designing rescaling functions on all dimensions according to the distribution statistics of data. Assume the rescaling functions of different axes in M-R coordinate are $R_1(\bullet), ..., R_i(\bullet), .., R_k(\bullet)$ *(1≤i≤k)*, where $R_i(\bullet)$ could be either linear functions or nonlinear functions. Especially, if the adopted functions are linear functions, i.e., $R_i(x_{\bullet,i}) = \xi_i x_{\bullet,i} + \ell_i$ *(1≤i≤k)*, and we directly introduce them into traditional distance measure, more simplified form of the rescaling operation could be obtained.

M-R Distance: If the adopted rescaling functions on all dimensions are linear functions i.e., $R_i(x_{\bullet,i}) = \xi_i x_{\bullet,i} + \ell_i$ *(1≤i≤k)*, we may directly introduce them into traditional Euclidian distance to form a new distance measure, referred as M-R distance:

$$d_{M-R}(\mathbf{x}_i,\mathbf{x}_j)=\sqrt{\sum_{t=1}^{k}\left(R_t(\mathbf{x}_{i,t}^c)-R_t(\mathbf{x}_{j,t}^c)\right)^2}=\sqrt{\sum_{t=1}^{k}\left(\xi_t(\mathbf{x}_{i,t}^c-\mathbf{x}_{j,t}^c)\right)^2}\ . \qquad (3)$$

where ξ_i *(1≤i≤k)* could be regarded as the rescaling coefficients of different axes. The scales of axes are expanded or shrunken according to the rescaling coefficients such that the coordinate values on different axes are more comparable.

Considering that the rescaling operations are applied to regularize the data distribution, an intuitive but effective solution is to choose a statistic that reflects the distribution characteristics on the corresponding directions as the rescaling coefficient. Since the standard deviation is a parameter that reflects how a distribution spreads out, it is a reasonable choice for the rescaling coefficient. Thus, we can calculate the standard deviation of a cluster's projection on the corresponding axis, and take the reciprocal value of the standard deviation, $\xi_i=1/\sigma_i$, as the axis' rescaling coefficient. Therefore, Formula (3) could be represented as:

$$d_{M-R}(\mathbf{x}_i,\mathbf{x}_j)=\sqrt{\sum_{t=1}^{k}(\mathbf{x}_{i,t}^c-\mathbf{x}_{j,t}^c)^2/\sigma_t^2}\ . \qquad (4)$$

where

$$\sigma_t=\sqrt{\frac{1}{n_t}\sum_{\mathbf{x}\in C_t}\left(\mathbf{x}_{j,i}^c-\|\mathbf{m}_t-\mathbf{m}\|_2\right)^2}\ . \qquad (5)$$

By using $\xi_i=1/\sigma_i$ as the rescaling coefficient, the scales of clusters on the corresponding directions are regularized according to their standard deviations. As a result, the scales of different axes are more comparable and the distance measure is more reasonable.

From another perspective, the rescaling operation could be regarded as a kind of space transformation. The feature space is transformed according to the rescaling coefficient, such that under the new scale the distribution of the dataset fits the model assumptions of the algorithm better. It is worth noting that the choice of $\xi_i=1/\sigma_i$ is only a special case of all possible rescaling functions. Any kinds of dedicated and reasonable rescaling function could be constructed according to the real distribution of datasets to bring better clustering results.

3.3 The Framework

The remaining question is that how we can obtain information of different clusters. Our solution is to execute the given clustering algorithm to generate a rough partition that reflects the intrinsic clusters to some extent. With this rough partition obtained, M-R coordinate could be constructed and rescaling coefficients could be calculated via statistical analysis. With more reasonable distance measure in the M-R coordinate, all documents could be re-clustered under the given clustering algorithm. The new result will be more appropriate than the initial rough partition. As a result, we may conduct the M-R framework iteratively along with the clustering algorithm to constantly improve the clustering performance. And the optimization procedure is quite similar to that of the co-clustering algorithm [16]. By taking iteration strategy, our framework and the clustering partition will be improved simultaneously until the final result is obtained.

For any clustering algorithm, the M-R framework works as bellow:

```
Step 1. Start up the given clustering algorithm to generate a rough partition;
Step 2. Construct/reconstruct the M-R coordinate according to the current partition.
         Calculate/recalculate the rescaling coefficients of different axes in the M-R
        coordinate;
Step 3. Execute the given clustering algorithm in the M-R coordinate with M-R distance to
        generate a new partition;
Step 4. If the stop criterion is achieved, then stop the iteration. Otherwise, go to Step 2.
```

Fig. 2. Applying M-R framework on a given clustering algorithm

4 M-R K-means

To give a direct example, we apply our M-R framework on the traditional k-means to form the M-R k-means algorithm. For a dataset with *n* documents, the M-R k-means clustering algorithm works as follows.

```
1     Run r iterations of k-means to generate a rough partition of k rough clusters;
2     Re-calculate the centroids of all clusters and Re-construct the M-R coordinate;
3     For i=1 to n do:
      3. 1    Calculate coordinate value of $\mathbf{x}_i$ in the new M-R coordinate;
      3.2     Find the nearest cluster for $\mathbf{x}_i$ in the M-R coordinate and assign $\mathbf{x}_i$ to it.
4.    Repeat steps 2 and 3 until no documents shift to other clusters or maximum iteration
      time is achieved.
```

Fig. 3. M-R K-means algorithm

The parameter *r* that controls the initial k-means iteration time is a small integer to guarantee fast generation of the rough partition. Usually, *r=2* or *3*.

Obviously, the time complexity of generating the rough partition of k-means is *O(krn)*, where *k* is the cluster number, *r* is the iteration time and *n* is the total document account. In each iteration of M-R k-means, there are mainly three kinds of operations: adjusting the M-R coordinate, calculating the coordinate values of every document, and calculating the document-to-cluster distances while finding the nearest cluster. The time complexity of adjusting the M-R coordinate is *O(n)* which is mainly induced by updating centroids of all clusters. When calculating the coordinate values of all documents, *k* dot products per document is required according to Formula (2), so the time complexity is *O(kn)*. With new coordinate values calculated, the dimensionality is reduced to *k*, i.e. the number of clusters. Comparing with the operations in the original space with tens of thousands of dimensions, the complexity of the document-to-cluster distance computation in the new coordinate could be omitted. Therefore, the time complexity of each iteration is *O(n)+O(kn)=O(kn)*. Assume that the total iteration time of M-R k-means is *T* (including the *r* iterations while generating the rough partition), the time complexity of M-R k-means is *O(knT)*, which remains the same as k-means.

5 Experiments

In this section, we describe the datasets and evaluation measures used in our experiments, analyze the model misfit problem on the real datasets and eventually conduct experiments to prove the effectiveness of our M-R framework.

5.1 The Datasets

In our experiments, we use two corpora: RCV1-v2 [17] and 20Newsgroup [18].

RCV1-v2. The RCV1 dataset contains a corpus of more than 800,000 newswire stories in 103 classes from Reuters. From RCV1 we randomly select documents to construct a series of datasets (R1, R2, R3, R4 and R5) with cluster numbers vary from 7 to 15. And the document numbers vary from 1932 to 3330.

20NewsGroup. The 20Newsgroup (20NG) contains approximately 20,000 articles evenly divided into 20 Usenet newsgroups. From 20NG, we construct another series of datasets (N1, N2, N3, N4 and N5) by randomly select classes. The cluster numbers of these datasets vary from 6 to 15, and document numbers vary from 2112 to 3406.

The overview of the 10 datasets is illustrated in Table 1. Both RCV1 series and 20NG series are designed to keep a large range on cluster numbers and document numbers. The reason is that we want to give a global view of performance comparison on datasets of different sizes. Stop words are removed, and simple feature selection is applied, e.g., words appear in less than three documents or more than 80% of the documents are automatically removed. Finally, the normalized VSM vector of every document is calculated.

Table 1. Overview of the datasets (corpus type: R- RCV1 series, N-20NG series)

Dataset	R1	R2	R3	R4	R5	N1	N2	N3	N4	N5
#Classes	7	9	11	13	15	6	9	11	13	15
#Documents	1932	2482	2932	3220	3330	2112	3168	3248	3398	3406
Avg. class size	276	257.8	266.5	247.7	222	352	352	464	377.6	227.1

5.2 Evaluation Measures

For clustering, there are many different quality measures, among which the most commonly used ones are F-measure and entropy.

F-measure combines the precision and recall metrics from information retrieval. For a given class in the dataset, the F-measure is determined by the most similar cluster in the result. F-measures of all classes, weighted by the size of each class, are finally averaged to form the total F-measure.

Entropy is a measure that analyzes the homogeneity of all clusters (with the caveat that the best entropy is obtained when each cluster contains exactly one data point).

The total entropy for a set of clusters is calculated as the sum of the entropy of each cluster weighted by the size of each cluster.

Both F-measure and entropy are used to evaluate our experimental results.

5.3 Model Misfit Problem Verification

As presented before, the M-R framework is proposed since the data distribution often misfit the model assumptions of the clustering algorithms. Here we conduct experiments to show this problem on the real datasets. Specifically, standard deviation is used to illustrate the irregular data distribution of clusters on the directions of corresponding axes defined in the M-R coordinate. Therefore, standard deviation is also used to illustrate the model misfit problem in our experiment.

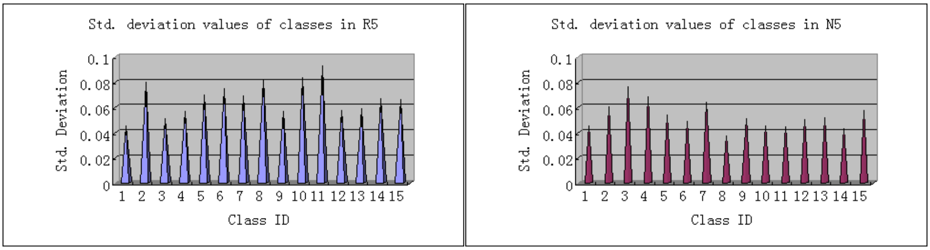

Fig. 4. Std. deviation values of classes in R5 and N5

As an example, we choose the datasets with most clusters and documents (R5 and N5) to verify the model misfit problem. For both datasets, we first construct the M-R coordinate according to the document labels and then calculate the standard deviations of clusters' projections on corresponding axes. Those standard deviations are plotted in the form of column sections in Figure 4. As shown in this figure, for dataset R5, the standard deviation values vary from 0.045 (class ID=1) to 0.092 (class ID=11). And for N5, the values vary from 0.038 (class ID=8) to 0.077 (class ID=3). The results clearly show the differences of standard deviation values, which reflect the misfit between data distribution and algorithm model. Therefore, it is improper to run clustering algorithms with original model assumptions while ignoring the irregular nature of real data distributions. It is better to combine clustering algorithms with M-R framework which is capable of normalizing the irregular data.

5.4 Performance Improvement

In this section, we conduct experiments to verify the performance of M-R framework. Three clustering algorithms are included in our experiments for performance comparison. They are k-means, M-R k-means and spectral clustering with normalized cut (Ncut) [11]. K-means and M-R k-means are compared to verify the performance improvement of M-R framework. Ncut is selected since it is one of the most successful clustering algorithms proposed by researchers in recent years. We can demonstrate the effectiveness of our M-R framework through the comparison between our approach and the state-of-the-art clustering method.

The k-means and M-R k-means in our experiments are implemented with C++ language while the Ncut code is obtained from Spectral Clustering Toolbox provided by University of Washington[1]. For all algorithms in our experiments, i.e. k-means, Ncut and M-R k-means, the algorithm results are affected by the selection of initial points. To avoid the influence of initial points, we run 10 times of all algorithms. For every run, k-means, Ncut and M-R k-means are launched with the same set of randomly selected initial points. At last, we average the F-measure and entropy scores of 10 runs for comparison. For experiments of every dataset, the cluster number of all algorithms is set the same with the category numbers in the dataset. We use Euclidean distance in k-means for comparison with our M-R distance. The k-means algorithm stops when the difference value of the sum-of-square criterion function in the recent two iterations is less than 0.001, or maximum iteration time (which is set to 20) is achieved. For Ncut, the convergence criterion is set to the same value with K-means. For M-R k-means we run 3 iterations of k-means to generate a rough partition to setup the M-R framework. M-R k-means iterates until no documents shift to other clusters or maximum iteration time (which is also set to 20) is achieved.

The F-measure and entropy results of the experiments are shown in Table 2 and Table 3. Those results are also drawn in Figure 5 and Figure 6 to give an intuitive view of performance comparison. From the results we observe that:

1. Comparing with k-means, M-R k-means achieves overall improvement in all datasets on both F-measure and entropy scores. It proves the effectiveness of M-R framework.

2. As for comparison with Ncut, the result is quite interesting. When cluster number is small, the performance of Ncut is superior to M-R k-means. However, with the growth of cluster numbers, M-R k-means is capable of getting comparable results with Ncut. In detail, the results of M-R k-means are comparable to or slightly weaker than Ncut on datasets of RCV1 series, and are comparable to or even better than Ncut on datasets of 20NG series when the cluster number is large. Here we give some brief explanation on the result. On one hand, Ncut constructs the new feature space with the best *k* eigenvectors. Therefore, when *k* is small, the quality of new feature space would be very good. However, when *k* increases, the quality of new feature space would decrease. On the other hand, for our M-R k-means, the improvement comparing with k-means is quite significant and stable. As a result, for large datasets, M-R k-means would generate comparable results with Ncut.

We omitted the execution time comparison in this section because we have proved that the time complexity of M-R k-means remains the same as k-means in Section 4. That is to say, M-R k-means is capable of executing as fast as k-means but generating better result.

Table 2. Average F-measure of the clustering result for all datasets

Dataset	**R1**	**R2**	**R3**	**R4**	**R5**	**N1**	**N2**	**N3**	**N4**	**N5**
k-means	0.568	0.534	0.529	0.483	0.468	0.702	0.672	0.647	0.605	0.632
Ncut	**0.698**	0.567	**0.578**	**0.550**	0.521	**0.811**	0.691	0.662	0.652	0.701
M-R k-means	0.630	**0.577**	0.562	0.528	**0.535**	0.792	**0.737**	**0.699**	**0.673**	**0.713**

[1] http://www.cs.washington.edu/homes/sagarwal/code.html

Table 3. Average entropy of the clustering result for all datasets

Dataset	R1	R2	R3	R4	R5	N1	N2	N3	N4	N5
k-means	0.545	0.548	0.513	0.538	0.514	0.424	0.399	0.388	0.408	0.362
Ncut	**0.403**	0.472	**0.446**	0.474	0.488	0.324	0.407	0.393	0.382	0.317
M-R k-means	0.468	**0.463**	0.474	**0.470**	**0.476**	**0.313**	**0.319**	**0.318**	**0.311**	**0.269**

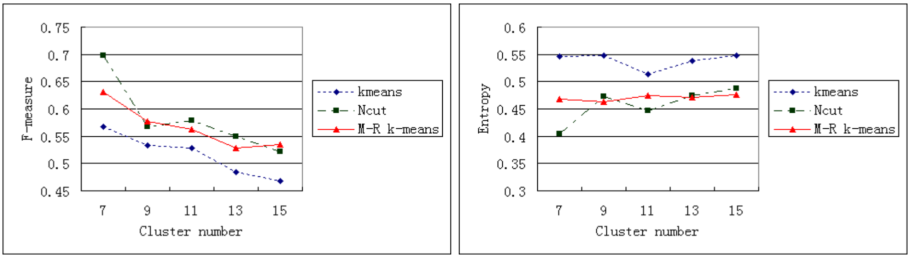

Fig. 5. F-measure and entropy scores of experiments on datasets of RCV1 series

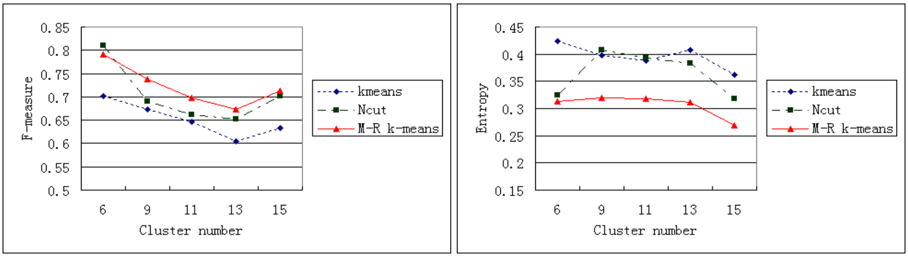

Fig. 6. F-measure and entropy scores of experiments on datasets of 20NG series

6 Conclusions and Future Work

In this paper, we propose a novel clustering framework based on adaptive space mapping and rescaling, referred as the M-R framework. The M-R framework maps all documents into a low dimensional coordinate which is constructed with respect to the cluster centers. In this way, the distribution statistics of each cluster could be analyzed on the corresponding dimension. A rescaling operation is then conducted with respect to such statistics to regularize the data distribution based on the model assumptions. By conducting the framework iteratively along with the clustering algorithm, we are able to constantly improve the clustering performance. It is worth noting that M-R framework does not introduce any time consuming operation to the original algorithm. Experiments on well known dataset show that by combining the M-R framework, traditional algorithm like k-means is capable of achieving comparable clustering performance with respect to the state-of-the-art methods.

The distribution regularization idea introduced by M-R framework is novel and interesting, and is applicable to more text mining areas. In future work, we will focus on theoretical study of our M-R framework and try to apply the framework to supervised learning area like text classification.

References

1. Dumais, S.T.: LSI Meets TREC: A Status Report. In: Harman, D. (ed.) The First Text REtrieval Conference (TREC1), pp. 137–152. National Institute of Standards and Technology Special Publication 500-207 (1993)
2. Van Rijsbergen, C.J.: Information Retrieval, 2nd edn. Buttersworth, London (1989)
3. Liu, X., Croft, W.B.: Cluster-Based Retrieval Using Language Models. In: Proc. of SIGIR 2004, pp. 186–193 (2004)
4. Cutting, D.R., Karger, D.R., Pedersen, J.O., Tukey, J.W.: Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections. In: SIGIR 1992, pp. 318–329 (1992)
5. Zamir, O., Etzioni, O., Madani, O., Karp, R.M.: Fast and Intuitive Clustering of Web Documents. In: KDD 1997, pp. 287–290 (1997)
6. Han, J., Kamber, M.: Data Mining: Concepts and Techniques, 2nd edn. Morgan Kaufmann Publishes, San Francisco (2006)
7. Wu, H., Phang, T.H., Liu, B., Li, X.: A Refinement Approach to Handling Model Misfit in Text Categorization. In: SIGKDD, pp. 207–216 (2002)
8. Tan, S., Cheng, X., Ghanem, M.M., Wang, B., Xu, H.: A Novel Refinement Approach for Text Categorization. In: Proc. of the 14th ACM CIKM 2005, pp. 469–476 (2005)
9. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press, Cambridge (2004)
10. Ng, A., Jordan, M., Weiss, Y.: On Spectral Clustering: Analysis and an Algorithm. In: Dietterich, T., Becker, S., Ghahramani, Z. (eds.) Advances in Neural Information Processing Systems, vol. 14. MIT Press, Cambridge (2002)
11. Shi, J., Malik, J.: Normalized Cuts and Image Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(8), 888–905 (2000)
12. Chan, P.K., Schlag, D.F., Zien, J.Y.: Spectral K-way Ratio-Cut Partitioning and Clustering. IEEE Trans. Computer-Aided Design 13, 1088–1096 (1994)
13. Ding, C., He, X., Zha, H., Gu, M., Simon, H.D.: A Min-Max Cut Algorithm for Graph Partitioning and Data Clustering. In: Proc. of ICDM 2001, pp. 107–114 (2001)
14. Liu, X., Gong, Y.: Document Clustering with Cluster Refinement and Model Selection Capabilities. In: Proc. of SIGIR 2002, pp. 191–198 (2002)
15. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley-Interscience Publishes, Hoboken (2000)
16. Dhillon, I.: Co-clustering Documents and Words using Bipartite Spectral Graph Partitioning (Technical Report). Department of Computer Science, University of Texas at Austin (2001)
17. Lewis, D.D., Yang, Y., Rose, T., Li, F.: RCV1: A New Benchmark Collection for Text Categorization Research. Journal of Machine Learning Research (2004)
18. 20 Newsgroups Data Set, `http://www.ai.mit.edu/people/jrennie/20Newsgroups/`