Concept Lattice Reduction by Singular Value Decomposition

© Vaclav Snasel Martin Polovincak Hussam M. Dahwa

VSB Technical University Ostrava, Czech Republic
{vaclav.snasel, martin.polovincak.fei, hussamdahwa}@vsb.cz

Abstract

High complexity of lattice construction algorithms and uneasy way of visualising lattices are two important problems connected with the formal concept analysis. Algorithm complexity plays significant role when computing all concepts from a huge incidence matrix. In this paper we try to modify an incidence matrix using matrix decomposition, creating a new matrix with fewer dimensions as an input for some known algorithms for lattice construction. Results are presented by visualising 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 analyse.

1 Introduction

We are dealing with uses of matrix decompositions for reduction of concept lattice. These methods are well known in the area of information retrieval under the name Latent Semantic Indexing (LSI) or Latent Semantic Analysis (LSA). LSI and LSA have been used for discovery of latent dependencies between terms (or documents) [1], [6]. We would like to apply this approach in the area of formal concept analysis (FCA). In this paper, we want to present decomposition's results with neural network [5]. First, we will introduce basic terms of Formal Concept Analysis (FCA) [3] and then we will describe the rank-k singe value decomposition (SVD) [8]. Singular Value Decomposition is one of the various matrix decomposition techniques arising from numerical linear algebra. SVD reduces both the column space and the row space of the term-document matrix to lower dimensional spaces [4]. After using SVD on input matrix we can build lattices and visualise them using neural networks[5] and u-matrix [9]. Numerous studies suggest that graphical representation and display of results can improve information retrieval performance [10]. Once having visualisation we can analyze result of matrix reduction.

2 Concept Lattice Reduction

In the following paragraphs we would like to introduce important basic terms of formal concept analysis, singular value decomposition, self organized maps and unified distrance matrix.

2.1 Formal concept analysis

Formal Concept Analysis was first introduced by Rudolf Wille in 1980. FCA is based on the philosophical understanding of the world in terms of objects and attributes. It is assumed that a relation exists to connect objects to the attributes they possess. Formal context and formal concept are the fundamental notions of Formal Concept Analysis [3].

A formal context $C = (G, M, I)$ consists of two sets; G and M, with I in relation to G and M. The elements of G are defined as objects and the elements of M are defined as attributes of the context. In order to express that an object $g \in G$ is related to I with the attribute $m \in M$, we record it as gIm or $(g, m) \in I$ and read it as the object g has the attribute m. I is also defined as the context incidence relation.

For a set $A \subset G$ of object we define

$$A^{'} = \{m \in M \mid gIm \; for \; all \; g \in A\}$$

(the set of attributes common to the objects in A). Correspondingly, for a set B of attributes we define

$$B^{'} = \{g \in G \mid gIm \; for \; all \; m \in B\}$$

(the set of objects which have all attributes in B).

A formal concept of the context (G, M, I) is a pair (A, B) with $A \subseteq G$, $B \subseteq M$, $A^{'} = B$ and $B^{'} = A$. We call A the extent and B the intent of the concept (A, B). $B(G, M, I)$ denotes the set of all concepts of context (G, M, I).

The concept lattice $\underline{B}(G, M, I)$ is a complete lattice in which infimum and supremum are given by:

$$\bigwedge_{t \in T} (A_t, B_t) = \left(\bigcap_{t \in T} A_t, \left(\bigcup_{t \in T} B_t \right)^{''} \right)$$

$$\bigvee_{t \in T} (A_t, B_t) = \left(\left(\bigcup_{t \in T} A_t \right)^{''}, \bigcap_{t \in T} B_t \right).$$

We refer to [3].

2.2 Singular Value Decomposition

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].

Proceedings of the Spring Young Researcher's Colloquium on Database and Information Systems, Moscow, Russia, 2007

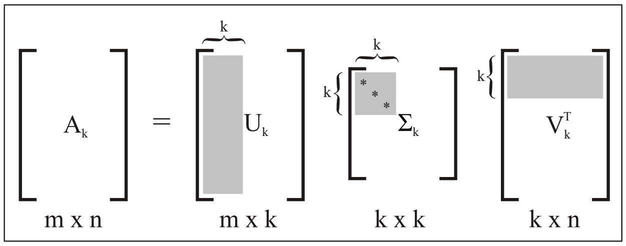

Figure 1: k-reduced singular value decomposition

Theorem 1: Let A is an m n rank-r matrix. Be $\sigma_1 \geq \cdots \geq \sigma_r$ eigenvalues of a matrix $\sqrt{AA^T}$. Then there exist orthogonal matrices $U = (u_1, \ldots, u_r)$ and $V = (v_1, \ldots, v_r)$, whose column vectors are orthonormal, and a diagonal matrix $\Sigma = diag(\sigma_1, \ldots, \sigma_r)$. The decomposition $A = U\Sigma V^T$ is called *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 called *left (or right) singular vectors* of matrix A.

Now we have a decomposition of the original matrix A. It is not needed to say, that 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. However, we would not conserve much memory by storing the term-by-document matrix this way. Luckily, because the singular values usually fall quickly, we can take only k greatest singular values and corresponding singular vector co-ordinates and create a *k-reduced singular decomposition* of A.

Let us have $k, 0 < k < r$ and singular value decomposition of A

$$A = U\Sigma V^T = (U_k U_0) \begin{pmatrix} \Sigma_k & 0 \\ 0 & \Sigma_0 \end{pmatrix} \begin{pmatrix} V_k^T \\ V_0^T \end{pmatrix}$$

We call $A_k = U_k \Sigma_k V_k^T$ a *k-reduced singular value decomposition* (*rank-k SVD*).

In information retrieval, if every document is relevant to only one topic, we obtain a latent semantics - semantically related words and documents will have similar vectors in the reduced space. For an illustration of rank-k SVD see figure 1, the grey areas determine first k coordinates from singular vectors, which are being used.

Theorem 2: (Eckart-Young) Among all $m \times n$ matrices C of rank at most k A_k is the one, that minimises $||A_k - A||_F^2 = \sum_{i,j} (A_{i,j} - C_{w,j})^2$.

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.

Note: From now on, we will assume rank-k singular value decomposition when speaking about SVD.

2.3 Self-organizing maps

Kohonen Self-Organizing Map (SOM) [5] is a competitive artificial neural network. They are used to clasify and cluster data set according to similarity [2]. SOM artifical network is structured in two layers. The first one represents the input data, the second one is a neuron's grid, usually bidimensional, 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.

2.4 Unified distance matrix

The unified distance matrix (U-matrix) makes the 2D visualization of multi-variate data possible using SOM's codevectors as data source [9]. This is achieved by using topological relations property among neurons after the learning process. U-matrix contains the distances from each unit center 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 color schema.

3 Our Experiment

3.1 Input data

Data from the Ostrava Mass Transit Authority was used in this experiment. We can describe data in a graph. Rows represent tram stops, columns represent tram lines. Let 1 represent a tram stopping at a tram stop in the appropriate row and column crossing, otherwise zero. This data was chosen because we can easily see if the SVD is working. Many tram lines share the same tram stops. These parts of tram lines should be minimized with SVD and it should also be visible after using SVD.

3.2 Steps of our Experiment

Our experiment included several steps. The first step was to read and transform data into adjacency matrix. After transforming and computing (using SVD) three new matrixes were obtained as results from SVD $A = U\Sigma V^T$. After choosing rank-k, a new matrix was computed. Lattices were build from both original matrix and matrix computed by SVD. Concepts of lattices acted as the source for SOMs learning phase. Having learned SOMs, visualisation as done using U-matrix technique.

3.3 Reading and transforming data

Data was obtained from tram timetables. The first ten tram lines and their tram stops were chosen. The transformed matrix had 92 rows and 10 columns. Transformation was simple. If a tram stops at a particular tram stop, the number 1 appeared in the appropriate row and column crossing, otherwise zero. The matrix is also an

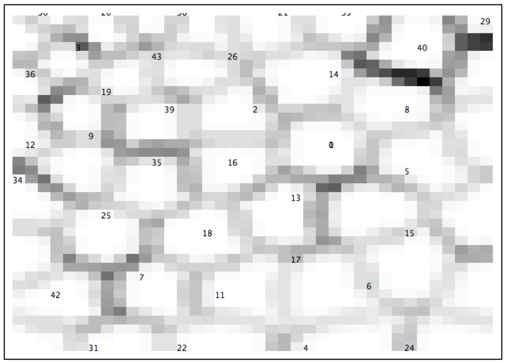

Figure 2: Original data

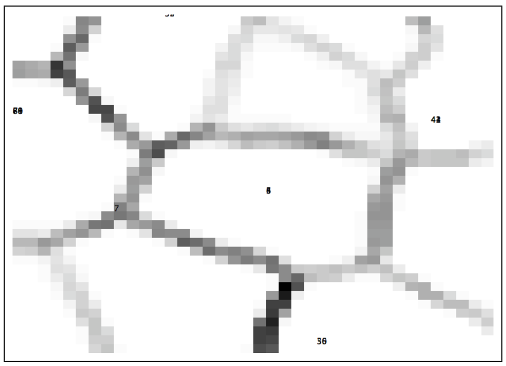

Figure 3: Data after using SVD

incidence matrix and was used in SVD, NMF and FCA computations.

3.4 Using SVD

SVD computations were made by SVDLIBC-fix software. This software is free and is written in C language. An incidence matrix acts as the input. Software can compute all three matrixes - U and Σ and V^T. k-rank was chosen $k = 4$. Matrixes were U(92x10), Σ(10x10) and V^T(10x10).

3.5 Visualising result

Data from the Ostrava Mass Transit Authority was used in this experiment. We can describe data in a graph. Rows represent tram stops, columns represent tram lines. Let ones represent a tram stopping at the tram stop in the appropriate row and column crossing, otherwise zero.

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 visualising SOM network via unified distance matrix (U-matrix) algorithm. Euclidean metric was used for computing distances between nodes. We got two greyscale pictures (figure 2, figure 3) that we are able to analyze, especialy number and form of visible clusters.

4 Conclusion

There are two u-matrix visualisations. Figure 2 matches concepts of lattice built from the original matrix, figure 3 matches concepts of lattice built from the original matrix changed with SVD. The number of differencies between rows in tables was decreased by SVD computation. Thus you can see smaller number of clusters. There is no doubt method was successful in reducing number of concepts.

References

[1] M. Berry and M. Browne. Understanding search engines. *Mathematical Modeling and Text Retrieval*, 1999.

[2] C. Lee C. Kiu. Discovering ontological semantics using fca and som. *M2USIC 2004*, 2004.

[3] B. Ganter and R. Wille. *Formal Concept Analysis*. Springer-Verlag Berlin Heidelberg, 1999.

[4] Douglas Vogel Karen S.K. Cheung. Complexity reduction in lattice-based information retrieval. *Information Retrieval, 8*, page 285299, 2005.

[5] T. Kohonen. *Self-organizing maps*. Springer, 2001.

[6] T.Letsche M. Berry, S. Dumais. Computation methods for intelligent information access. *Proceedings of the 1995 ACM/IEEE Supercomputing Conference*, 1995.

[7] T.Letsche M. Berry, S. Dumais. Computation methods for intelligent information access. *Proceedings of the 1995 ACM/IEEE Supercomputing Conference*, 1995.

[8] P. Moravec P. Gajdos. Concept lattice generation by singular value decomposition. *CLA 2004*, pages 13–22, 2004.

[9] J. Vesanto. Som-based data visualization methods. pages 6–9, 1999.

[10] Lin X. Map displays for information retrieval. *Journal of the American Society of Information Science*, pages 40–54, 1997.