Concept Lattice Generation by Singular Value Decomposition

Petr Gajdoš, Pavel Moravec, and Václav Snášel

Department of Computer Science, VŠB – Technical University of Ostrava
17. listopadu 15, 708 33 Ostrava–Poruba, Czech Republic
{petr.gajdos, pavel.moravec, vaclav.snasel}@vsb.cz

Abstract. Latent semantic indexing (LSI) is an application of numerical method called singular value decomposition (SVD), which discovers latent semantic in documents by creating concepts from existing terms. The application area is not limited to text retrieval, many applications such as image compression are known. We propose usage of SVD as a possible data mining method and lattice size reduction tool. We offer in this paper preliminary experiments to support usability of proposed method.

Keywords: LSI, SVD, FCA, concept lattice

1 Introduction

We are studying the possible usage of *k-reduced singular value decomposition* (*rank-k SVD*) for reduction of concept lattice size. This method is well known in the area of Information Retrieval under the name *latent semantic indexing* (*LSI*), where it has been used for discovery of latent dependencies between terms (or documents). We would like to apply this approach in the area of *formal concept analysis* (*FCA*). The main goal of this paper is to provide primary results obtained with this method.

We will first mention FCA and describe the rank-k SVD decomposition and LSI method. In second chapter we will mention our proposed approach and two parameters which help us to tune it together with an example. In third chapter we will show preliminary experiment on small set of objects and attributes from real-life data. In conclusion, we will mention the expected outcome of our research.

1.1 Formal Concept Analysis

Definition 1. *A* **formal context** $C := (G, M, I)$ *consists of two sets* G *and* M *and relation* I *between* G *and* M*. The elements of* G *are called the objects and the elements of* M *are called the attributes of the context. In order to express that*

© V. Snášel, R. Bělohlávek (Eds.): CLA 2004, pp. 102–110, ISBN 80-248-0597-9.
VŠB – Technical University of Ostrava, Dept. of Computer Science, 2004.

an object g is in a relation I with an attribute m, we write gIm or $(g, m) \in I$ *and read it as "the object g has the attribute m". The relation I is also called the incidence relation of the context.*

Definition 2. *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).

Definition 3. *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). $\mathfrak{B}(G, M, I)$ *denotes the set of all concepts of context* (G, M, I)

Definition 4. *The concept lattice* $\underline{\mathfrak{B}}(G, M, I)$ *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 [4].

1.2 Singular Value Decomposition

Singular value decomposition (*SVD*) is well known because of its application in information retrieval – *Latent semantic indexing* (*LSI*) [1, 2]. SVD is especially suitable in its variant for sparse matrices (Lanczos [5]).

Theorem 1 (Singular value decomposition). *Let A is an* $n \times m$ *rank-r matrix. Be* $\sigma_1 \geq \ldots \geq \sigma_r$ *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 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 coordinates and create a *k-reduced singular decomposition* of A.

Definition 5. *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.

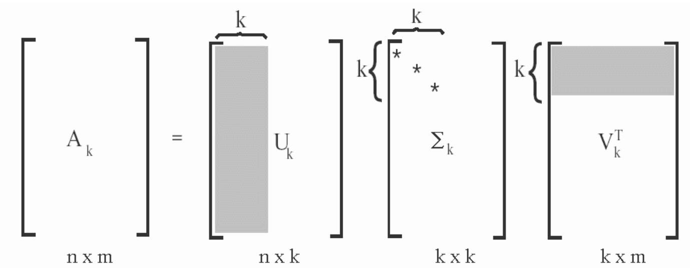


Fig. 1. k-reduced singular value decomposition

Theorem 2 (Eckart-Young). *Among all $n \times m$ 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 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* [6] is a partial solution, but since the error slightly increases with inserted rows and columns, if the updates happen frequently, the recalculation of SVD may be needed soon or later.

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

2 Proposed Usage of Singular Value Decomposition

As we already mentioned, our goals are the ability to build concept lattice on binary data which may contain noise and the reduction of node count in concept lattice.

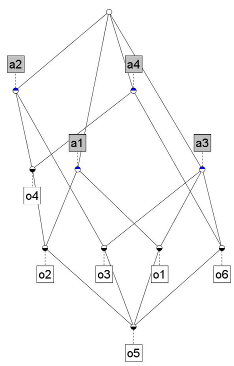
The singular value decomposition may be applied on incidence matrix of objects and attributes (features) A. Since rank-k SVD is known to remove noise by ignoring small differences between row and column vectors of A – they will correspond to small singular values, which we drop by the choice of k – it can be used for data mining.

Because the SVD is mainly used on real numbers, we must solve a problem that the resulting decomposition will also provide real numbers, regardless to the fact that the input matrix is binary. The matrices U_k and V_k don't allow us to create the reduced concept lattice directly, but we can build the reconstructed matrix A_k. It will not contain binary values, so we will have to choose a threshold t discriminating between ones and zeros.

This done, we have two parameters of our method – the reduced dimension k and the threshold t, where generally lower k will result in more unified values and less noise (but if used incorrectly it could ignore important attributes to some extent, too) and higher t in less attributes of objects and thus in less nodes in concept lattice.

	a_1	a_2	a_3	a_4
o_1	1	0	1	0
o_2	1	1	0	1
o_3	0	1	1	0
o_4	0	1	0	1
o_5	1	1	1	1
o_6	0	0	1	1

a)



b)

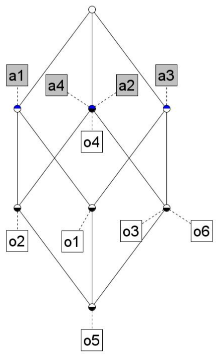
Fig. 2. Example 1: a) incidence matrix, b) concept lattice

Moreover, we can say that SVD creates equivalence classes of nodes in original lattice, merging them into one node in the modified lattice.

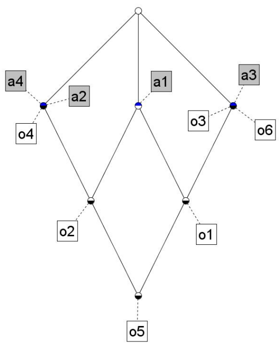
The value of k in information retrieval was experimentally determined as several tens or hundreds (e.g. 50–250), exact value of k for our method is currently not known and has yet to be determined, for example by observing the singular numbers, as their value should drop quickly from some point.

On the other hand we will not encounter the problem with updating the SVD, since the incidence matrix is usually small enough for fast calculation and the lattice has to be reconstructed every time the incidence matrix changes.

Example 1. To exemplify the method, suppose there are six objects with four attributes, as shown in figure 2. We calculated SVD with k 3 and 2 and chosen thresholds $t_1 = 0.1, t_2 = 0.6$. The results are shown in figures 3 and 4.



a)



b)

Fig. 3. Example 1: Generated concept lattices for $k = 3$ and a) $t = 0.1$, b) $t = 0.6$

As one can see, lower k led to a higher reduction as well as higher threshold values, which was expected. In case of $k = 2$ the reduction seem to be to high, leading to a concept lattice where only two groups of object and attributes exist. The question, whether this method creates a viable reduction of concept lattice has to yet to be evaluated in the future.

You can also see, that i.e. nodes in example 1 labeled o_3 and o_6 correspond to one node in reduced lattice in figure 3 a).

There are other methods of lattice size reduction like constraining the concept lattice by attribute dependencies [3], but they require table of relationships between attributes. We are trying to construct dependencies between attributes automatically by the usage of SVD.

3 Experimental Results

We used a data set of Bacterial Taxonomy often used for clustering tests, which was given by Rataj & Schindler. Data are presented for six bacteria species, most

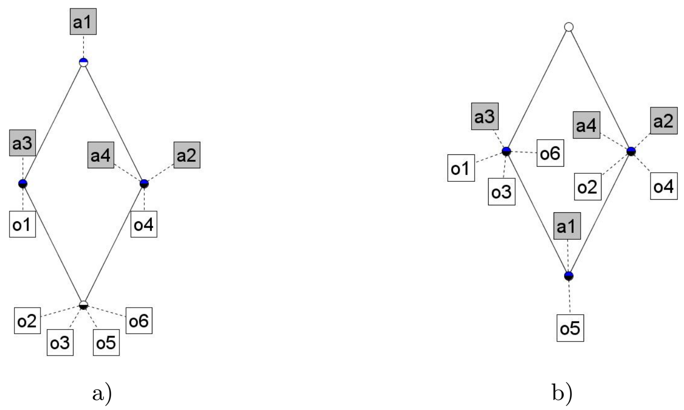


Fig. 4. Example 1: Generated concept lattices for $k = 2$ and a) $t = 0.1$, b) $t = 0.6$

having data for more than one strain and 16 phenotypic characters[1] (0 – absent, 1 – present). The incidence matrix is shown in table 1.

Table 1. Original Bacteria incidence matrix

	H_2S	MAN	LYS	IND	ORN	CIT	URE	ONP	VPT	INO	LIP	PHE	MAL	ADO	ARA	RHA
$ecoli_1$	0	1	1	1	0	0	0	1	0	0	0	0	0	0	1	1
$ecoli_2$	0	1	0	1	1	0	0	1	0	0	0	0	0	0	1	0
$ecoli_3$	1	1	0	1	1	0	0	1	0	0	0	0	0	0	1	1
$styphi_1$	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0
$styphi_2$	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
$styphi_3$	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0
$kpneu_1$	0	1	1	1	0	1	1	1	1	1	0	0	0	1	1	1
$kpneu_2$	0	1	1	1	0	1	1	1	1	1	0	0	1	0	1	1
$kpneu_3$	0	1	1	1	0	1	1	1	1	1	0	0	1	1	1	1
$kpneu_4$	0	1	1	1	0	1	1	1	0	1	0	0	1	1	1	1
$kpneu_5$	0	1	1	1	0	1	0	1	1	1	0	0	1	1	1	1
$pvul_1$	1	0	0	1	0	1	1	0	0	0	0	1	0	0	0	0
$pvul_2$	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0
$pvul_3$	1	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0
$pmor_1$	0	0	1	1	1	0	1	0	0	0	0	1	0	0	0	0
$pmor_2$	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0
smar	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	0

[1] For full specie names see e.g. http://149.170.199.144/multivar/ca_eg.htm#Example2

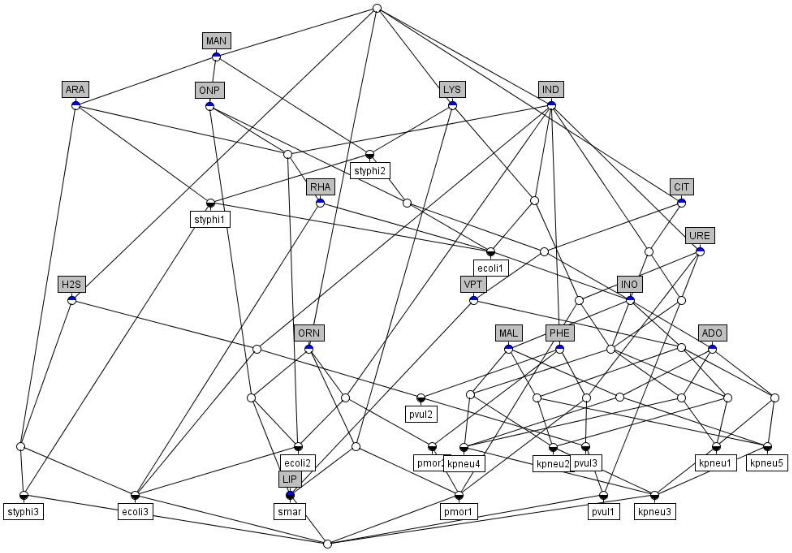


Fig. 5. Original Bacteria concept lattice

The original concept lattice is shown in figure 5, whilst reduced ones in figures 7 and 6. Some modified lattices, i.e. one with $k = 9$ and $t = 0.5$ correspond to original concept lattice.

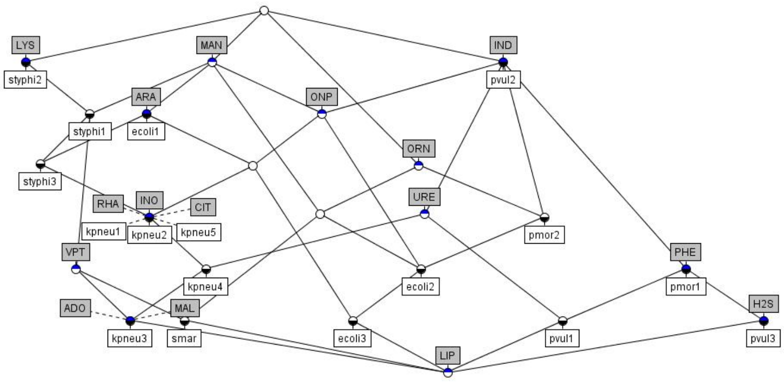


Fig. 6. Modified Bacteria concept lattice, $k = 5, t = 0.85$

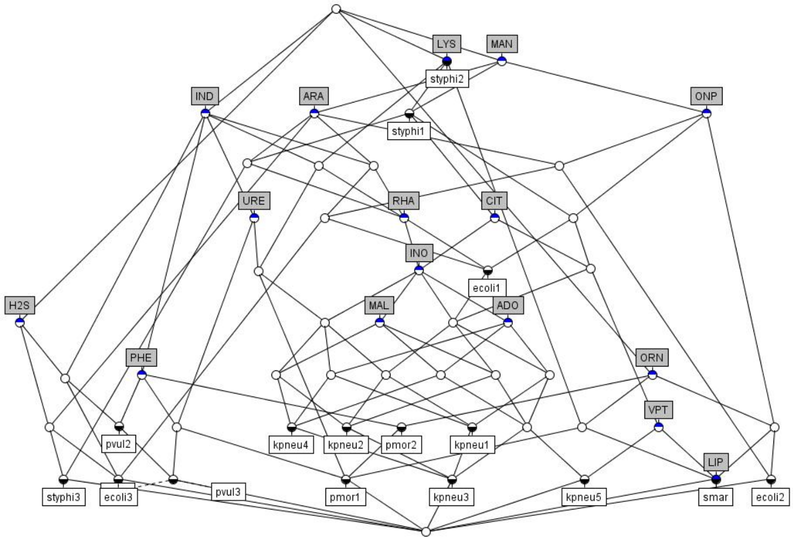


Fig. 7. Modified Bacteria concept lattice, $k = 9, t = 0.85$

One can see, that the modified lattices are quite smaller than original one, the question whether this reduction is a good one is still open.

4 Conclusion and Future Work

In our work we have shown new approach to lattice size reduction and creation from data which may contain some noise through the usage of k-reduced singular decomposition. Our method brings some promising results and should be tested on larger data.

We have yet to determine, if the resulting lattice is a suitable one and what are the main properties of this reduction technique. The other problem is a good choice of k and t and decision, if they are fixed values or functions of the highest and lowest values produced by SVD.

The proposed method could lead to the construction of a new technique of lattice traversal, but it has yet to be determined, whether the complexity would not be too high for its usage.

We would like to compare our method with results of Palacký University team, led by Doc. Belohlavek, in the area of attribute dependencies [3] and equivalence relations on concept lattices.

References

1. M. Berry and M. Browne. *Understanding Search Engines, Mathematical Modeling and Text Retrieval.* Siam, 1999.
2. M. Berry, S. Dumais, and T. Letsche. Computation Methods for Intelligent Information Access. In *Proceedings of the 1995 ACM/IEEE Supercomputing Conference*, 1995.
3. R. Bělohlávek, V. Sklenář, and J. Zacpal. Concept Lattices Constrained by Attribute Dependencies. In *Proceedings of Dateso 2004, CEUR Workshop Proceedings*, volume 98, pages 56–66, 2004.
4. B. Ganter and R. Wille. *Formal Concept Analysis.* Springer-Verlag Berlin Heidelberg, 1999.
5. R. M. Larsen. Lanczos bidiagonalization with partial reorthogonalization. Technical report, University of Aarhus, 1998.
6. G. W. O'Brien. Information Management Tools for Updating an SVD-Encoded Indexing Scheme. Technical Report ut-cs-94-258, The University of Tennessee, Knoxville, USA, December, 1994.