Refinement of Document Clustering by Using NMF[*]

Hiroyuki Shinnou and Minoru Sasaki

Department of Computer and Information Sciences, Ibaraki University,
4-12-1 Nakanarusawa, Hitachi, Ibaraki JAPAN 316-8511
{shinnou, msasaki}@mx.ibaraki.ac.jp

Abstract. In this paper, we use non-negative matrix factorization (NMF) to refine the document clustering results. NMF is a dimensional reduction method and effective for document clustering, because a term-document matrix is high-dimensional and sparse. The initial matrix of the NMF algorithm is regarded as a clustering result, therefore we can use NMF as a refinement method. First we perform min-max cut (Mcut), which is a powerful spectral clustering method, and then refine the result via NMF. Finally we should obtain an accurate clustering result. However, NMF often fails to improve the given clustering result. To overcome this problem, we use the Mcut object function to stop the iteration of NMF.

Keywords: document clustering, Non-negative Matrix Factorization, spectral clustering, initial matrix

1. Introduction

In this paper, we use non-negative matrix factorization (NMF) to improve the document clustering result generated by a powerful document clustering method. Using this strategy, we can obtain an accurate document clustering result.

Document clustering is a task that divides a given document data set into a number of groups according to document similarity. This is the basic intelligent procedure, and an important factor in text-mining systems, from Berry (2003). Relevant feedback in information retrieval (IR), where retrieved documents are clustered, is a specific application that is actively researched by Hearst *et al.* (1996), Leuski (2001), Zeng *et al.* (2001) and Kummamuru (2004).

NMF is a dimensional reduction method and an effective document clustering method, because a term-document matrix is high-dimensional and sparse, from Xu *et al.* (2003).

Let X to be a $m \times n$ term-document matrix, consisting of m rows (terms) and n columns (documents). If the number of clusters is k, NMF decomposes X to the matrices U and V^t as follows:

$$X = UV^t$$

where U is $m \times k$, V is $n \times k$ and V^t is the transposed matrix of V. The matrix U and V are non-negative. In NMF, each k dimensional column vector in V corresponds to a document. An actual clustering procedure is usually performed using these reduced vectors. However, NMF does not need such a clustering procedure. The reduced vector expresses its cluster by itself, because each column axis of V represents a topic of the cluster. Furthermore, the matrices V and U are

[*] This research was partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research on Priority Areas, “Japanese Corpus”, 19011001, 2007.

obtained by a simple iteration, from Lee (2000), where the initial matrices U_0 and V_0 are updated. Therefore, we can regard NMF as a refinement method for a given clustering result, because the matrix *V* represents a clustering result.

In this paper, we use NMF to improve clustering results. Providing NMF with an accurate document clustering result, we can ensure a more accurate result, because NMF is effective for document clustering. However, NMF often fails to improve the initial clustering result. The main reason for this is that the object function of NMF does not properly represent the goodness of clustering. To overcome this problem, we use another object function. After each iteration of NMF, the current clustering result is evaluated by that object function.

We first need the initial clustering result. To obtain this, we perform min-max cut (Mcut) proposed by Ding *et al.* (2001), which is a spectral clustering method. Mcut is a very powerful clustering method, and we can obtain an accurate clustering result by improving the clustering result generated through Mcut,

In the experiment, we used 19 data set provided via the CLUTO website. Our method improved the clustering result generated by Mcut. In addition, the accuracy of the obtained clustering result was higher than those of NMF, CLUTO and Mcut.

2. Refinement using NMF

2.1. Features of NMF

NMF decomposes the $m \times n$ term-document matrix *X* to the $m \times k$ matrix *U* and the transposed matrix V^t of the $n \times k$ matrix V, from Xu *et al.* (2003), where *k* is the number of clusters:

$$X = UV^t .$$

NMF attempts to find the axes corresponding to the topic of the clusters, and represents the document vector and the term vector as a linear combination of the found axes.

NMF has following three features:

i. *V* and *U* are non-negative.
The element of *V* and *U* refers to the degree of relevance to the topic corresponding to the axis of its element. It is therefore natural to assign a non-negative value to the element. SVD can also reduce dimensions, but negative values appear unlike with NMF.

ii. The matrix *V* represents the clustering result.
The dimensional reduction translates high-dimensional data to lower-dimensional data. Therefore, we usually must perform actual clustering for the reduced data. However, NMF does not require this, because the matrix *V* represents the clustering result. The *i*-th document d_i corresponds to the *i*-th row vector of *V*, that is, $d_i = (v_{i1}, v_{i2}, 4\ , v_{ij})$. The cluster number is obtained from $\arg\max_j v_{ij}$.

iii. *V* and *U* do not need to be an orthogonal matrix.
LSI constructs orthogonal space from document space. On the other hand, in NMF, the axis in the reduced space corresponds to a topic, therefore, these axes do not need to be orthogonal. As a result, NMF attempts to find the axis corresponding to the cluster that has documents containing identical words.

2.2. NMF algorithm

For the given term-document matrix *X*, we can obtain *U* and *V* by the following iteration, shown by Lee (2000).

$$u_{ij} \leftarrow u_{ij} \frac{(XV)_{ij}}{(UV^tV)_{ij}} \qquad \text{(Eq.1)}$$

$$v_{ij} \leftarrow v_{ij} \frac{(X^tU)_{ij}}{(VU^tU)_{ij}} \qquad \text{(Eq.2)}$$

Here, u_{ij}, v_{ij} and $(X)_{ij}$ are the *i*-th row and the *j*-th column element of *U*, *V* and a matrix *X* respectively.

After each iteration, U must be normalized as follows:

$$u_{ij} \leftarrow \frac{u_{ij}}{\sqrt{\sum_i u_{ij}^2}}$$

The iteration stops by the fixed maximum iteration number, or the distance *J* between *X* and UV^t:

$$J = \left\| X - UV^t \right\| \qquad \text{(Eq.3)}$$

Here, J is the decomposition error.

2.3. Clustering result and initial matrices

In general, the initial matrices U_0 and V_0 are constructed using random values. In this paper, we construct the U_0 and V_0 through a clustering result.

In particular, if the cluster number of the *i*-th data is clustered into the *c*-th cluster, the *i*-th row vector of V_0 is constructed as follows:

$$v_{ij} = \begin{cases} 1.0 & (j = c) \\ 0.1 & (j \neq c) \end{cases}$$

Here, U_0 is constructed via XV_0.

2.4. Problem of the object function of NMF

We can use NMF as a refinement method for a clustering result, because the initial matrix of NMF corresponds to a clustering result. However, NMF often fails to improve the given clustering result. This is because the object function of NMF, that is, Eq. 3, does not properly represent the goodness of clustering.

To confirm this problem, we performed NMF using the document data set ``tr45" which is a part of the data set used in Section 5. The initial matrix was constructed using the clustering result obtained by Mcut. Figure 1 shows the results of this experiment. LINE-1 and LINE-2 in Figure 1 show the change in J in each iteration and the change in the clustering accuracy, respectively. From Figure 1, we can confirm that a smaller *J* does not always mean a more accurate clustering.

To overcome this problem, we evaluated the current clustering result using another object function after each iteration of NMF.

Specifically, we used the object function of Mcut. We calculated the value of the object function after each iteration of NMF. If the best value was not improved for three consecutive iterations, we stopped NMF.

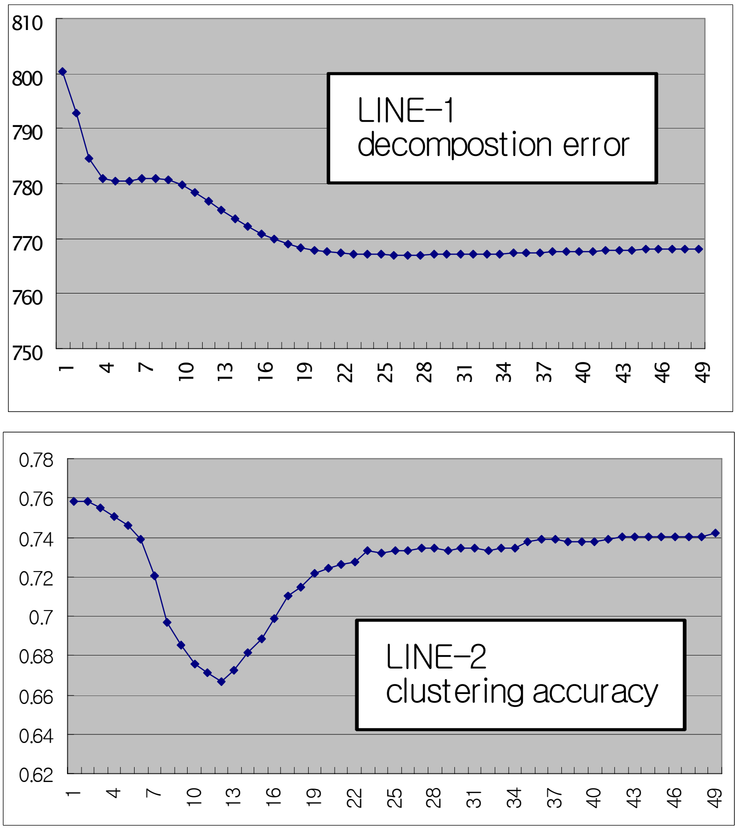

Figure 1: Decomposition error and clustering accuracy

3. Mcut

Next, we needed the initial clustering result. To obtain this, we used Mcut proposed by Ding *et al.* (2001) which is a type of spectral clustering.

In this spectral clustering method, the data set is represented as a graph. Each data point is represented as a vertex in the graph. If the similarity between data A and B is non-zero, the edge between A and B is drawn and the similarity is used as the weight of the edge. From this graph, clustering can be seen to correspond to the segmentation of the graph into a number of subgraphs by cutting the edges. The preferable cutting is such that the sum of the weights of the edges in the subgraph is large and the sum of weights of the cut edges is small. To find the ideal cut, the object function is used. The spectral clustering method finds the desirable cut by using the fact that an optimum solution of the object function corresponds to the solution of an eigenvalue problem. Different object functions are proposed. In this paper, we use the object function of Mcut.

First, we define the similarity *cut(A,B)* between the subgraph *A* and *B* as follows:

$$cut(A,B) = W(A,B).$$

The function *W(A,B)* is the sum of the weights of the edges between *A* and *B*. We define *W(A)* as *W(A,A)*.

The object function of Mcut is the following:

$$Mcut = \frac{cut(A,B)}{W(A)} + \frac{cut(A,B)}{W(B)} \qquad \text{(Eq.4)}$$

The clustering task is to find *A* and *B* to minimize the above equation.

Note that the spectral clustering method divides the data set into two groups. If the number of clusters is larger than two, the above procedure is iterated recursively.

The minimization problem of Eq.4 is equivalent to the problem of finding the *n* dimensional discrete vector *y* to minimize the following equation:

$$J_m = \frac{y^t (D - W) y}{y^t W y} \qquad \text{(Eq.5)}$$

where *W* is the similarity matrix of data, *D* = *diag(We)* and $e = (1,1,4\ \ ,1)^t$. Each element in the vector *y* is *a* or −*b*, where $a = \sqrt{\frac{d_B}{d_A d}}$ $, b = \sqrt{\frac{d_A}{d_B d}}$, $d_X = \sum_{i \in X} (D)_{ii}$ and $d = d_A + d_B$. If the *i*-th element of the vector *y* is *a* (or −*b*), the *i*-th data element belongs to the cluster *A* (or *B*). We can solve Eq.5 by converting the discrete vector *y* to the continuous vector *y*. Finally, we can obtain an approximate solution to Eq.5 by solving the following eigenvalue problem:

$$(I - D^{-1/2} W D^{1/2}) z = \lambda z \qquad \text{(Eq.6)}$$

We obtain the eigenvector *z*, that is, Fielder vector, corresponding to the second minimum eigenvalue by solving the eigenvalue problem represented by Eq.6. We can obtain the solution *y* to Eq.5 from $z = D^{1/2} y$. By the sign of the *i*-th value of *y*, we can judge whether the *i*-th data element belongs to cluster *A* or *B*.

Note that Eq.4 is the object function when the number of clusters is two. The object function used in NMF is the following general object function for *k* clusters $\{G_i\}_{i=1:k}$.

$$Mcut_K = \frac{cut(G_1, \overline{G_1})}{W(G_1)} + \frac{cut(G_2, \overline{G_2})}{W(G_2)} + 4\ \ + \frac{cut(G_k, \overline{G_k})}{W(G_k)} \qquad \text{(Eq.7)}$$

where $\overline{G_k}$ is the complement of G_k . The smaller $Mcut_K$ is, the better it is.

4. Experiment

In the experiment, we used the data set provided via the CLUTO website

http://glaros.dtc.umn.edu/gkhome/cluto/cluto/download.

In total, 24 data sets are available. We used data sets that had less than 5,000 data elements. As a result, we used 19 data sets, shown in Table 1. In each data set, the document vector is not normalized. We normalize them by TF-IDF.

Table 1: Document data sets

Data	# of documents	# of terms	# of non-zero elements	# of classes
cacmcisi	4,663	41,681	83,181	2
cranmed	2,431	41,681	140,658	2
fbis	2,463	2,000	393,386	17
hitech	2,301	126,373	346,881	6
k1a	2,340	21,839	349,792	20
k1b	2,340	21,839	349,792	6
la1	3,204	31,472	484,024	6
la2	3,075	31,472	455,383	6

mm	2,521	126,373	490,062	2
re0	1,504	2,886	77,808	13
re1	1,657	3,758	87,328	25
reviews	4,069	126,373	781,635	5
tr11	414	6,429	116,613	9
tr12	313	5,804	85,640	8
tr23	204	5,832	78,609	6
tr31	927	10,128	248,903	7
tr41	878	7,454	171,509	10
tr45	690	8,261	193,605	10
wap	1,560	6,460	220,482	20

Table 2 shows the result. NMF-rfn in the table refers to our method. That is, we obtained the initial clustering results by Mcut and then improved it by performing NMF. The NMF-rfn column in Table 2 shows the ratio of values of Eq.7 obtained using our method to those obtained using Mcut. As shown in Table 2, the value of Eq.7 of our method is less than (or equal to) Mcut absolutely. This means that our method absolutely improves the clustering results considering Eq.7.

Table 2: Comparison of the object function value

Data	NMF-rfn
cacmcisi	1.0000
cranmed	1.0000
Fbis	0.9350
Hitech	0.9345
k1a	0.6340
k1b	0.9630
la1	1.0000
la2	0.9862
Mm	0.9979
re0	1.0000
re1	0.9974
reviews	0.6503
tr11	0.8971
tr12	1.0000
tr23	0.9806
tr31	0.9728
tr41	0.9409
tr45	0.8242
Wap	0.7679
Average	0.9201

Next, we checked the accuracy of our method. Table 3 and Figure 2 show the results. The column of NMF, CLUTO[1] and Mcut in Table 3 shows the accuracy of NMF, CLUTO and Mcut respectively. And the column of NMF-ref is the accuracy of our method.

Table 3: Accuracy of each method

Data	NMF	CLUTO	Mcut	NMF-rfn
cacmcisi	0.5788	0.6054	0.6858	0.6858
cranmed	0.5825	0.9975	0.9930	0.9930
fbis	0.4125	0.4921	0.5278	0.4941
hitech	0.4633	0.5228	0.3859	0.5059
k1a	0.4107	0.4799	0.4658	0.5684
k1b	0.6389	0.6081	0.5205	0.5342
la1	0.6798	0.7147	0.6879	0.6879
la2	0.5873	0.6582	0.7028	0.6924
mm	0.5470	0.5331	0.9583	0.9556
re0	0.3710	0.3198	0.3670	0.3670
re1	0.3826	0.4146	0.4490	0.4599
reviews	0.7196	0.6316	0.6776	0.6424
tr11	0.5556	0.6812	0.6546	0.7295
tr12	0.6422	0.6869	0.7764	0.7764
tr23	0.3971	0.4559	0.4363	0.4363
tr31	0.5696	0.5674	0.7228	0.6624
tr41	0.5239	0.6412	0.5661	0.6014
tr45	0.6347	0.5986	0.7580	0.7101
wap	0.4686	0.4487	0.4109	0.5096
Average	0.5350	0.5821	0.6182	0.6322

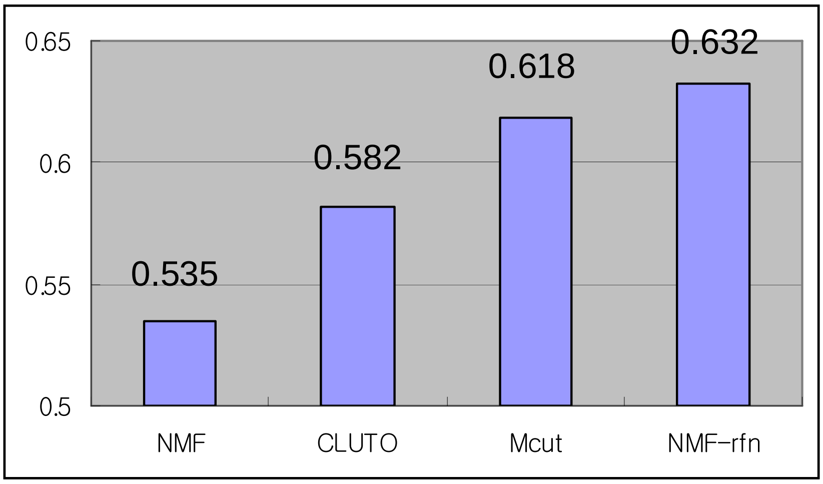

Figure 2: Average accuracy of each method

[1] CLUTO is a very powerful clustering tool. We can get from the following website. http://glaros.dtc.umn.edu/gkhome/views/cluto (version 2.1.2a)

Clustering accuracy is the most rigorous evaluation of the clustering result. However, accuracy is difficult to measure. First of all, all data must be labeled. Fortunately the data sets used satisfy this condition. Next, we must map each obtained cluster to the cluster label. This mapping is usually difficult. In this paper, we assigned the label to the cluster to assure the accuracy is high, by using dynamic programming. As a result, we obtain accurate clustering.

The measure of similarity and the clustering method of CLUTO must also be examined We can select these via the optional parameter of CLUTO. In our experiments, we conducted CLUTO without any optional parameters, that is, by using the default setting. In this case, CLUTO uses the cosine similarity measure and the k-way clustering method, which takes a top-down approach to divide data into two partitions and iterates this division until *k* partitions are obtained. In general, the k-way clustering method is more powerful than k-means for document clustering.

There were six data sets for which the accuracy was degraded by performing NMF after Mcut. But in seven data sets, the accuracy was improved by NMF. In the remaining six data sets, the accuracy was not changed. Figure 2 shows that the average accuracies of CLUTO, Mcut and NMF-rfn were 58.21%, 61.82% and 63.22% respectively. That is, our method showed the best performance.

5. Discussions

5.1. Search for the optimum solution

The object function value of the end clustering result is never degraded from the value in Mcut. However, as shown in Table 2, there are some data sets for which the clustering accuracy of NMF-rfn is worse than that of Mcut.

This is because the object function used does not refer to the goodness of clustering in a precise sense. All object functions suffer from the same problem. Especially, the object function *J* in Eq. 3 is not so good. In fact, we confirmed that the Mcut object function is better than *J* in Eq.3 for NMF from another experiment.

The clustering task has two parts: one is the object function, and the other is the search method for the optimum solution to the object function. Mcut-rfn uses Eq.7 as the object function and combines the search methods of Mcut and NMF as its search method.

Recent theoretical analysis shows the equivalence between spectral clustering and other clustering methods. For example, Dhillon *et al.* (2005) show that a search for an optimum solution via spectral clustering can be performed using the weighted kernel k-means. Additionally, Ding *et al.* (2005) show the equivalence between spectral clustering and NMF. By using these techniques, a search for an optimum solution may be constructed in a consistent manner, unlike with Mcut-rfn.

However, such a consistent manner cannot avoid falling into a local optimum solution. It is therefore helpful to add a mechanism to jump out from a local optimum solution. Our hybrid approach is an example of such a method.

The ``*local search*'' proposed by Dhillon *et al.* (2002) is relevant to our approach. This method first obtains a solution by k-means and then improves it by the ``*first variation*'' and iterates these two steps alternately. Mcut-rfn first obtains a solution by Mcut and then improves it by NMF, but it does not iterate them, because the input of Mcut does not need to be a clustering solution. Using the weighted kernel k-means, we can take the ``*ping-pong*'' strategy like the local search.

5.2. Initial matrices and accuracy of NMF

In NMF, clustering accuracy depends on the initial matrices. This is because the local optimum solution obtained by NMF varies according to the initial value. Therefore, deciding what initial matrices should be used is a difficult problem, from Wild *et al.* (2004).

Regarding the object function, initial accuracy must be improved. Thus, we took the approach to set the value that had a high accuracy as the initial value. However, even if NMF starts from initial values that have low accuracy, NMF can still obtain highly accurate results. For example, for the data set ``k1a" and ``tr11" in our experiments, CLUTO was better than Mcut. Using the result of CLUTO as the initial value, accuracy was not improved by NMF. On the other hand, in the case of Mcut, accuracy was improved by NMF, and the final accuracy was better than that of CLUTO.

Finally, clustering is an NP-hard combinatorial optimization problem after the object function is fixed. It is impossible to find the optimal initial value. Thus, the clustering algorithm must take an approach that improves the solution gradually. Under such a situation, our approach to set a feasible solution to the initial value is practical.

5.3. Future works for document clustering

The clustering task is a purely engineered problem once data is translated into vectors. To get more accurate clustering, we should actively use knowledge on data at the pre-translated stage. In the case of document clustering, we should remember that the data is a document. It may be important to ensure that meta-information such as the publication place, author, aim of clustering is incorporated into the clustering process or vector-translation process.

Clustering is unsupervised learning. The effective way to raise accuracy is therefore to assign supervised labels to data. Recently, semi-supervised clustering using user-interaction has been actively researched by Basu *et al.* (2002), Bilenko *et al.* (2004) and Xing *et al.* (2003). This semi-supervised clustering using meta-information shows promise.

6. Conclusion

In this paper, we have shown that NMF can be used to improve clustering result. For practical use, we used another object function, and we evaluated the current clustering result using that object function after each iteration of NMF. By performing Mcut to obtain the initial clustering result, we can obtain an accurate clustering result. In the experiment, we used 19 data set provided via the CLUTO website. Our method improved the clustering result obtained by Mcut. In addition, the accuracy of the obtained clustering result was higher than those of NMF, CLUTO and Mcut. In future, we will research semi-supervised clustering using meta-information.

References

Basu, S., A. Banerjee, and R. J. Mooney. 2002. Semi-supervised Clustering by Seeding. *Proceedings of ICML-2002,* pp.19-26.

Berry, M. W. 2003. *Survey of Text Mining: Clustering, Classification, and Retrieval*. Springer.

Bilenko, M., S. Basu and R. J. Mooney. 2004. Integrating Constraints and Metric Learning in Semi-Supervised Clustering. *Proceedings of ICML-2004*, pp.81-88.

Dhillon, I. S., Y. Guan and J. Kogan. 2002. Iterative Clustering of High Dimentional Text Data Augmented by Local Search. *The 2002 IEEE International Conference on Data Mining*, 131-138.

Dhillon, I. S., Y. Guan and B. Kulis. 2005. A Unified View of Kernel k-means, Spectral

Clustering and Graph Cuts. *The University of Texas at Austin, Department of Computer Sciences. Technical Report TR-04-25*.

Ding, C., X. He and H. D. Simon. 2005. On the Equivalence of Nonnegative Matrix Factorization and Spectral Clustering. *Proceedings of SDM 2005*.

Ding, C., X. He, H. Zha, M. Gu and H. Simon. 2001. Spectral Min-max Cut for Graph Partitioning and Data Clustering. *Lawrence Berkeley National Lab. Tech. report 47848.*

Hearst, M. A. and J. O. Pedersen. 1996. Reexamining the Cluster Hypothesis: Scatter/gather on Retrieval Results. *Proceedings of SIGIR-96*, pp.76-84.

Kummamuru, K., R. Lotlikar, S. Roy, K. Singal and R. Krishnapuram. 2004. A Hierarchical Monothetic Document Clustering Algorithm for Summarization and Browsing Search Results. *Proceedings of WWW-04*, pp.658-665.

Lee, D. D. and H. S. Seung. 2000. Algorithms for Non-negative Matrix Factorization. *Proceedings of NIPS-2000*, pp.556-562.

Leuski, A. 2001. Evaluating Document Clustering for Interactive Information Retrieval. *Proceedings of CIKM-01*, pp.33-40.

Wild, S., J. Curry and A. Dougherty. 2004. Improving Non-negative Matrix Factorizations through Structured Initialization. *Pattern Recognition*, Vol.37, No.11, 2217-2232.

Xing, E. P., A. Y. Ng, M. I. Jordan and S. Russell. 2003. Distance Metric Learning, with Application to Clustering with Side-information. A*dvances in Neural Information Processing Systems* 15, 505-512.

Xu, Wei., X. Liu and Y. Gong. 2003. Document Clustering Based on Non-negative Matrix Factorization. *Proceedings of SIGIR-03*, pp.267-273.

Zeng, H.-J., Q.-C. He, Z. Chen, W.-Y. Ma and J. Ma. 2001. Learning to Cluster Web Search Results. *Proceedings of SIGIR-04*, pp.33-40