POZNAN UNIVERSITY OF TECHNOLOGY ACADEMIC JOURNALS
No. 80 Electrical Engineering 2014

Hussam D. ABDULLA*
Abdella S. ABDELRAHMAN*
Vaclav SNASEL*

USING SINGULAR VALUE DECOMPOSITION (SVD) AS A SOLUTION FOR SEARCH RESULT CLUSTERING

There are many search engines in the web, but they return a long list of search results, ranked by their relevancies to the given query. Web users have to go through the list and examine the titles and (short) snippets sequentially to identify their required results. In this paper we present how usage of Singular Value Decomposition (SVD) as a very good solution for search results clustering. Results are presented by visualizing neural network. Neural network is responsive for reducing result dimension to two dimensional space and we are able to present result as a picture that we are able to analyze.

KEYWORDS: singular value decomposition, clustering, self-organizing map

1. INTRODUCTION

In the last few years the world observes exponential growing of the amount of information. Easiness of using this information and easiness of access to this information brew a big problem to retrieval of information, and the results contain a lot of data and it can be hard to choose or find the relevant information in the result. The huge numbers of data and inability to recognize the type of data lead to inability for the right searching for information. For users with no prior experience, searching for topic manually in the web can be difficult and taking time.

The major difficulties are the complicacy of the content and the classification of the huge information in the web, and identifying and naming topics and relationships between these topics. In this situation, clustering data gives us a good result for data analysis. We can use the search result clustering in width area from different fields. In this paper we present one of the methods for clustering data to be used in the search result clustering. We use the singular value decomposition as a mathematical method to reduce a big value of objects by combining the attributes of these objects [1].

* Technical University of Ostrava.

2. SEARCH RESULTS CLUSTERING

The recent years, search result clustering has attracted a substantial amount of research (e.g. information retrieval, machine learning, human-computer interaction, computational linguistics, data mining, formal concept analysis, graph drawing). Search result clustering groups search results by topic. Thus provides us with complementary view to the information returned by big documents ranking systems. This approach is especially useful when document ranking fails to give us a precise result. This method allows a direct access to a subtopic; search result clustering reduces the information, helps filtering out irrelevant items, and favours exploration of unknown or dynamic domains. Search result clustering, is different from the conventional document clustering. When clustering takes place as a post-processing step on the set of results retrieved by an information retrieval system on a query, it may be both more efficient, because the input consists of few hundred of snippets, and more effective, because query-specific text features are used. On the other hand, search result clustering must fulfil a number of more stringent requirements raised by the nature of the application in which it is embedded [2].

3. SINGULAR VALUE DECOMPOSITION (SVD)

Singular Value Decomposition (SVD) breaks a (n×m) matrix A into three matrices U, Σ and V such that $A = U\Sigma V^T$. U is a (n×k) orthogonal matrix whose column vectors are called the left singular vectors of A, V is a (k×m) orthogonal matrix whose column vectors are termed the right singular vectors of A, and Σ is a (k×k) diagonal matrix having the singular values of A ordered decreasingly. Columns of U form an orthogonal basis for the column space of A. Singular value decomposition (SVD) is well-known because of its application in information retrieval as LSI. SVD is especially suitable in its variant for sparse matrices [7, 8, 9]. Since only the first k concepts can be considered are semantic important (the singular values are high), we can approximate the decomposition as:

$$A = U_k \sum_k V_k^T$$

where U_k contains the first k most important concept vectors, Σ_k contains the respective singular values and $\sum_k V_k^T$ contains the pseudo-document vectors represented using the first k concept vectors. In other words, by SVD the original m-dimensional vectors are projected into a vector space of dimension k ($k << m$). The SVD approximation (so-called rank-k SVD) can be created either by "trimming" the full-SVD matrices or by usage of a special method designed to perform directly the rank-k SVD.

Theorem (SVD): Let A be an m×n rank-r matrix. Be $\sigma_1,\ldots,\sigma_y$ eigenvalues of a matrix $\sqrt{AA^T}$. There exist orthogonal matrices $U=(u_1,\ldots,u_r)$ and $V=(v_1,\ldots,v_r)$, whose column vectors are orthonormal, and diagonal matrix $\Sigma = \text{diag}\,(\sigma_1,\ldots,\sigma_r)$. The decomposition $A=U\Sigma V^T$ is referred to as a singular value decomposition of matrix A and numbers $\sigma_1,\ldots,\sigma_r$ are singular values of the matrix A. Columns of U (or V) are referred to as left (or right) singular vectors of matrix A. Now we have a decomposition of the original matrix A. Needless to say, the left and right singular vectors are not sparse. We have at most r nonzero singular numbers, where rank-r is the smaller of the two matrix dimensions. Because the singular values usually decrease quickly, we only need to take k greatest singular values and corresponding singular vector coordinates and create a k-reduced singular decomposition of matrix A.

Definition: Let us have k, 0 < k < r and SVD of A

$$A = U\Sigma V^T = (U_K U_0)\begin{pmatrix}\Sigma_K & 0\\ 0 & \Sigma_0\end{pmatrix}\begin{pmatrix}V_K^T\\ V_0^T\end{pmatrix}$$

$A = U_K \Sigma_K V_K^T$ is referred to as a k-reduced SVD (rank-k SVD). In information retrieval, if every document is relevant to only one topic, we obtain a latent semantics related words and documents will have similar vectors in the reduced space. For an illustration of rank-k SVD see Fig. 1, the grey areas determine first k coordinates from singular vectors, which are being used.

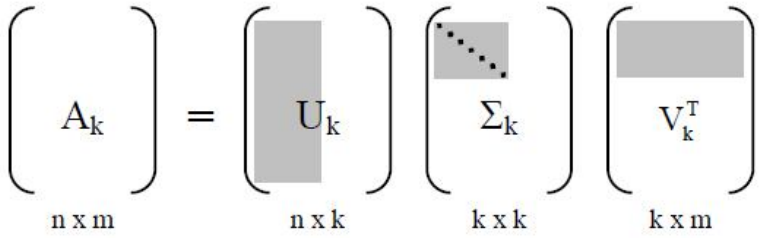


Fig. 1. k-reduced singular value decomposition

Theorem: Among all m×n matrices C of rank at most k, is the one that minimizes

$$\| A_k - A \|_F^2 = \sum_{i,j} (A_{i,j} - C_{w,j})^2 \cdot$$

Because rank-k SVD is the best rank-k approximation of original matrix A, any other decomposition will increase the sum of squares of matrix $A - A_K$.

The SVD is hard to compute and once computed, it reflects only the decomposition of the original matrix. The recalculation of SVD is expensive, so it is impossible to recalculate SVD every time new rows or columns are inserted.

The SVD-Updating is a partial solution, but since the error increases slightly with inserted rows and columns when updates occur frequently, the recalculation of SVD may be needed [3, 4, 5, 6].

4. SELF-ORGANIZING MAPS

Self-Organizing Map (SOM) [5] is a competitive artificial neural network. They are used to classify and cluster data set according to similarity [2]. SOM artificial network is structured in two layers. The first one represents the input data, the second one is a neuron's grid, usually bi-dimensional, full connected. Each input is connected to all output neurons. Output neurons are arranged in low dimensional (usually 2D or 3D) grid. Attached to every neuron there is a weight vector with the same dimensionality as the input vectors. The number of input dimensions is usually a lot higher than the output grid dimension. SOMs are mainly used for dimensionality reduction rather than expansion.

5. UNIFIED DISTANCE MATRIX

The unified distance matrix (U-matrix) makes the 2D visualization of multi-variant data possible using SOM's code vectors as data source [8]. This is achieved by using topological relations property among neurons after the learning process. U-matrix contains the distances from each unit centre to all of its neighbours. By U-matrix we can detect topological relations among neurons and infer about the input data structure. High values in the U-matrix represent a frontier region between clusters, and low values represent a high degree of similarities among neurons on that region, clusters. This can be a visual task when we use some colour scheme.

6. PROBLEM FORMALIZATION AND ALGORITHM

The distinctive characteristic of the algorithm is that it identifies meaningful cluster labels and only then assigns search results to these labels to build proper clusters.

The algorithm consists of 6 steps:

1. Pre-processing the input snippets, this includes tokenization, stemming and stop-word marking.
2. Identifies words and sequences of words appearing in the input snippets.

3. Matrix decomposition is used to induce cluster labels.
4. Snippets are assigned to each of these labels to form proper clusters.
5. Post-processing, which includes cluster merging and pruning.
6. Present the results by Self-Organizing Map (SOM)

The step 3 is the core of the algorithm, because this step relies on the Vector Space Model and a term-document matrix A having n rows, where n is the number of input snippets, and m columns, where m is the distinct words found in the input snippets. Each element A*mn* of A numerically represents the relationship between word *m* and snippet *n*.

The singular value decomposition may be applied on the binary matrix A which created from the step 2, where the rows of the matrix are the input snippets (objects), and the columns are the distinct words found in the input snippets (attributes), presented as 0 and 1.(0 – the distinct word not found in the input snippet, 1 – the distinct word found in the input snippet).

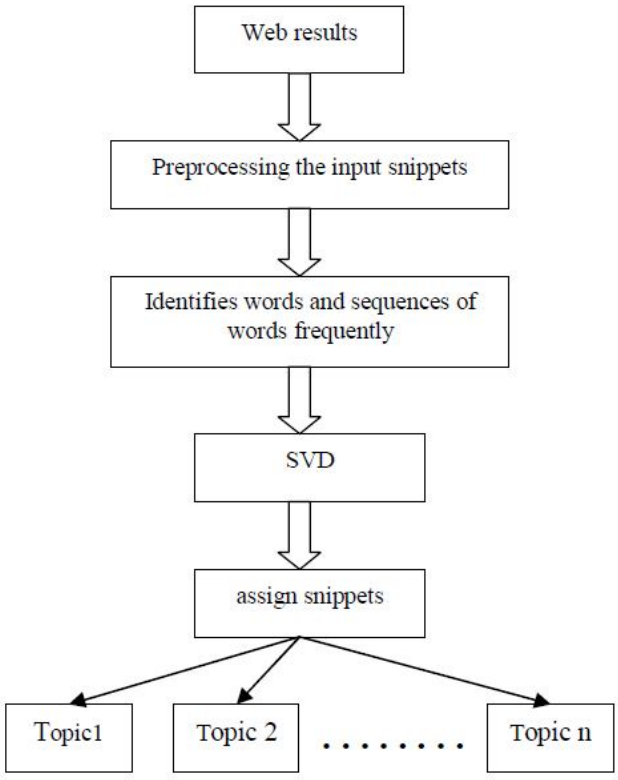


Fig. 2. Steps of the algorithm process

Since the rank-k SVD is known to remove noise by ignoring small differences between row and column vectors of A (they will correspond to small singular values, which we drop by the choice of k), it can be used in our algorithm, Because SVD creates equivalence classes of data from the original data through deleting and adding some no primary attributes in the objects, this process leads the objects to have similarity in their attributes. From this similarity the algorithm combines these objects that have the same attributes and presents them like one object. The new object created from this process represent a group of snippets has a similar distinct word. That means that the algorithm can minimize the huge volume of snippets received as a result from the searching in the web.

7. EXPERIMENT

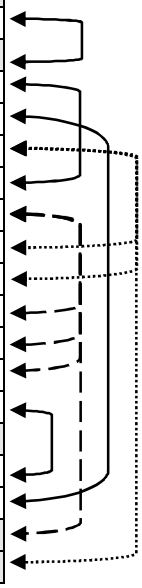
We applied our experiment on data created from Google search engine contained from 100 snippets with 35 distinct different words, but for easy explanation how the algorithm works, we deduct a small part from these data. The data which we deducted from the result of Google search engine (Table 1) are 20 objects (snippets) with 6 attributes (distinct words).

Table 1. Data result Google search engine

	A1	A2	A3	A4	A5	A6
O1	0	0	0	0	0	1
O2	1	0	0	1	0	1
O3	0	0	0	1	0	1
O4	0	0	0	1	0	0
O5	1	0	0	1	0	0
O6	1	0	0	0	0	0
O7	0	1	1	1	1	0
O8	0	0	0	1	1	0
O9	0	1	1	0	0	0
O10	0	1	0	1	1	0
O11	0	0	1	1	1	0
O12	0	1	1	0	1	0
O13	0	1	1	1	0	0
O14	0	1	0	1	0	0
O15	0	1	0	0	1	0
O16	0	0	1	1	0	0
O17	0	0	1	0	1	0
O18	1	0	0	1	1	0
O19	0	1	1	0	0	1
O20	1	1	1	0	0	1

Table 2. Data after using SVD rank-k

A1	A2	A3	A4	A5	A6
0	0	0	0	0	0
1	0	0	1	0	0
1	0	0	0	0	1
1	0	0	0	0	1
1	0	0	0	1	0
0	0	0	0	1	0
0	1	1	1	1	0
1	0	0	0	1	0
0	1	1	1	0	0
0	1	1	1	1	0
0	1	1	1	1	0
0	1	1	1	0	0
0	1	1	1	0	0
0	1	1	1	0	0
0	0	0	1	0	0
1	1	1	1	0	1
0	0	0	1	0	0
0	0	0	0	1	0
0	1	1	1	0	0
0	1	1	1	1	0



After using SVD with good choice rank-k, we can see that the output data from the SVD method are changed and brewed new attributes combinations in the objects, these combinations repeated in more than one object (Table 2).

From this repeating of attributes combination in the objects, not imperative to represent all the objects with the same attributes, we can only represent the objects that have peculiar combinations of attributes (Table 3).

From the above table, we can see how these attributes changed to have similar attributes combination. And we can also see that we have a more active change when the object has more attributes.

This process minimizes the representing data to 40% from the original data with saving the primary attributes in the objects.

Table 3. Representing data after combining the similar Objects

	A1	A2	A3	A4	A5	A6
O1	0	0	0	0	0	0
O2	1	0	0	1	0	0
O3	1	0	0	0	0	1
O4	1	0	0	0	1	0
O5	0	0	0	0	1	0
O6	0	1	1	1	1	0
O7	0	1	1	1	0	0
O8	0	0	0	1	0	0

The activity of this method is shown clearly on the huge data with big quantity of attributes, and for a good choice rank-k from the diagonal matrix when applying SVD. Fig. 3 showed how the rank-k value plays role in the number of attributes combination.

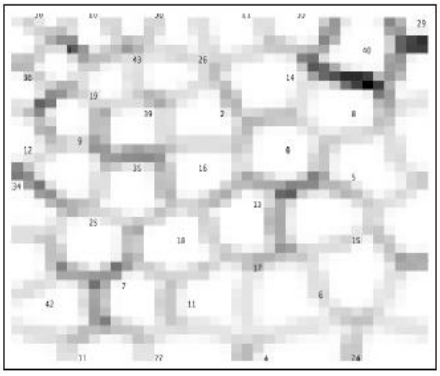


Fig. 3. Original data

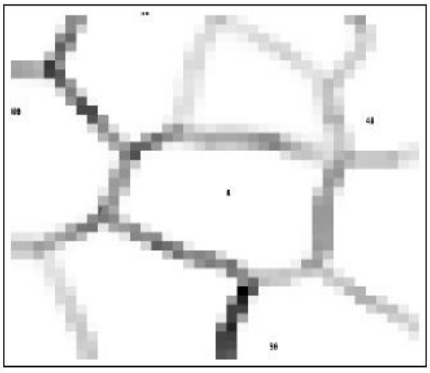


Fig. 4. Data after using SVD

8. VISUALIZING RESULT

First SOM was learned by giving these initial data Second SOM was learned by giving data after applying SVD on initial data. There were two SOM networks according to type of input data. Next part of research was about visualizing SOM network via unified distance matrix (U-matrix) algorithm. Euclidean metric was used for computing distances between nodes. We got two grey scale pictures (Fig. 3, Fig. 4) that we are able to analyze, especially number and form of visible clusters.

9. CONCLUSION AND FUTURE WORK

Singular Value Decomposition (SVD) with rank-k depends on deleting and adding some attributes from the objects in the original data. This way can give us minimum primary attributes to collect more objects that have the same combination of attributes. Applying this method of decomposition on web searching problems gives us a good solution for search results clustering.

REFERENCES

[1] Stanislaw Osinski, "Improving Quality of Search Results Clustering with Approximate Matrix Factorisations", ECIR 2006.
[2] Hua-Jun Zeng, Qi-Cai He, Zheng Chen, Wei-Ying Ma, Jinwen Ma, "Learning to cluster web search results", SIGIR 2004.
[3] Vasclav Snasel, Petr Gajdos, Hussam Dahwa Abdulla and martin Polovincak, "Concept Lattice Reduction by Matrix Decompositions", DCCA 2007.
[4] Vaclav Snasel, Hussam Dahwa Abdulla and Martin Polovincak, "Behavior of the Concept Lattice Reduction to visualizing data after Using Matrix Decompositions", IEEE Innovations'07, 2007.
[5] Vasclav Snasel, martin Polovincak, Hussam Dahwa Abdulla and Zdenek Horak, "On Knowledge Structures Reduction", IEEE CISIM 2008.
[6] Václav Snásel, Martin Polovincak, Hussam M. Dahwa Abdulla, Zdenek Horak: On Concept Lattices and Implication Bases from Reduced Contexts. ICCS Supplement 2008.
[7] M. Berry and M. Browne, "Understanding Search Engines, Mathematical Modelling and Text Retrieval", Siam, 1999.
[8] R. M. Larsen, "Lanczos bidiagonalization with partial reorthogonalization", Technical report, University of Aarhus, 1998.