Scalable conceptual hierarchy based algorithm for knowledge sharing in digital ecosystem

Huaiguo Fu
Telecommunications Software & Systems Group
Waterford Institute of Technology, Waterford, Ireland
hfu@tssg.org

Abstract

Knowledge sharing between participants of the ecosystem is one core of the digital ecosystem. This paper proposes a framework of distributed knowledge sharing in digital ecosystem and a scalable algorithm for knowledge sharing in digital ecosystem. The algorithm is based on concept lattice that is a popular mathematical structure for modeling conceptual hierarchies. The algorithm can decompose the search space of formal concepts into independent non-overlapping subspaces that can be extracted independently to generate formal concepts. The experimental results show the algorithm is efficient to extract and present formal concepts in large data.

Keywords: *Formal concept analysis, Concept lattice, Knowledge sharing, Digital ecosystem, Algorithm*

1 Introduction

In the natural world, an ecosystem is a system whose members benefit from each other's participation via symbiotic relationships. In digital environment, a sharing collaborative system is formed by the digital components such as digital data, information and knowledge. Collaborative knowledge sharing between participants of the ecosystem is one core of the digital ecosystem. The digital ecosystem should be an effective platform to enable the participants of the ecosystem to learn and share divers knowledge such as domain knowledge, skills, expertise, useful information, experiences and perspectives from others.

Knowledge is the high level digital component. Knowledge is useful and appropriate collection of information, and understanding patterns. In the digital ecosystem, some knowledge can be existing, but some knowledge can be previously unknown, implicit, hidden in large data, and should be derived from data. We therefore should extract the knowledge from large data with the techniques such as data mining or KDD (Knowledge Discovery in Databases) [16, 4, 1]. The techniques of data mining are widely used in research and application to look for relationships and knowledge that are implicit in large volumes of data and are interesting in the sense of impacting an organization's practice. For example, data mining techniques can help companies to provide better, customized services and support decision making. Hence, the techniques of data mining can be used to extract knowledge for the digital ecosystem.

Concept is the base of the knowledge representation and an atomic knowledge unit. Formal concept should be a basic knowledge unit in the digital ecosystem. Formal concept analysis (FCA) [20] is an efficient tool for conceptual analysis and data mining [21, 12, 13, 19, 22]. FCA is different from some of the traditional, statistical means of data analysis and knowledge representation because of its focus on human-centered approaches. From the formal concepts, we can analyze data such as revealing stronger association or relation between attribute set and the set of their common objects, classifying objects, generating implications of attributes or knowledge rules, extracting the hierarchical relation among formal concepts, etc. The core of FCA is concept lattice. Theoretical foundation of concept lattice is derived from the mathematical lattice theory [2, 7] that is a popular mathematical structure for modeling conceptual hierarchies. Concept lattice also provides an effective tool of knowledge visualization. Concept lattice facilitates exploring, searching, recognizing, identifying, analyzing, visualizing, restructuring and memorizing conceptual structures [17]. The application of concept lattice has been an area of active and promising research in various fields such as knowledge discovery, information retrieval, software engineering and machine learning [10, 9] and bioinformatics [11, 15, 14, 18].

In this paper, we propose a framework of distributed knowledge sharing in digital ecosystem. We also propose a scalable lattice-based algorithm to discover hierarchical concepts in large and high-dimensional data for knowledge sharing in digital ecosystem. The algorithm is based on the density of attribute and intent of concept, and the hierarchical order between the concepts. The algorithm partitions the search space of formal concepts into independent non-overlapping subspaces that can be extracted independently to generate formal concepts. The concepts can be used to explore, present and interpret the knowledge from the large data in digital ecosystem so that facilitates knowledge sharing in digital ecosystem.

Contributions. This paper makes the following contributions:

- Propose a framework of distributed knowledge sharing in digital ecosystem.
- Propose a scalable lattice-based algorithm to extract hierarchical concepts in large and high-dimensional data for knowledge sharing in digital ecosystem.

The rest of this paper is organized as follows. The framework of distributed knowledge sharing in digital ecosystem is presented in the next section. Section 3 introduces the basic concepts of concept lattice. Section 4 analyzes the search space of formal concept and discuss how to partition the search space of concept. The new algorithm of mining concepts is introduced in section 5. We show the experimental results of the algorithm in section 6. The paper ends with a short conclusion in section 7.

2 Framework for collaborative knowledge sharing

In the digital ecosystem, the first challenge is "what knowledge we can share in the digital ecosystem". Digital ecosystem is an open, demand-driven, self-organizing collaborative environment. When we need to share some knowledge, the system should provide the inherent or existing knowledge, or the system can investigate, compare, summarize, merge, decompose and analyze the relevant data, information or knowledge to extract the required knowledge from data if the knowledge is previously unknown, implicit, hidden in large data. The system should teach the participant to learn and understand the knowledge.

How to share knowledge is another challenge for collaborative knowledge sharing. We need to develop efficient, dynamic, flexible and scalable models and services for distributed knowledge sharing in the digital ecosystems.

We introduce a framework (see figure 1) for collaborative knowledge sharing between participants of the ecosystem. The framework has the following features:

- Address the challenges for collaborative knowledge sharing between participants of the ecosystem;
- Analyze the environment of sharing knowledge and propose new platforms of sharing distributed knowledge for digital ecosystems: community networks, organization networks and digital ecosystem networks. The networks can be virtual networks. The sharing platforms can integrate all digital resource for knowledge sharing. For example, the platforms can provide the behaviours recorded through distributed accounting to facilitate collaboration through the trading of information using community currencies in a manner that optimises the operation of the ecosystem as a whole;
- Describe the knowledge stream in the framework and propose to use Sharing Requirement to control the knowledge stream with eco-features.

Knowledge is the core of the system of for collaborative knowledge sharing. Knowledge discovery, knowledge generation, knowledge collection, knowledge transformation, knowledge distribution, and knowledge sharing constitute a cycle of knowledge sharing. We propose a framework for distributed knowledge sharing (see figure 1) to control the knowledge stream in the cycle in the digital ecosystems and facilitate collaborative knowledge sharing.

The framework includes 6 parts: Infrastructure, Sharing platforms, Sharing requirement, Knowledge resource, Knowledge engine and Knowledge Sharing. The core of the framework is sharing platforms that provide the digital eco-environment of collaborative knowledge sharing. We propose the integration of community networks, organization networks and digital ecosystem networks three overlapped platforms for sharing distributed knowledge in digital ecosystems. The sharing platforms contain all digital resource in the digital ecosystems so that the platforms can afford the knowledge resource including various data, information and knowledge. Especially, the sharing requirements are derived from the sharing platforms.

Knowledge engine is for acquirement of knowledge that responds to where and how to find knowledge. Knowledge Sharing is for sharing knowledge in digital ecosystem. It focuses on the answer to how to share knowledge.

We introduce the main parts of the framework in the following sections.

2.1 Sharing requirements

The digital ecosystem is an user-driven and demand-driven system. The source of sharing knowledge is sharing need. The digital ecosystem and participants of the ecosystem all can produce the sharing requirements. The requirements from participants of the ecosystem are specific sharing requirements. The features of ecosystem, self-adapting, self-organizing and self-management of the digital ecosystem can enable the sharing environment to automatically generate sharing requirements to recommend

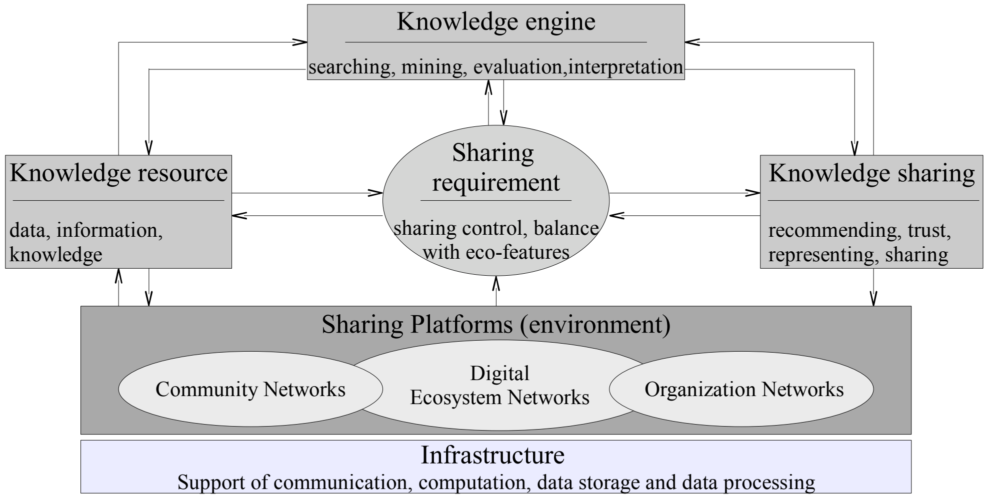

Figure 1. Framework of collaborative knowledge sharing

sharing knowledge to the participants of the ecosystem.

Sharing requirement service can analyze, interpret and transform the different needs, organize Knowledge resource and Knowledge engine to provide required knowledge, and deploy Knowledge distributing and sharing. Sharing requirement service will control and balance other services to adapt the sharing task according to the eco-features and the features of sharing environment.

2.2 Sharing environment

The sharing environment is constituted by three different and associated platforms: community networks, organization networks and digital ecosystem networks. The sharing environment is the base of knowledge sharing and it facilitates to provide efficient, dynamic, flexible and scalable models and services for distributed knowledge sharing in the digital ecosystems.

The organization networks provide a platform for local knowledge sharing, mapping and storage by one organization, e.g. one enterprise. The community networks overlap with organization networks to share distributed knowledge by common topics for a community, e.g. OKS research group. The digital ecosystem networks provide more wide, open and dynamic knowledge sharing space by the integration of community networks, organization networks and other knowledge that are not included in community networks and organization networks.

The sharing platforms provide an open, self-adapting, self-organizing environment with the following services:

- Knowledge organization, classification and clustering
- Dynamic distributed knowledge indexing
- Dynamic description of knowledge with keywords, abstract, links and references
- Embedded engine of data mining
- Web-based searching and extracting

2.3 Knowledge engine

Knowledge engine is for acquirement of knowledge that responds to where and how to find knowledge. Knowledge engine provides methods of knowledge searching if the knowledge is already existed, and the techniques and algorithms of data mining if knowledge is implicit in data. When knowledge is acquired, Knowledge engine will provide the services of evaluation, interpretation for knowledge. Knowledge engine also contains learning service to learn to adapt the environment and acquire and understand sharing knowledge.

2.4 Knowledge sharing

Knowledge Sharing focuses on the answer to how to share knowledge. Knowledge Sharing will provide the methods, models and strategies to share distributed knowledge. Knowledge Sharing also has the services to trust, accept, store, distribute and update knowledge in digital ecosystem.

2.5 Infrastructure

In this framework, the basic infrastructure is Peer-to-peer (or abbreviated P2P) architecture that is a type of network in which each workstation has equivalent capabilities and

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
1	×	×					×	
2	×	×					×	×
3	×	×	×				×	×
4	×		×				×	×
5	×	×		×		×		
6	×	×	×	×		×		
7	×		×	×	×			
8	×		×	×		×		

Figure 2. Example of formal context

responsibilities. This differs from client/server architectures. Generally, P2P networks are used for sharing files, but a P2P network can also mean Grid Computing. The infrastructure of P2P provides a novel distributed environment, computational model, and unprecedented opportunities for unlimited computing and storage resources. It's distinguished from conventional distributed computing by its focus on large-scale resource sharing, innovative applications, and, in some cases, high-performance orientation. Grids can be used as effective infrastructures for distributed high-performance computing and data processing. P2P environment provides high performance computing facilities and transparent access to them in spite of their remote location, different administrative domains and hardware and software heterogeneous characteristics. In this framework, P2P provide the support of communication, computation, data storage and data processing for distributed knowledge sharing in digital ecosystem.

3 Concept lattice

In this section, we introduce some basic notions of concept lattice.

Definition 3.1 **Formal context** *is defined by a triple* (O, A, R)*, where* O *and* A *are two sets, and* R *is a relation between* O *and* A*. The elements of* O *are called objects, while the elements of* A *are called attributes.*

For example, Figure 2 represents a formal context (O, A, R). $O = \{1, 2, 3, 4, 5, 6, 7, 8\}$ is the set of objects, and $A = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8\}$ is the set of attributes. The crosses in the table describe the relation R of O and A.

A formal context is usually represented by the binary data, but in practice, the values of attribute are not binary, we can transform many-valued data context to binary values context by concept scaling [7].

Definition 3.2 *Two* **closure operators** *are defined as* $O_1 \to O_1''$ *for set* O *and* $A_1 \to A_1''$ *for set* A*.*

$$O_1' := \{a \in A \mid oRa \text{ for all } o \in O_1\}$$

$$A_1' := \{o \in O \mid oRa \text{ for all } a \in A_1\}$$

These two operators are called the **Galois connection** for (O, A, R). These operators are used to determine a formal concept.

Definition 3.3 *A* **formal concept** *of* (O, A, R) *is a pair* (O_1, A_1) *with* $O_1 \subseteq O$*,* $A_1 \subseteq A, O_1 = A_1'$ *and* $A_1 = O_1'$*.* O_1 *is called extent,* A_1 *is called intent.*

Definition 3.4 *We say that there is a hierarchical order between two formal concepts* (O_1, A_1) *and* (O_2, A_2)*, if* $O_1 \subseteq O_2$ *(or* $A_2 \subseteq A_1$*). And* (O_1, A_1) *is called the* **sub-concept** *of* (O_2, A_2)*, or* (O_2, A_2) *is called the* **super-concept** *of* (O_1, A_1)*, if there is no concept* (O_3, A_3)*,* $A_2 \subseteq A_3 \subseteq A_1$ *or* $O_1 \subseteq O_3 \subseteq O_2$*.*

All formal concepts with the hierarchical order of concepts form a complete lattice called **concept lattice**.

The concept lattice of the formal context of Figure 2 is presented in Figure 3.

Definition 3.5 *Given a formal context* (O, A, R)*,* (O_1, A_1) *is a concept, the number of objects of* (O_1, A_1) *is called the* **density** *of* (O_1, A_1)*.*

4 Partitioning search space of concepts

In this section, we propose an approach to determine a concept in a formal context and then study the hierarchical order between each concept and its sub-concepts to analyze how to partition the search space of concepts.

Definition 4.1 *Given a context* (O, A, R)*, an attribute* a_i *is called* **maximal attribute***, if* $a_i \in A$*, for all* $a_j \in A$*,* $i \neq j$ *and* $\{a_i\}' \not\subset \{a_j\}'$ *.*

It is easy to infer the following proposition from the definition 4.1.

Proposition 4.1 *Given a context* (O, A, R)*, if attribute* $a_i \in A$ *and* a_i *is a maximal attribute, and for all* $a_j \in A$*,* $i \neq j$ *and* $\{a_i\}' \neq \{a_j\}'$*, then* $\{a_i\}$ *is the intent of concept* $(\{a_i\}', \{a_i\})$*.*

From the proposition 4.1, we have a simple approach to determine a concept in the formal context (O, A, R):

- Merge the attributes that contain the same objects as a single attribute;
- Find the maximal attributes in the merged context;

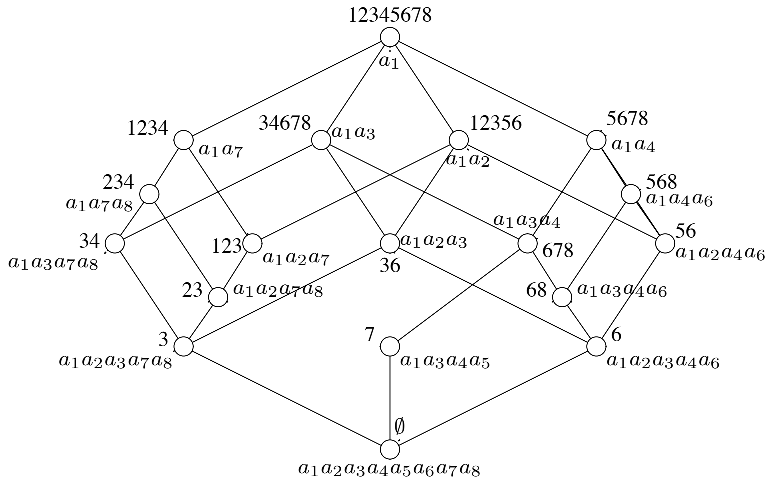

Figure 3. Example of concept lattice

- Each maximal attribute must be an intent of concept.

Definition 4.2 *Given a formal context (O, A, R), (O_1, A_1) and (O_2, A_2) are two concepts. We say (O_2, A_2) is more dense than (O_1, A_1) if the density of $(O_1, A_1) <$ the density of (O_2, A_2).*

There are different methods to partition the search space of concepts. We will explore our method by the following objectives:

- The concepts more dense can be generated preferentially;
- The approach to determine a concept can be used in subspaces.

Proposition 4.2 *Given a concept (A_i', A_i) of context (O, A, R) and its sub-concept (A_{ij}', A_{ij}) where $j = 1, 2, 3, ..., ((A_{ij} - A_i)', A_{ij} - A_i)$ is a concept in sub-context $({A_i}', A - A_i, R)$.*

Proof : A_{ij} is an intent of sub-concept of (A_i', A_i), then we have $A_{ij} - A_i = (A_{ij} - A_i)''$ in the sub-context $({A_i}', A - A_i, R)$. Thus, $A_{ij} - A_i$ is an intent of concept in the sub-context $({A_i}', A - A_i, R)$. So $((A_{ij} - A_i)', A_{ij} - A_i)$ is a concept in sub-context $({A_i}', A - A_i, R)$.

Proposition 4.3 *All sub-concepts of $({A_i}', A_i)$ can be generated from the sub-context $({A_i}', A - A_i, R)$.*

Proof : Given an concept (A_j', A_j) of the sub-context $({A_i}', A - A_i, R)$, $(A_j \cup A_i)$ is an intent of sub-concept of $({A_i}', A_i)$. Thus all sub-concepts can be generated by the intent of concepts of the sub-context $({A_i}', A - A_i, R)$.

From above propositions, we can consider sub-contexts as the subspaces of concepts. The sub-context can be $({A_i}', A - A_i, R)$, $({A_i}', A_i)$ is one super-concept. In each sub-context or subspace, the maximal attributes or attributes with high density can generate intent of concepts. For example (see figure 4), a_2, a_3, a_4 and a_7 are maximal attributes in the sub-context of a_1 and they can generate the intent of concepts: a_1a_2, a_1a_3, a_1a_4 and a_1a_7. Then the sub-concepts of (a_1', a_1)=(12345678, a_1) are generated: (12356, a_1a_2), (34678, a_1a_3), (5678, a_1a_4) and (1234, a_1a_7).

Definition 4.3 *Given an attribute a_i of context (O, A, R), $(\{a_i\}', A - \{a_i\}'', R)$ is called* **projective sub-context** *of a_i.*

We analyze the projective sub-contexts of a_1a_2 and a_1a_3 (see figure 5 and figure 6).

In sub-contexts of a_1a_2 and a_1a_3, we can find that (36, $a_1a_2a_3$) is a common concept for two sub-contexts because the sub-contexts are overlapping. To reduce the redundant concepts, we do not change the projective sub-context of a_1a_2, but remove a_2 from the projective sub-contexts of a_1a_3 to reduce the sub-context of a_1a_3. Thus, all the concepts that include a_2 in the sub-context of a_1a_3 will not be generated.

	a_2	a_3	a_4	a_5	a_6	a_7	a_8
1	×					×	
2	×					×	×
3	×	×				×	×
4		×				×	×
5	×		×		×		
6	×	×	×		×		
7		×	×	×			
8		×	×		×		

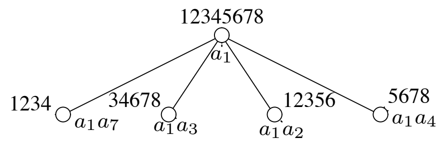

Figure 4. Example of sub-context and sub-concepts of (a_1', a_1)

sub-context of a_1a_2:
$(\{a_1a_2\}', A - \{a_1a_2\}, R)$

	a_3	a_4	a_5	a_6	a_7	a_8
1					×	
2					×	×
3	×				×	×
5		×		×		
6	×	×		×		

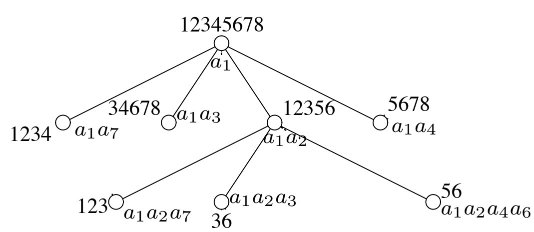

Figure 5. Example of sub-context and sub-concepts of a_1a_2

The summary of the main idea of partitioning the search space is the reduced projective sub-contexts of high dense concepts from top concept are the subspaces of concepts.

sub-context of a_1a_3:
$(\{a_1a_3\}', A - \{a_1a_3\}, R)$

	a_2	a_4	a_5	a_6	a_7	a_8
3	×				×	×
4					×	×
6	×	×		×		
7		×	×			
8		×		×		

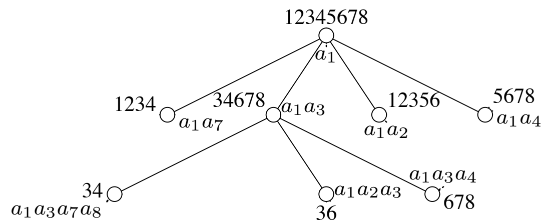

Figure 6. Example of sub-context and sub-concepts of a_1a_3

5 Discovering concepts

Given a formal context, we propose a scalable algorithm to generate concepts by following steps:

1) First of all, the algorithm needs to generate the ordered context.

Definition 5.1 *A formal context is called* **ordered context** *if we order the attributes of formal context by number of objects of each attribute from the smallest to the biggest one, and the attributes with the same objects are merged as one attribute. We note ordered context* $(O, A^{\triangleleft}, R)$ *of the formal context* (O, A, R).

Ordered context can count the density of each attribute and facilitate the generation of maximal attributes and reduced sub-contexts. Ordered context allows to generate concepts according to dense priority: the concept more dense will be generated preferentially.

2) The second step: from the ordered context, every maximal attribute forms an intent of concept of the first level.

We have the following proposition by the definition 5.1 and the proposition 4.1.

Proposition 5.1 *Each maximal attribute of the ordered data context* $(O, A^{\triangleleft}, R)$ *is an intent of concept.*

For example, the concept in first level for the formal context of figure 2 is a_1. The density of (a_1', a_1) is 8.

This step is the core of the algorithm. For any sub-context, we only see about the maximal attributes that can form the concept.

3) The third step: determine the number of partitions according to the size and dimension of data, or the user's requirement. We can generate more partitions if data is large or high-dimensional. The partitions are independent to generate concepts. The partition is the reduced sub-context. The partitions can be generated from the concepts of first level to the next level if we need.

4)The fourth step: in each partition, we will generate the concepts from the reduced sub-context to all sub-contexts.

5) At the end, we can get all concepts from all partitions.

For example, the algorithm generates concepts (see figure 3) from the formal context of figure 2. In the first, we generate the ordered context. Then we will partition the search space of concept into 4 subspaces.

a_1 is maximal attribute in the ordered context so $\{a_1\}$ is an intent of concept. All sub-concepts can be generated from the sub-context in figure 4. The sub-contexts $(\{a_1a_2\}', A-\{a_1a_2\}, R), (\{a_1a_3\}', A-\{a_1a_3\}-\{a_2\}, R)$, $(\{a_1a_4\}', A-\{a_1a_4\}-\{a_2\}-\{a_3\}, R)$ and $(\{a_1a_7\}', A-\{a_1a_7\} -\{ a_2\} -\{ a_3\}, R)$ are 4 subspaces of searching concepts. The 4 subspaces are independent. We can generate all concepts in each subspace. For example, in sub-contexts $(\{a_1a_2\}', A-\{a_1a_2\}, R)$, we can find intents: a_3, a_7, a_7a_8, a_4a_6. In whole context, we need to add $\{a_1a_2\}$ to each intent in sub-context to form the intent of whole context. Therefore, $a_1a_2a_3$, $a_1a_2a_7$, $a_1a_2a_7a_8$, $a_1a_2a_4a_6$ are intents of the whole context. As we mine the intent of concepts from the reduced sub-contexts, 7 redundant intents are avoided to be repeatedly mined in different sub-contexts.

6 Experimental results

We have implemented the algorithm in Java to generate concepts. We test the algorithm in some real data and simulation data. We compare the partitioning algorithm with NextClosure algorithm [6]. Experimental comparisons of lattice-based algorithms show that NextClosure algorithm is one of the best for large and dense data [8, 5].

The preliminary experimental results in figure 7 show the efficiency of the algorithm. In the the experimental results, the run time of partitioning algorithm is the total time of all subspaces mining. The experimental results show the partitioning algorithm is much faster than NextClosure algorithm.

Real data (see table 1) for our experiment comes from machine learning benchmarks: UCI repository [3].

The algorithm is easy to be used to extract some interesting concepts according to the density of the concepts such as big concepts or small concepts. For example, the figure 8 illustrates the run time to generate concepts with different density on the dataset: dermatogogy.

7 Conclusion and further work

In this paper, we proposed a framework of distributed knowledge sharing in digital ecosystem and developed a scalable algorithm to generate hierarchical formal concepts from large data in digital ecosystem. The algorithm can partition searching space of concepts into independent reduced subspaces and then extract concepts in each subspace. The experimental results show the algorithm is efficient to generate formal concepts in large data.

The future work will focus on development of distributed algorithms to analyze huge and heterogeneous distributed data for distributed knowledge sharing in digital ecosystem.

Acknowledgements

This work is supported by the PRTLI project of Higher Education Authority (HEA), Ireland and the project of EU IST Network of Excellence ”OPAALS”.

References

[1] P. Adriaans and D. Zantinge. *Data mining*. Addison-Wesley, 1997.

[2] G. Birkhoff. *Lattice Theory*. American Mathematical Society, Providence, RI, 3rd edition, 1967.

[3] C. Blake, E. Keogh, and C. Merz. UCI repository of machine learning databases, 1998. http://www.ics.uci.edu/∼mlearn/MLRepository.html.

[4] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors. *Advances in Knowledge Discovery and Data Mining*. AAAI/MIT Press, 1996.

[5] H. Fu and E. Mephu Nguifo. How well go lattice algorithms on currently used machine learning testbeds? In *ICFCA 2003, First International Conference on Formal Concept Analysis*, 2003. Also in EGC04, p. 373-384.

[6] B. Ganter. Two basic algorithms in concept analysis. Technical Report 831, Technische Hochschule, Darmstadt, Germany, 1984. preprint.

[7] B. Ganter and R. Wille. *Formal Concept Analysis. Mathematical Foundations*. Springer, 1999.

[8] S. Kuznetsov and S. Obiedkov. Comparing performance of algorithms for generating concept lattices. *Special Issue on Concept Lattice for KDD of JETAI*, 14(2/3):189–216, 2002.

[9] E. Mephu Nguifo, V. Duquenne, and M. Liquière. *Journal of Applied Artificial Intelligence (AAI) Special Issue on Concept Lattice-based applications for Knowledge Discovery in Databases*. Taylor and Francis, 2003.

[10] E. Mephu Nguifo, M. Liquiere, and V. Duquenne. *JETAI Special Issue on Concept Lattice for KDD*. Taylor and Francis, 2002.

DataSet	ID	Objects	Attributes	Closed item-sets
soybean-small	d1	47	79	3253
car	d2	1728	21	7999
breast-cancer-wisconsin	d3	699	110	9860
house-votes-84	d4	435	18	10642
audiology.standardized	d5	26	110	30401
tic-tac-toe	d6	958	29	59503
nursery	d7	12960	31	147577
lung-cancer	d8	32	228	186092
agaricus-lepiota	d9	8124	124	227594
promoters	d10	106	228	304385
soybean-large	d11	307	133	806030
dermatogogy	d12	366	130	1484088

Table 1. The datasets of real data for experiment

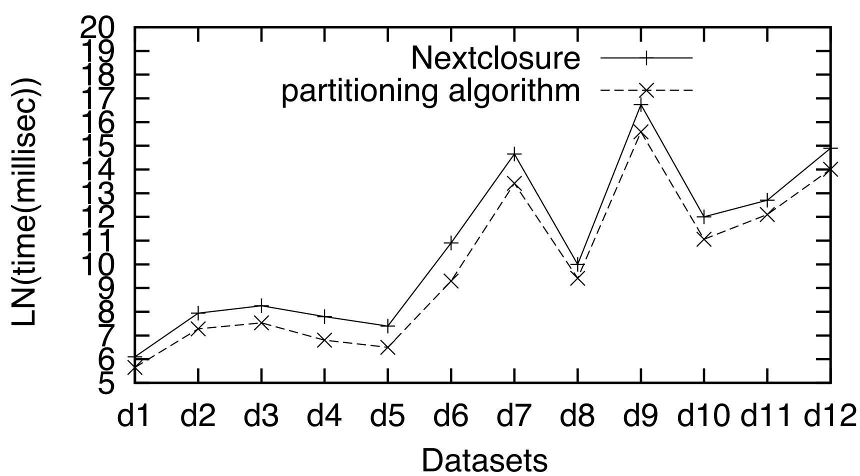

Figure 7. Experimental comparison of partitioning algorithm and NextClosure algorithm

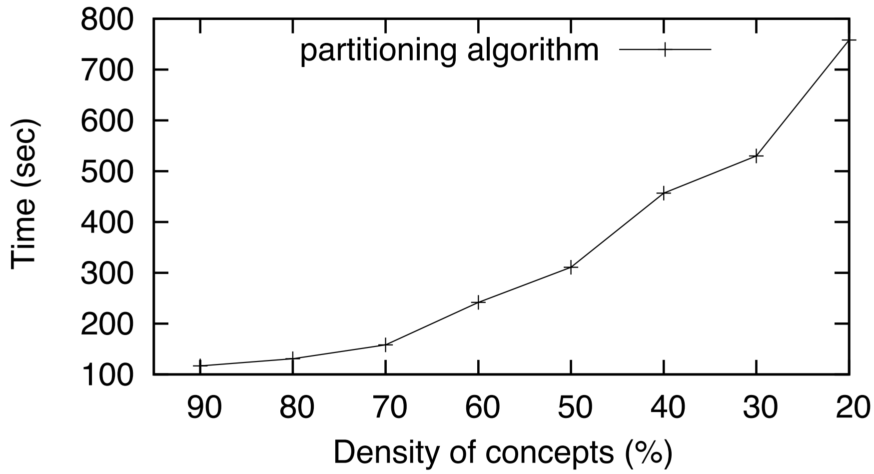

Figure 8. Concepts generation with different density

[11] F. Pan, G. Cong, A. K. H. Tung, J. Yang, and M. J. Zaki. Carpenter: finding closed patterns in long biological datasets. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 637–642, New York, NY, USA, 2003. ACM Press.

[12] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Efficient mining of association rules using closed itemsets lattices. *Journal of Information Systems*, 24(1):25–46, 1999.

[13] J. Pei, J. Han, and R. Mao. CLOSET: An efficient algorithm for mining frequent closed itemsets. In *ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, pages 21–30, 2000.

[14] J. Pfaltz and C. Taylor. Closed set mining of biological data. In J. T. L. W. Mohammed J. Zahi and H. T. T. Toivonen, editors, *Proceedings of the 2nd ACM SIGKDD Workshop on Data Mining in Bioinformatics*, 2002.

[15] T. H. Pham, K. Satou, and T. B. Ho. Mining yeast transcriptional regulatory modules from factor dna-binding sites and gene expression data. *Genome Informatics*, 15(2):287–295, 2004.

[16] G. Piatetsky-Shapiro and W. J. Frawley, editors. *Knowledge Discovery in Databases*. AAAI/MIT Press, 1991.

[17] U. Priss. Formal concept analysis in information science. *Cronin, Blaise (ed.),Annual Review of Information Science and Technology*, 40:521–543, 0062.

[18] F. Rioult, J. Boulicaut, B. Crémilleux, and J. Besson. Using transposition for pattern discovery from microarray data. In *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, pages 73–79. ACM Press, 2003.

[19] J. Wang, J. Han, and J. Pei. Closet+: Searching for the best strategies for mining frequent closed itemsets. In *In Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'03)*, Washington, DC, USA, 2003.

[20] R. Wille. Restructuring lattice theory: an approach based on hierarchies of concepts. In I. Rival, editor, *Ordered Sets*, pages 445–470. D. Reidel, 1982.

[21] R. Wille. Why can conceptual lattices support konwledge discovery in databases ? In E. M. Nguifo, V. Duquenne, and M. Liquiere, editors, *International Workshop on Concept Lattices-based KDD*, ICCS'01, pages 7–20, Stanford University, Palo Alto, July 2001.

[22] M. J. Zaki and C. Hsiao. CHARM: An efficient algorithm for closed itemset mining. Technical Report 99-10, Rensselaer Polytechnic Institute, 1999.