



Complexity Reduction in Lattice-Based Information Retrieval

KAREN S.K. CHEUNG

DOUGLAS VOGEL

Department of Information Systems, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon, Hong Kong

Abstract. Though lattice-based information representation has the advantage of providing efficient visual interface over textual display, the complexity of a lattice may grow rapidly with the size of the database. In this paper we formally draw the analogy between Vector Space Model and Concept Lattice, from which we introduce the notion of Term-Document Lattice as a model for information retrieval. We then propose to use the idea of quotient lattice to reduce the complexity of a Term-Document Lattice. The equivalence relation required to construct the quotient lattice is obtained by performing a Singular Value Decomposition on the original term-document matrix.

Keywords: information retrieval, lattice theory, concept lattice

1. Introduction

The amount of information available to the public has been growing exponentially in recent decades. In the past few years, the World Wide Web has facilitated an explosion of informal information as well. Information are available everywhere, the potential for the retrieval of information is vast, and at times daunting.

Numerous studies suggest that graphical representation and display of searched results can improve information retrieval performance (Lin 1997, Bruza and Dennis 1997, Bruza and McArthur 2000, McArthur and Bruza 2000). In contrast to the conventional search methods such as keyword search and textual display of relevant documents, a graphical information display can provide a broad and concise representation of the searched results from which the searchers can quickly comprehend their relevance and importance. The graphical information display not only can relieve the cognitive overload of the searchers, but also it can improve the low precision and low recall of the searched results. In addition, a user-friendly graphical display enables those users who are lacking of precise information requirement to browse and navigate easily during the search process. Further similar findings have been found in a recent study (Lin 1997), in which a detailed survey is presented on how visualizations can enhance information retrieval by allowing searchers to browse through a graphical representation of the requested documents. It is therefore of utmost importance for an information retrieval system to equip with a good graphical user interface that organizes the information into an effective visual structure for the searchers to browse through during the information retrieval process.

A number of researchers have proposed the use of lattice for graphical organization and visualization structuring in the construction of information retrieval systems (Carpineto and Romano 1995, 1996, Cole and Eklund 1996, 1999, Missaoui, Godin and April 1993,

Priss 1997, 2000a, 2000b, 2000c). Lattice is a network-like classification structure that can be generated automatically from a term-document indexing relationship. Such a network structure outperforms hierarchical classification structure since the former enables many paths to a particular node while the latter restricts each node to possess only one parent. Hence lattice navigation provides an alternate browsing-based approach which can overcome the weakness of hierarchical classification browsing. However, the lattice navigation approach has its inherent problem—the curse of dimension, namely, the lattice representation of a document collection is too large to fit in a screen even for small databases. To visualize large structures, researchers have developed interfaces that allow multiple local and global views (Crouch 1990, Wille 1989). However, the method has the disadvantage that the searchers need to map different graphical representation. Carpineto and Romano (1996a, 1996b, 2000) extended the work to adopt a variant of the fisheye view technique (Furnas 1986) to show individual nodes of the lattice on a standalone symbolic lisp machine. Despite these many research efforts, the display and comprehension of the lattice associated with a large database remains an open problem.

We propose in this paper a novel approach which incorporates the advantages of both the Vector Space Model and Concept Lattice to resolve the dimension problem of information retrieval using lattice. We introduce a formal model (the Term-Document Lattice Model) to represent a database as a lattice making use of the notion of formal concept. Then, we apply the idea of quotient lattice to reduce the complexity of a Term-Document Lattice. In the reduction process we use the method of Singular Value Decomposition (SVD) to define the equivalence relation required for the construction of quotient lattices.

The remaining sections are organized as follows. In Section 2 we review the underlying principle of Vector Space Model, SVD, and Concept Lattice. The mathematical model of lattice reduction will be discussed in Section 3. We use an example in Section 4 to show how Singular Value Decomposition can be applied to obtain a reduced Term-Document Lattice for solving the dimension problem in lattice-based information retrieval. In Section 5 the application of lattice reduction to the computation of concept lattices is discussed. In Section 6 we give a conclusion of the present work and suggest some future research directions.

2. Vector space model, SVD and concept lattice

The Vector Space Model (Salton et al. 1975) is well known in information retrieval. Its main idea is to represent the database as an $m \times n$ term-document matrix A (Table 1) and the query an $m \times 1$ vector q . Then matrix analysis can be performed on A and q to explore the relationships between the query and the documents. In particular the angle between the query and a document (both are $m \times 1$ vectors) is computed as a measure of how close is the two objects. Another important extension is the use of Singular Value Decomposition to obtain a low-rank approximation of A before the angle computations. This method is known as Latent Semantic Indexing (LSI) (Deerwester et al. 1990) for tackling the synonymy and polysemy problems.

The elements a_{ij} of the term-document matrix A and the elements of the query vector q may assume only one of the two values 1 and 0, with 1 (respectively 0) indicating the

Table 1. A 4×4 Term-document matrix A .

Term/Document	1	2	3	4
a	1	1	0	0
b	1	0	1	0
c	0	1	0	1
d	0	0	1	1

presence (respectively absence) of the i th term in the j th document (or in the query). Hence A is an incidence matrix. In other cases positive real numbers may be assigned to these elements to reflect the relative importance of a term in a document (or the query).

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 to address the errors or uncertainties associated with the Vector Space Model (due to problems like synonymy and polysemy). The main idea of SVD is to project the very high dimensional documents and query vectors into a low dimensional space. In this new space it is reasoned that the underlying structure of the document collection is revealed thus enhancing retrieval performance.

Concept Lattices stem from Formal Concept Analysis (FCA)—an area of applied mathematics that brings mathematical methods into the field of data analysis and knowledge processing. Formal Concept Analysis was first introduced by Rudolf Wille in 1980 (Davey and Priestley 1990, Ganter and Wille 1996, Grätzer 1998 are our general references). 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 (Ganter and Wille 1996).

Definition 1. A formal context is defined as a triple (G, M, I) consisting of two sets, G and M , and a binary relation $I \subseteq G \times M$. The elements of G and M are called *objects* and *attributes* respectively. I is a relation defined between G and M . To represent an object g is in a relation I with an attribute m , we write gIm or $(g, m) \in I$.

Example 1. The information presented in Table 2 gives a (limited) formal context for a crude classification of people. Here the objects are girl, woman, boy, and man. The attributes are female, juvenile, adult, and male.

Definition 2. Let (G, M, I) be a formal context. A formal concept of (G, M, I) can be defined as an *ordered pair* (A, B) where $A \subseteq G$, $B \subseteq M$, $A' = B$, $B' = A$. We call A the *extent* and B the *intent* of the concept (A, B) .

Definition 3. For $A \subseteq G$, and $B \subseteq M$, we define

$$A' := \{m \in M \mid (g, m) \in I \text{ for all } g \in A\}$$

Table 2. An example of a formal context.

Attribute/Object	Female	Juvenile	Adult	Male
Girl	×	×		
Woman	×		×	
Boy		×		×
Man			×	×

denotes the set of attributes common to all the objects in A . Similarly,

$$B' := \{g \in G \mid (g, m) \in I \text{ for all } m \in B\}$$

denotes the set of objects possessing the attributes in B .

Definition 4. The formal concepts of a given formal context can be ordered by the *generalization-specialization* relation. If (A_1, B_1) and (A_2, B_2) are formal concepts of a formal context, then (A_1, B_1) is called a *subconcept* of (A_2, B_2) if and only if A_1 is a subset of A_2 (or equivalently, if and only if B_2 is a subset of B_1). Then (A_2, B_2) is called a *superconcept* of (A_1, B_1) , and we write $(A_1, B_1) \preceq (A_2, B_2)$, with \preceq denotes the *hierarchical order* (or simply *order*) of the formal concepts. The set of all concepts of (G, M, I) ordered in this way is denoted by $L(G, M, I)$ and is defined as the *concept lattice* of the formal context (G, M, I) .

Graphically, a concept lattice is visualized by a *Hasse diagram* (or line diagram) with *nodes* representing formal concepts and *edges* representing the subconcept-superconcept relations between formal concepts. Concept lattice allows the investigation and interpretation of relationships between concepts, objects, and attributes. Figure 1 is the concept lattice generated from the formal context in Table 2.

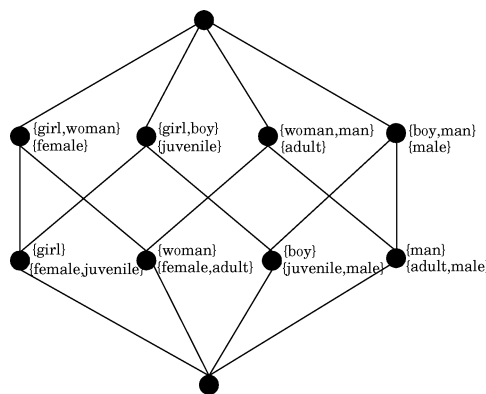


Figure 1. Concept lattice for the formal context in Table 2.

Putting Tables 1 and 2 side-by-side we observe immediately that mathematically a term-document matrix is structurally identical to a formal context if the ones (respectively the zeros) in the former are identified with the crosses (respectively the empty boxes) of the latter. In this case the documents in the term-document matrix become objects and the terms are regarded as attributes. Furthermore, a lattice structure is induced by the term-document matrix when the latter is considered as a formal context. We call such a lattice *Term-Document Lattice (TDL)*. We can draw Hasse diagrams for Term-Document Lattices as concept lattices. Examples of TDLs and their Hasse diagrams are given in the next section.

3. Mathematical model of lattice reduction

We first introduce in this section some mathematical concepts and its application to our problem of information retrieval from term-document matrices. Of particular emphasis are the notions of *congruence* and *quotient* of lattices that will play crucial roles in complexity reduction of our information retrieval process. More specifically, the number of nodes and edges of a lattice could be significantly reduced by introducing an *equivalence relationship* between certain nodes of the lattice. The resulting lattice would be much easy to display and comprehend compared with the original lattice. We then give in Section 3.1 two examples of transforming a lattice to another lattice of reduced complexity to illustrate this basic idea.

Definition 5. Let S be a set. The binary relation \sim on S is said to be an *equivalence relation* on S if for all a, b, c , in S

1. $a \sim a$ (reflexive);
2. $a \sim b$ implies $b \sim a$ (symmetric);
3. $a \sim b$ and $b \sim c$ imply $a \sim c$ (transitive).

Definition 6. Given a set S , and \sim is an equivalence relation on S , the set of all elements of S that are equivalent to a given element x constitute the *equivalence class* or *block* of x , denoted $[x]$. Then $[x] = \{s \in S \mid s \sim x\}$.

Definition 7. An equivalence relations θ on a lattice L which is *compatible with both join and meet* is said to be a *congruence* on L . Since a lattice L is a partial order set of which $a \vee b$ (*a join b*) and $a \wedge b$ (*a meet b*) exist for all $a, b \in L$, where $a \vee b := \sup\{a, b\}$ and $a \wedge b := \inf\{a, b\}$.

As shown in figure 2, we note that a congruence on a lattice L can be indicated on a diagram by placing a loop around the elements in each block of the corresponding partition.

We require that a lattice after reduction is again a lattice. In other words, the reduction process must preserve the algebraic structure of a lattice. If a lattice L is reduced to another lattice K by defining an equivalence relation, the natural map f from L to K must therefore satisfy certain properties.

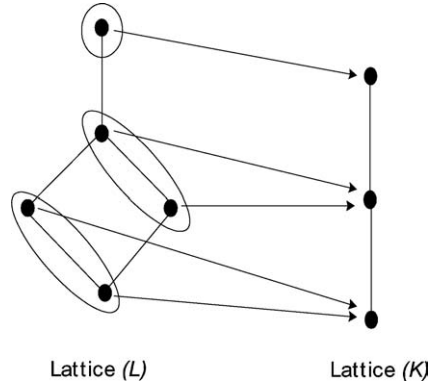


Figure 2. Lattices of congruences.

Definition 8. Let L and K be lattices. A map $f : L \rightarrow K$ is said to be a *homomorphism* (or, for emphasis, *lattice homomorphism*) if f is *join-preserving* and *meet-preserving*, that is, for all $a, b \in L$,

$$f(a \vee b) = f(a) \vee f(b) \quad \text{and} \quad f(a \wedge b) = f(a) \wedge f(b).$$

L and K are *homomorphic* if there exists an homomorphism between L and K .

Definition 9. If L and K are lattices and $f : L \rightarrow K$ is a lattice homomorphism, then the associated congruence θ on L is known as the *kernel* of f and is denoted by $\ker f$. The set of all congruences on L is denoted by $\text{Con}L$.

We say that an equivalence relation θ on a lattice L is compatible with join and meet if for all $a, b, c, d \in L$,

$$a \equiv b(\text{mod } \theta) \quad \text{and} \quad c \equiv d(\text{mod } \theta)$$

imply

$$a \vee c \equiv b \vee d(\text{mod } \theta) \quad \text{and} \quad a \wedge c \equiv b \wedge d(\text{mod } \theta).$$

Figure 3 is an example of homomorphism and its kernels.

Definition 10. Given an equivalence relation θ on a lattice L , we define operations \vee and \wedge on the set

$$L/\theta := \{[a]_\theta \mid a \in L\}$$

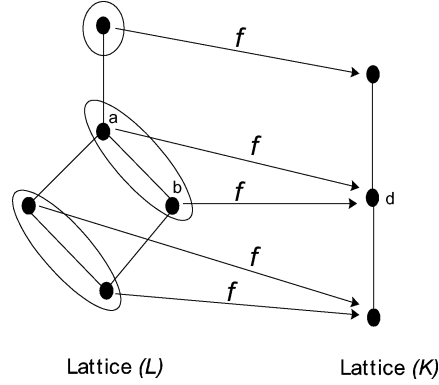


Figure 3. Example of homomorphism and kernels.

of blocks. Namely, for all $a, b \in L$, we define

$$[a]_\theta \vee [b]_\theta := [a \vee b]_\theta \quad \text{and} \quad [a]_\theta \wedge [b]_\theta := [a \wedge b]_\theta.$$

The above operations are well defined, namely, if

$$[a_1]_\theta = [a_2]_\theta \quad \text{and} \quad [b_1]_\theta = [b_2]_\theta$$

then

$$[a_1 \vee b_1]_\theta = [a_2 \vee b_2]_\theta \quad \text{and} \quad [a_1 \wedge b_1]_\theta = [a_2 \wedge b_2]_\theta$$

for all $a_1, a_2, b_1, b_2 \in L$. Since, for all $a_1, a_2 \in L$,

$$[a_1]_\theta = [a_2]_\theta \leftrightarrow a_1 \in [a_2]_\theta \leftrightarrow a_1 \equiv a_2 \pmod{\theta},$$

it follows that \vee and \wedge are well defined on L/θ if and only if θ is a congruence. When θ is a congruence on L , we call $\langle L/\theta; \vee, \wedge \rangle$ the *quotient lattice of L modulo θ* .

3.1. Illustrative example

Given a small-size 4×4 term-document matrix A and its associated lattice L shown in figure 4.

Example 2. We first note that the 4×4 term-document matrix A can be regarded as a formal context. Suppose that the terms a and b are considered as equivalent, we may define a new term a' and a map f such that $f(a) = f(b) = a'$. Furthermore, we obtain a new

4 × 4 Term-Document Matrix (A)

Term / Document	1	2	3	4
a	×	×		
b	×		×	
c		×		×
d			×	×

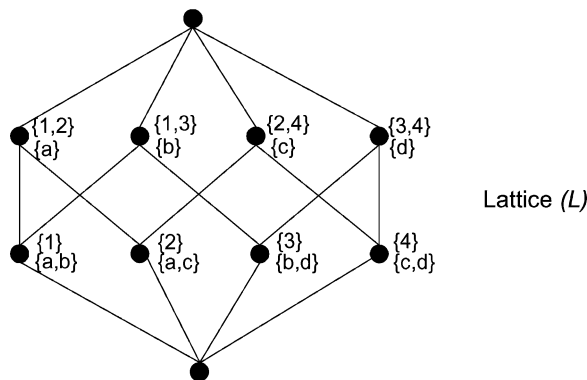


Figure 4. A 4 × 4 term-document matrix and its associated lattice.

formal context from the old one, in which the terms a and b are replaced by a' (Table 3). We see that although Document 2 contains a but not b , it does contain the new term a' because a and b are equivalent. Similarly, Document 3 contains b but not a . However, it contains a' because a and b are equivalent. Note that the number of rows are reduced from four to three. Associated with the new table is a new lattice K_1 as shown in figure 5. We see that the nodes a and b collapse into a single node a' and the complexity of the lattice is reduced since the number of nodes and edges decrease. We also observe that K_1 retains the structure of a lattice. Mathematically, we say that K_1 is “homomorphic” to the original lattice L .

Example 3. In this example we will see a more significant reduction of the complexity of the lattice L . Suppose the terms b and c are now considered as equivalent, then we assign a new term, b' and a map f for this assignment such that $f(b) = f(c) = b'$. Table 4 shows

Table 3. Reduced formal context for Example 2.

Term/Document	1	2	3	4
a'	×	×	×	
c		×		×
d			×	×

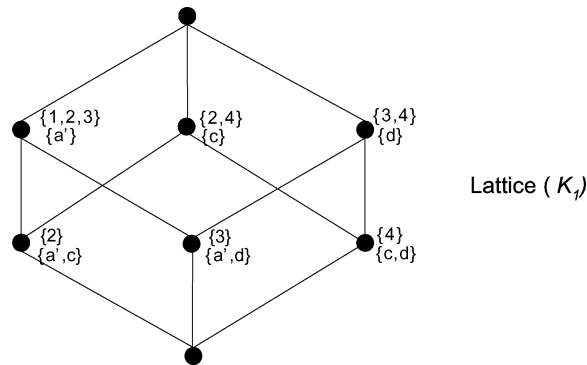


Figure 5. Concept lattice for the reduced formal context in Table 3.

the new formal context. Though Document 2 does not contain b , as b and c are equivalent, Document 2 does contain the new term b' . Similarly, Document 3 contains b but not c . However, it contains b' because b and c are equivalent. Again, the number of rows are reduced from four to three. Figure 6 shows the corresponding new lattice K_2 from which a significant reduction of complexity is observed.

Definition 11. Let C be a formal context representing a term-document matrix and L its associated lattice. We call C_1 a *collapse* of C if C_1 is a formal context and its i th row is obtained from C by taking unions of rows of C .

Table 4. Reduced formal context for Example 3.

Term/Document	1	2	3	4
a	×	×		
b'	×	×	×	×
d			×	×

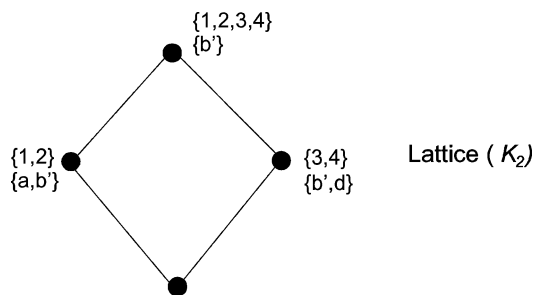


Figure 6. Concept lattice for the reduced formal context in Table 4.

Proposition 1. *Let θ be a congruence on the lattice L . Then $\langle L/\theta; \vee, \wedge \rangle$ is a lattice and the natural quotient map $f : \rightarrow L/\theta$, defined by $f(a) := [a]_\theta$, is a homomorphism.*

Proof: (See Davey and Priestley 1990). □

Proposition 2. *Let C be a formal context, and L its associated lattice. If C_1 is a collapse of C , then there is a lattice K associated with C_1 , where K is homomorphic to L .*

Proof: The process of collapsing C to C_1 clearly induces an equivalence relation θ on L and

$$K = L/\theta := \{[a]_\theta \mid a \in L\}$$

is the quotient lattice of L modulo θ . By Proposition 1, K is homomorphic to L . □

4. Singular value decomposition and quotient lattice

We examine in this section the use of Singular Value Decomposition to define the equivalence relation required for the construction of quotient lattices. Of particular emphasis is the significant reduction of the complexity of the Term-Document Lattice that can be achieved using this construction.

Our central idea is to equate clustering (using SVD) with collapse of rows of a formal context, thus introducing equivalence relations between certain nodes of the Term-Document Lattice. In the parlance of lattices, drawing equivalence relations between nodes of a lattice produces a congruence lattice with lower complexity and hence a reduced Term-Document Lattice. We use in the following a concrete information retrieval example, extracted from Berry and Brown, 1999, to illustrate the idea.

Example 4. In Berry and Brown (1999) the use of SVD to reveal the latent relationship between documents is demonstrated. The original term-document matrix and its associated Term-Document Lattice (figure 7) are shown as follows.

<u>Term</u> ($m = 9$)	<u>Document</u> ($n = 7$)
T1: Bab(y,ies,y's)	D1: <u>Infant</u> & <u>Toddler</u> First Aid
T2: Child(ren's)	D2: <u>Babies</u> & <u>Children's</u> Room (For Your <u>Home</u>)
T3: Guide	D3: <u>Child</u> <u>Safety</u> at <u>Home</u>
T4: Health	D4: Your <u>Baby's</u> <u>Health</u> and <u>Safety</u> : From <u>Infant</u> to <u>Toddler</u>
T5: Home	D5: <u>Baby</u> <u>Proofing</u> Basics
T6: Infant	D6: Your <u>Guide</u> to Easy Rust <u>Proofing</u>
T7: Proofing	D7: Beanie <u>Babies</u> Collector's <u>Guide</u>
T8: Safety	
T9: Toddler	

9 × 7 Term-Document Matrix (A)

Term / Document	1	2	3	4	5	6	7
Bab(y,ies,y's)	0	1	0	1	1	0	1
Child(ren's)	0	1	1	0	0	0	0
Guide	0	0	0	0	0	1	1
Health	0	0	0	1	0	0	0
Home	0	1	1	0	0	0	0
Infant	1	0	0	1	0	0	0
Proofing	0	0	0	0	1	1	0
Safety	0	0	1	1	0	0	0
Toddler	1	0	0	1	0	0	0

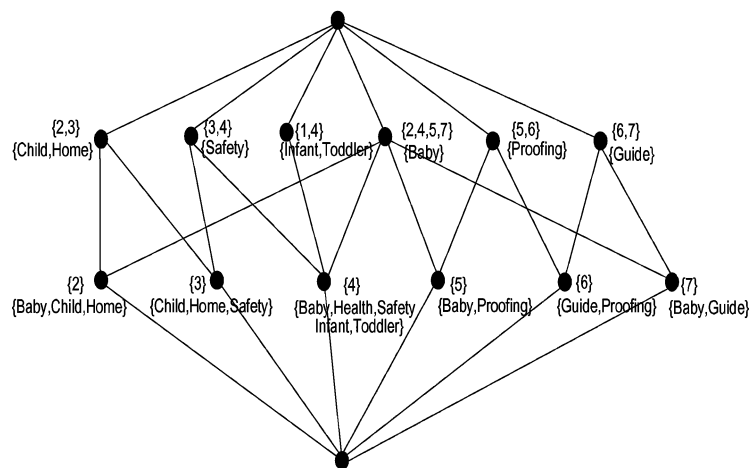


Figure 7. Term-document lattice (L) for the 9 × 7 term-document matrix (A).

After applying SVD to the 9 × 7 term-document matrix, we obtain a simplified formal context (Table 5), and the associated Term-Document Lattice (K) as shown in figure 8.

We see that the nodes with Document 1, 2, 3, and 4 of lattice L collapse into a single node with Document 1'. The complexity of the lattice is reduced, and we obtain a transformed lattice K. As shown in figure 9, the congruences on L for the equivalence relations θ are depicted as shown by placing loops around the respective blocks of elements. We see that L and K are homomorphic with arrows indicating the homomorphisms f . Hence we may write $K = L/\theta$ using the notation of quotient lattice. Note that θ is induced by collapsing columns of the 9 × 7 term-document matrix. We proved in the previous section that collapsing rows of an arbitrary term-document matrix indeed results in an equivalence

Table 5. A simplified formal context after performing SVD.

Term/Document	1'	5	6	7
Bab(y,ies,y's)	×	×		×
Child(ren's)	×			
Guide			×	×
Health	×			
Home	×			
Infant	×			
Proofing		×	×	
Safety	×			
Toddler	×			

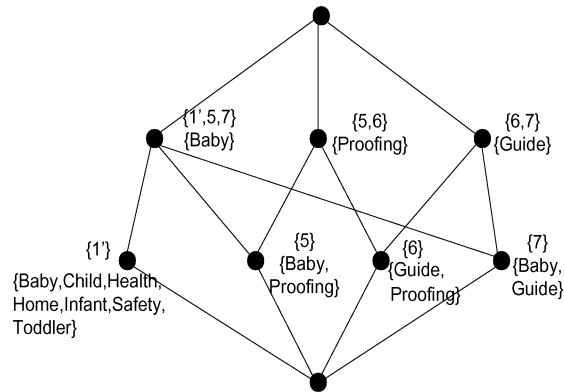


Figure 8. Term-document lattice (K) for the simplified formal context in Table 5.

relation that induces homomorphism, thus providing a concrete basis for our proposed use of low-rank approximation for lattice reduction.

5. Application to the computation of concept lattices

Computing the concept lattice $L(G, M, I)$ from a given context (G, M, I) is a fundamental step of concept analysis. Development of efficient algorithms for this computation is thus vital for successful analysis, and CONCEPTS, LATTICE, and NEXTCONCEPT are some well-known examples. Recently Lindig has studied in detail the effect of context parameters on the computational efficiency of concept lattices (Lindig 2000). It is shown that although the overall lattice structure has some effect on the running time of these algorithms, the dominant factor is the *lattice size* $|L|$, i.e., the total number of concepts. The following figure 10, extracted from (Lindig 2000), illustrates this important observation. It is also found that the *context size* $|I|$ (the number of crosses in a context table) is the primary

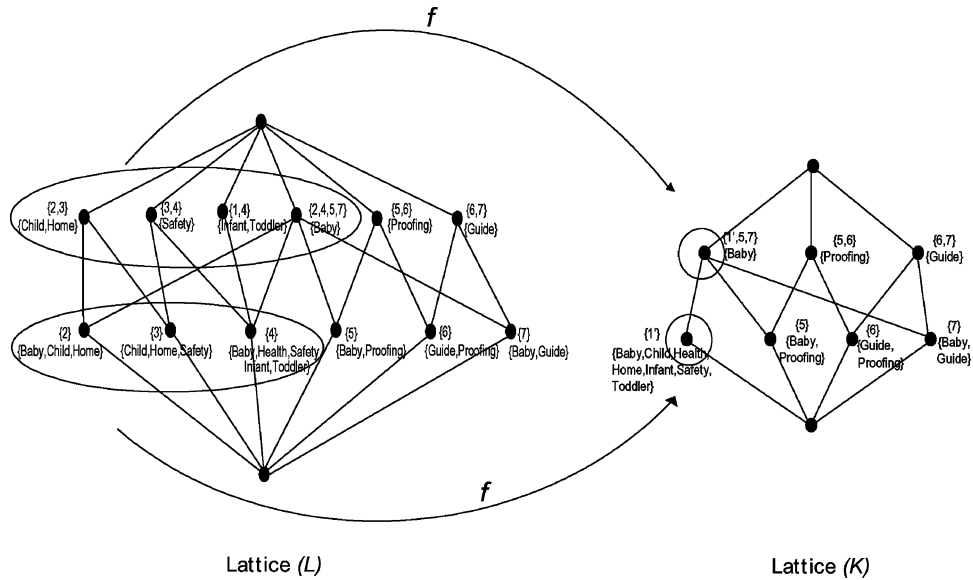


Figure 9. The homomorphism between the original TDL and the reduced TDL.

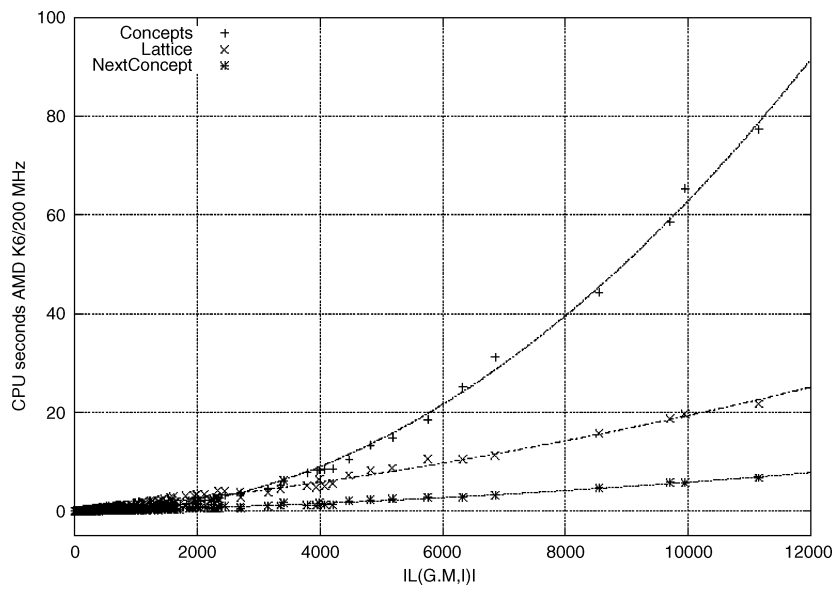


Figure 10. CPU-time versus lattice size $|L(G, M, I)|$.

Table 6. Context size and lattice size of Examples 2 to 4.

	Example 2		Example 3		Example 4	
	Original	Reduced	Original	Reduced	Original	Reduced
Context size $ I $ (Number of crosses)	8	7	8	8	19	13
Lattice size (Number of concepts)	10	8	10	4	14	9

variable that best predicts the lattice size $|L|$ of a lattice. Note that the smaller the $|I|$, the sparsely filled is the context table.

We show by using our previous Examples 2 to 4 that our proposed lattice reduction is relevant to the above consideration of lattice computation. In Table 6, we calculate the context size and the lattice size for each example and observe the noticeable reduction of the latter. The effect can be even more significant for big context size $|I|$ as evidenced by the following estimate of the lattice size proposed by Schütt (1987):

$$|L(G, M, I)| \leq \frac{3}{2} \times 2^{\sqrt{|I|+1}} - 1, \quad |I| > 2.$$

6. Conclusion

We have addressed in the present paper the complexity problem in lattice-based information retrieval. After the introduction of the Term-Document Lattice as the retrieval model we applied the notion of quotient lattice to reduce the complexity of a Term-Document Lattice. The construction of a quotient lattice requires the setting up of an equivalence relation. To obtain this relation we have proposed the application of Singular Value Decomposition to the original term-document matrix. We also discussed the application of lattice reduction to the computation of concept lattices. For future research, a substantial amount of work can be included to extend and make this research more comprehensive. For example, we are now in the process of developing a prototype system to implement our idea of concept lattice dimension reduction. Moreover, a usability study can be conducted to examine and evaluate how lattice reduction will have an impact in interface visualization and information retrieval performance as well.

References

- Berry MW and Brown M (1999) *Understanding Search Engines: Mathematical Modeling and Text Retrieval*. Siam Society for Industrial and Applied Mathematics, Philadelphia.
- Bruza PD and Dennis S (1997) Query reformulation on the internet: empirical data and the hyperindex search engine. In: *Proceedings of the RIAO97 Conference—Computer-Assisted Information Searching on Internet*.
- Bruza PD and McArthur R (2000) Interactive internet search: Keyword, directory and query reformulation mechanisms compared. In: *Proceedings of the 23rd Annual ACM Conference of Research and Development in Information Retrieval (SIGIR '2000)*, ACM Press.

- Carpineto C and Romano G (1995) ULYSSES: A lattice-based multiple interaction strategy retrieval interface. In: Gornostaev J, Blumenthal B and Unger C, Eds., *Lecture Notes in Computer Science—Human-Computer Interaction proceedings of a conference in Berlin in 1995*, Springer-Verlag, Berlin, pp. 91–104.
- Carpineto C and Romano G (1996a) Information retrieval through hybrid navigation of lattice representations. *International Journal of Human-Computer Studies*, 45:553–578.
- Carpineto C and Romano G (1996b) A lattice conceptual clustering system and its application to browsing retrieval. *Machine Learning*, 24:95–122.
- Carpineto C and Romano G (2000) Order-theoretical ranking. *Journal of the American Society for Information Science*, 51(7):587–601.
- Cole R and Eklund P (1996) Application of formal concept analysis to information retrieval using a hierarchically structured thesaurus. In: *Proceedings of a conference in Sydney in 1996*.
- Cole R and Eklund P (1999) Analyzing an email collection using formal concept analysis. In: *Proceedings of a conference on Knowledge and Data Discovery*, No. 1704 in LNAI. Springer-Verlag.
- Crouch CJ (1990) An approach to the automatic construction of global thesauri. *Information Processing and Management*, 26(5):629–640.
- Davey BA and Priestley HA (1990), *Introduction to Lattices and Order*, 2nd ed. Cambridge University Press, Cambridge.
- Furnas G (1986) Generalized fisheye views. In: *Proceedings of ACM CHI '86 Conference on Human Factors in Computing Systems*. Boston, MA, pp. 16–23.
- Ganter B and Wille R (1996) *Formal Concept Analysis: Mathematical Foundations*. Springer-Verlag, Berlin.
- Grätzer G (1998) *General Lattice Theory*. Birkhauser Verlag, Berlin, Germany.
- Lin X (1997) Map displays for information retrieval. *Journal of the American Society of Information Science*, 48:40–54.
- Lindig C (2000) Fast concept analysis. In: Stumme G, Ed., *Working with Conceptual Structures—Contributions to ICCS 2000*, Springer-Verlag, Aachen, Germany.
- McArthur R and Bruza PD (2000) The ranking of query refinements in interactive web-based retrieval. In: *Proceedings of the Information Doors Workshop (Held in conjunction with the ACM Hypertext and Digital Libraries Conferences)*.
- Missaoui R, Godin R and April A (1993) Experimental comparison of navigation in a Golis lattice with conventional information retrieval methods. *International Journal of Man-Machine Studies*, 38:747–767.
- Priss U (1997) A graphical interface for document retrieval based on formal concept analysis. In: Santo E, Ed., *AAAI Technical Report CF-97-01*, pp. 66–70.
- Priss U (2000a) Faceted information representation. In: Stumme G, Ed., *Working with Conceptual Structures [proceedings of a conference in Aachen in 2000]*. Springer-Verlag, Berlin, pp. 84–94.
- Priss U (2000b) Knowledge discovery in databases using formal concept analysis. *Bulletin of the American Society for Information Science*, October/November:18–20.
- Priss U (2000c) Lattice-based information retrieval. *Knowledge Organization*, 27(3):132–142.
- Schütt D (1987) *Abschätzungen für die Anzahl der Begriffe von Kontexten*. Diplomarbeit, DH Darmstadt.
- Wille R (1989) Lattices and data analysis: How to draw them using a computer. In: Rival I, Eds., *Algorithms and Order*. Kluwer, Boston, pp. 33–58.

