Double Clustering in Latent Semantic Indexing

Kristóf Csorba, István Vajk

Department of Automation and Applied Informatics
Budapest University of Technology and Economics
Goldmann Gy. tér 3, H-1111 Budapest, Hungary
kristof@aut.bme.hu, vajk@aut.bme.hu

Abstract: Document clustering is a widely researched area of information retrieval. The large amount of documents which must be handled needs automatic organizing. A popular approach to clustering documents and messages is the vector space model, which represents texts with feature vectors, usually generated from the set of terms contained in the message. The clustering based on the document-term frequency matrixes suffers from noise caused by the frequent use of different words with similar meanings. These semantic relations (like synonyms) need to be handled. The method described in this paper uses Singular Value Decomposition (SVD) technique combined with double clustering to reduce the dimension of the vector space. In this way the clustering is performed in a space with fewer dimensions and reduced noise.

Keywords: latent semantic indexing, double clustering

1 Introduction

Document clustering is a procedure to separate documents according to certain criteria, for instance documents of different topics. Document clustering is expected to recognize topics and to identify, to which one a given document belongs. Topics are often treated to be overlapping, which means multiple topics can be returned to a document by providing some proximity measure.

Document clustering based on unsupervised learning needs to measure the document similarity. A common approach is based on the document-term frequency matrix. (A kernel-based description of the area can be found in [1].) This contains the frequency of each term in each document. The most important problem of this approach is caused by terms with similar meaning: these are used in different documents for the same concept. As the terms are different, this leads to different column vectors for documents with similar topic in the document-term frequency matrix. As the inner product of these document (column) vectors is low, clustering methods based on inner product similarity (like cosine distance) cannot recognize the shared concepts. Since this situation is very frequent, a naive

clustering method based only on inner product of document vectors will treat hardly every document to be very far from hardly every other document. Thus useful clustering cannot be achieved in this way. This means that the later described k-means clustering – which is based on the document vectors – cannot be applied successfully.

If $\boldsymbol{d_1}$ and $\boldsymbol{d_2}$ are two column vectors in the $\boldsymbol{X}$ document-term matrix, the inner product $< \boldsymbol{d_1}; \boldsymbol{d_2} >$ is low if the corresponding two documents share only few terms. If the documents use other words (synonyms) for the same concepts, the inner product stays low, although the topics are near. This is the reason why the naive clustering methods fail. A frequently used solution is to capture the (latent) semantic relationships between the terms. The semantic similarity has many forms and many ways to capture. A popular approach assumes that terms occurring often together in the documents are related to similar topics with high probability. Singular value decomposition (SVD) has been shown to be capable of finding such similarities [2]. Although after SVD the terms and the documents can be immediately clustered in the same feature space, we used double clustering [3] in this paper to achieve a stronger noise reduction. In the first step singular value decomposition is used to cluster the terms and in the second step documents are clustered based only on the term-clusters which their terms belong to. That means that the document clustering is performed in a space with strong reduced dimensionality providing an effective noise reduction before the clustering step.

Figure 1 presents an overview of the process in a more formal way. The n-by-m document–term frequency matrix is the starting point of the procedure which is generated through the parsing of the documents. SVD is performed to retrieve a term representation in a reduced ($k < m$) space. K-means clustering is applied to the terms to generate term-clusters in this feature space as described in Section 2. After term-clustering is applied to the document-term frequency matrix $\boldsymbol{X}$ to calculate the document-term-cluster frequency matrix $\boldsymbol{Y}$, the second k-means is to cluster the documents in the space of the term-clusters. Details about this procedure are provided in Section 3.

The most important matrixes, sets and scalars used in this paper are the following:

- $\boldsymbol{X}$: n-by-m document–term frequency matrix
- $\boldsymbol{Y}$: t-by-m document–term-cluster frequency matrix
- $\boldsymbol{Z}$: t-by-d document-cluster–term-cluster frequency matrix
- $\boldsymbol{T}$: matrix of term-coordinates in the reduced space, retrieved from the results of SVD.
- C: set of term–clusters, C_i represents the set of terms in the i-th cluster.
- n: number of terms occurring in at least one of the documents.
- m: number of documents.

- k: number of dimensions after the dimension reduction through SVD.
- t: number of term–clusters.
- d: number of document–clusters.

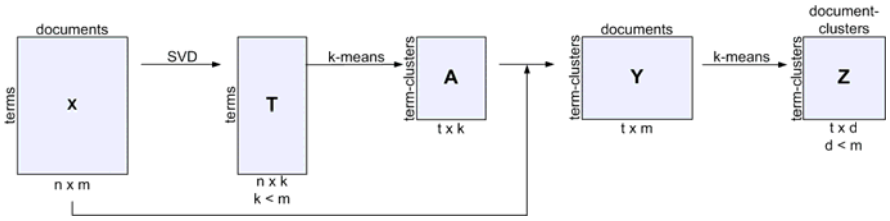

Figure 1
Overview of double clustering

2 Creating Term Clusters

Singular value decomposition is a matrix-analytical method to search for a space, where a given linear transformation (usually given by a rectangular matrix) is a scaling along the base vectors. More precisely if ***X*** is an n-by-m matrix, SVD has a result in the form

$$X = USV^T \tag{1}$$

where ***S*** is the r-by-r diagonal matrix of the singular values of ***X*** (The rank of ***X*** is *r*) and ***U*** and ***V*** have orthogonal columns and contain the left and right singular vectors.

If the row vectors of ***U*** and ***V*** are taken geometrically as coordinates in the *m* dimensional space, the terms and the documents became points in a vector space (in the ”feature space”). The diagonal elements in ***S*** serve as a scaling of the axes in this space reflecting the contribution of the dimensions in the overall similarity structure.

As the singular values in the diagonal of ***S*** are provided in descending order, one can easily select the most significant dimensions by selecting the first *k* columns of ***U*** and ***V***. The dimensionality reduction from *m* to *k* leads to noise filtering (by eliminating less significant dimensions) which makes clustering of terms and documents more effective.

The term clustering in this approach is based on the term coordinates retrieved from the row vectors of ***U*** after a reduction to *k* dimensions. This means

$$T \leftarrow US' \tag{2}$$

where ***S'*** contains zeros except the first k singular values in the diagonal retrieved from ***S***.

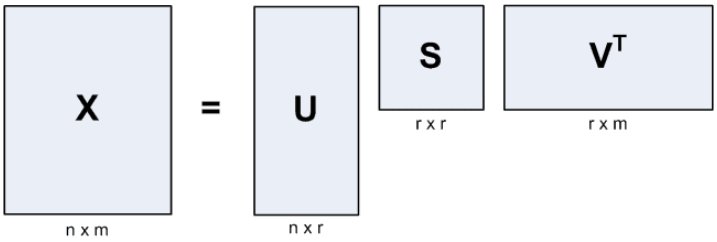

Figure 2
Singular value decomposition

In the feature space, the terms with similar occurrence behaviors are located near each other while words related to different topics (which means that they were used in different documents) are further in the sense of cosine distance[1]. Proximity based on similar occurrence behavior is achieved with two facts:

- The dimensionality reduction eliminates small differences in the occurrence statistics.
- We used a document–term frequency matrix, which contains the normalized frequency values. Normalization for the documents is important for avoiding bias produced by unequal length of documents. (This would cause noise for the SVD and make recognition of significant directions in the feature space harder.)

By applying an appropriate clustering algorithm to the term points in the feature space, clusters containing words belonging to similar concepts can be created. This needs the proper selection of the clustering algorithm and the *k* number of dimensions preserved from the result of SVD. In the current approach the clustering algorithm k-means was selected to create the term clusters. K-means is an iterative centroid–based clustering method, which orders the *n* data points (in this case, term–vectors) into *t* disjoint subsets C_j so as to minimize the sum-of-squares criterion

$$E = \sum_{i=1}^{t} \sum_{j \in C_i} \left| T_j - M_i \right|^2 , \qquad (3)$$

where $\boldsymbol{T_j}$ is the vector representing the j-th term and $\boldsymbol{M_i}$ is the centroid of the vectors in C_i. The algorithm is based on a simple re-estimation procedure: the vectors are assigned to the C_i sets randomly and the centroid of each initial cluster is calculated. In every iteration the vectors are assigned to the cluster with the

[1]The cosine distance of two vectors A and B is the cosine of their angle.

nearest centroid and then the new centroid is recalculated. The algorithm terminates when there are no more reassignments for the term vectors. In general, the global minimum of E is not achieved, but despite of the limitation the algorithm is frequently used as a result of its simplicity. [4]

The t number of the clusters created by k-means is set according to the expected number of term clusters capable to distinguish the document clusters. For the dimensions in the feature space extracted from the result of SVD, a lower limit for t can be estimated to $3k$, since at least the three value intervals ”negative”, ”around zero” and ”positive” should be distinguished for each dimension. The more term clusters we allow, the more chance we have not to loose important differences, but this means the weakening of noise filtering as well.

At this point, the number of dimensions used by k-means from the result of SVD is still an open question. This can be set based on a threshold for the singular values extracted from $\boldsymbol{S}$.

Now the k-means algorithm can be applied to the terms (described by k coordinates in the feature space) to create t term clusters.

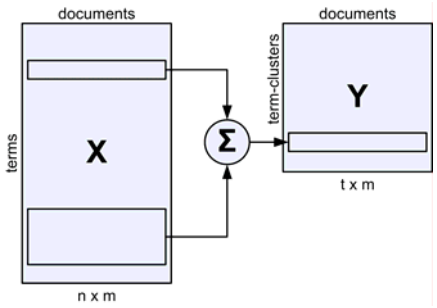

Figure 3
Applying term clusters to document–term frequency matrix

3 Double Clustering

Term clustering described in Section 2 has the advantage of eliminating noise generated by the usage of different words with similar topics. After the term clusters are created, this clustering has to be applied to the document–term frequency matrix. The sum of frequencies of terms occurring in the document and belonging to the same cluster is calculated as shown in Figure 3. The t-by-m document–term-cluster matrix $\boldsymbol{Y}$ is calculated as follows:

$$y_{i,j} = \sum_{c \in C_i} x_{c,j} \tag{4}$$

where following conditions are true:

$$\forall_{1\le i,j\le t} C_i \cap C_j = 0 \tag{5}$$

$$\forall_{1\le i\le t} word_i \in \bigcup_{j=1}^{t} C_j \tag{6}$$

The second clustering process of the double clustering can now be performed on the column vectors of $\boldsymbol{Y}$. For this step, we used the k-means algorithm again. The d number of document clusters was set based on the expected number of document clusters. As there might be ambiguous documents where the correct cluster is harder to determine, more document clusters are suggested to isolate outliers into separate clusters. Otherwise an outlier might be able to take a whole cluster and cause two correct document clusters to be merged by k-means. As the document clustering is performed based on term–clusters instead of single terms, noise caused by synonyms and words with similar usage behaviors are avoided and the result is much more distinct.

4 Experimental Results

The capabilities of our method were tested on a set of documents from two different topics, where many words can be found only in one of the two groups. In the first experiment 6-6 documents were selected from the two categories. The documents from the testing corpus were parsed without using any stop lists or stemming. After parsing the documents, the document–term frequency matrix was created and singular value decomposition was applied. As the number of used dimensions after SVD was $k=3$, the result cannot be visualized in an easy way.

Figure 4 shows the topological distribution of the terms in the feature space according to the second and third dimensions. The three types of notations represent the terms occurring only in one of the two document sets (circles and triangles) and the ones found in documents from both categories (dots). The separation ability of dimension two can be seen unambiguously. As we are using these coordinates to calculate the cosine distances of the documents to get the input of k-means, this separation ability is very important. In an optimal situation, the first dimensions would separate the two document categories and the others would describe differences between the terms inside the document categories. Using the results of the SVD, the terms represented by k dimensional feature vectors are clustered into $t=10$ term–clusters using the k-means algorithm (based on cosine distance). The second clustering of the double clustering method generates the document clusters.

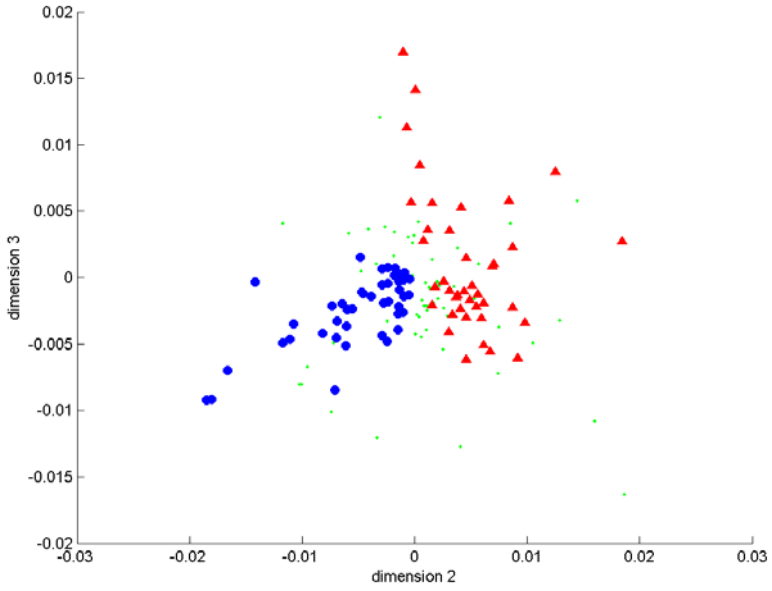

Figure 4
Terms in the feature space after dimensionality reduction

Figure 5, Figure 6, and Figure 7 show the resulting document clusters. The documents (shown along the horizontal axis) are represented by the circles and squares according to the two topic categories.

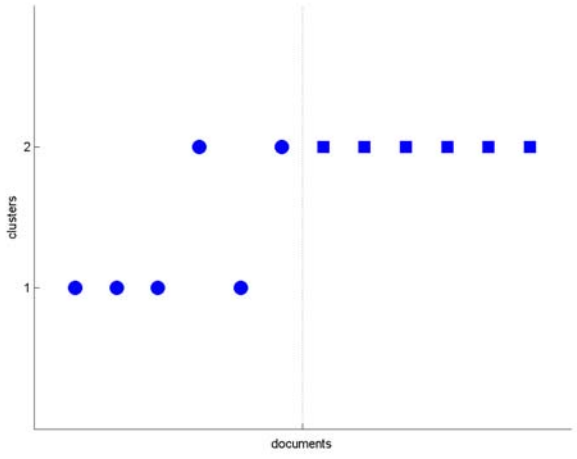

Figure 5
Results with 2 document clusters

The minimal necessary number of clusters would be *d=2*, but as Figure 5 shows, k-means captures wrong similarities and 2 documents are assigned in a wrong way. With *d=3* the classification becomes correct and the two previously false classified documents build the third cluster as shown in Figure 6.

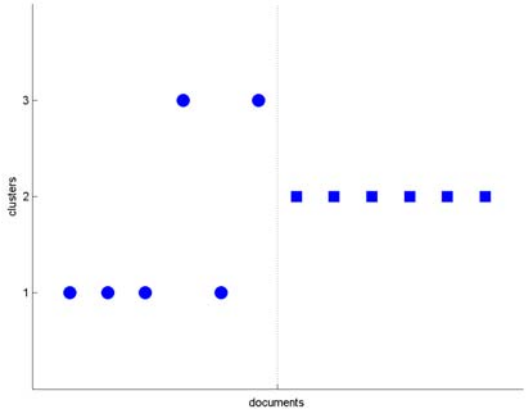

Figure 6
Results with 3 document clusters

Figure 7 shows the whole clustering process: for *d=2* the two clusters are created, but one of them still has to be divided to correctly separate the two document clusters represented by document indexes 1–6 and 7–12.

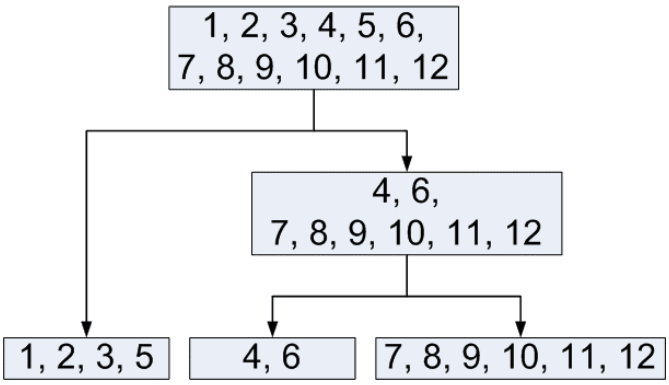

Figure 7
Document clustering results

After the measurement with 12 documents, a larger testing set was selected from the frequently used “20Newsgroups” corpus [5]. 40-40 messages were choosen from the “auto” and the “graphics” subsets in a random way. The original testing

set contained 5707 terms without the application of a stemming phase or stop list filtering. After removing the rare terms occurring less than once in the half of the documents in average, 234 terms remained in the document–term matrix.

A frequently used measure for the performance of clustering algorithms is the F-Measure defined as

$$F = \frac{2PR}{P+R} \tag{7}$$

where P is the precision defined as the rate of correctly assigned elements in the cluster and R is recall, which is the part of the elements originating from the data partition which is approximated by the clustering. If there is a cluster which should contain the documents about graphics, the precision P is the rate of documents about graphics in the cluster and recall is the rate of the correctly assigned documents from all the documents about graphics.

The measurements were performed with $k=3$ dimensions preserved after SVD, and $t=10$ term–clusters were built. Figure 8 shows the results with 2 document clusters. Most of the documents are clustered correctly, and the achieved F-Measure is 0.780179.

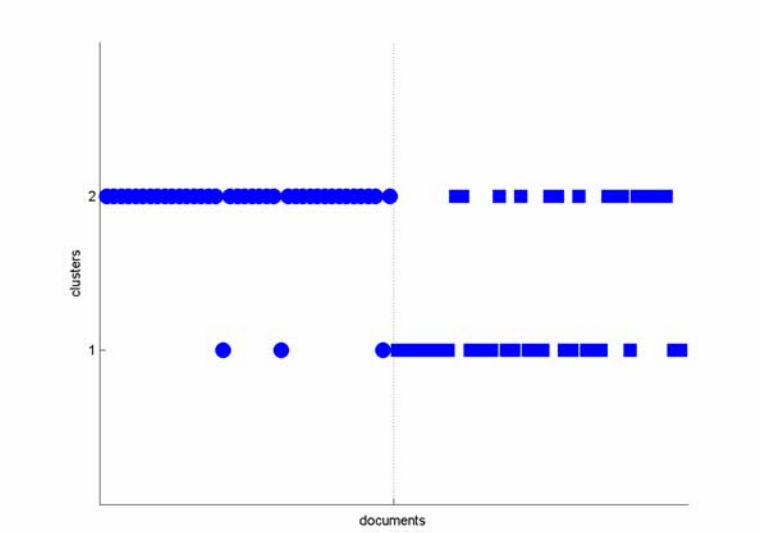

Figure 8
Document clustering results with 2 clusters

If we allow $d=3$ document clusters and leaving the other parameters unchanged, F-Measure becomes worse: 0.652818. This result is shown in Figure 9. In this

case, outliers from both previously recognized clusters were moved to the new cluster and they have been collected instead of separating them.

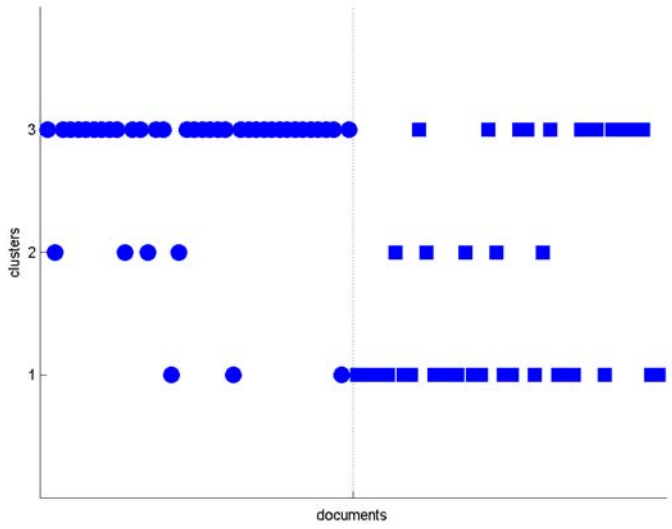

Figure 9
Document clustering results with 3 clusters

This second experiment shows, how important the outliers are: if an additional cluster is capable to collect the outliers, that does not necessarily improve the clustering performance expressed with F-Measure. If outliers cannot be included into the correct cluster, they should be separated without mixing the outliers from more categories together.

Conclusions

The singular value decomposition is an effective method to capture latent semantic information based on similar occurrence of terms in the documents. Through the dimensionality reduction and noise filtering which SVD provides, the performance of document clustering can be increased and the process can be accelerated effectively. The method can be extended with an additional noise filtering through double clustering by clustering the terms in the document corpus, and performing the document classification in a feature space derived from the term–clusters. In this paper a short description of the procedure was provided and two easy-to-visualize examples on small testing corpora have been shown.

References

[1] Cristianini, N., Shawe-Taylor, J., and Lodhi, H.: Latent semantic kernels. Journal of Intelligent Information Systems, 2002, 18(2/3): 127-152, Special Issue on Automated Text Categorization

[2] Furnas, G., Deerwester, S., Dumais, S. T., Landauer, T. K., Harshman, R., Streeter, L. A., and Lochbaum, K. E.: Information retrieval using a singular value decomposition model of latent semantic structure. In Chiaramella, Y., editor, Proceedings of the 11th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1988, pages 465-480

[3] Slonim, N. and Tishby, N.: Document clustering using word clusters via the information bottleneck method. In Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Clustering, 2000, pages 208-215

[4] Weisstein, E.: Mathworld, http://mathworld.wolfram.com/

[5] Lang, K. (1995). Newsweeder: Learning to filter netnews. In ICML, pages 331-339