A conceptual subspace clustering algorithm in e-learning

Huaiguo Fu, Mícheál Ó Foghlú
Telecommunications Software & Systems Group
Waterford Institute of Technology, Waterford, Ireland
{hfu, mofoghlu}@tssg.org

Abstract

In recent years, due to large amounts of network-based teaching and learning data continue to grow inexorably in size and complexity, knowledge clustering becomes more important in e-learning. This paper proposes a novel algorithm of cluster analysis to extract clusters in dense subspaces and the clusters can be described by overlapping hierarchical concepts. The experimental results show the algorithm is efficient to extract conceptual clusters in large data.

Keywords: *Cluster analysis, Conceptual clustering, Subspace clustering, Concept lattice, Algorithm*

1 Introduction

Information searching on the Web is important in e-learning. But the searching results are often mass and confused for users. Users often get too much information by Web search engines and they have to sift through so long ordered list of information that they waste too much time and can not find the exact information that they need. Some techniques such as clustering can be used for information searching and post-searching to categorize information, clarify and represent knowledge to users, and users can easily locate the desired information [14]. Especially, the techniques of hierarchical clustering can help us to generate the topic hierarchy for similar documents so that users are able to capture the "meaning" of a set of information or searching results.

Cluster analysis is unsupervised learning and one of the primary tasks of data mining. There are divers algorithms and approaches of cluster analysis [8, 13]. Due to large amounts of data continue to grow inexorably in size and complexity, the techniques and approaches of cluster analysis suffer from the challenges such as high-dimensional data clustering, complex data clustering and description of too many clusters. The purpose of data clustering is to extract the similarity between objects to gain better understanding of the data. However, description and interpretation of clusters of data is one of main issues of applications of the most clustering techniques, especially clustering often obtains large numbers of clusters. We investigate these problems and propose the integration of techniques of subspace clustering and conceptual clustering to address these challenges.

Subspace clustering [5, 11, 1] is a strategy to reduce the complexity of clustering high-dimensional data. Subspace clustering algorithms can divide data space or search space of clusters into subspaces and find clusters in different subspaces within a dataset. All clusters of whole dataset are included in all subspaces. Some subspaces include more interesting clusters but others that contain less interesting clusters or noisy data can be pruned.

Conceptual clustering [6, 9, 3] can seek clusters by concept structures so that the concept can describe the clusters. One approach of conceptual clustering is based on concept lattice [4]. The structure of concept lattice can be used to generate overlapping concepts. Theoretical foundation of concept lattice founds on the mathematical lattice theory [2, 7]. Lattice is a popular mathematical structure for modeling conceptual hierarchies. Concept lattice is a method for deriving conceptual structures out of data. It allows us to analyze and mine the complex data for such as classification, association rules mining, clustering, etc [10]. Furthermore, concept lattice also provides an effective tool of knowledge visualization that plays an important role in data mining. Concept lattice can facilitate pattern understanding, knowledge discovery, and interactive data exploration. 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 .

This paper integrates the techniques of subspace clustering and conceptual clustering and proposes a novel algorithm, called CSC, to extract conceptual clusters in dense subspaces and give the description of clusters by overlapping concepts. The algorithm CSC selects subspaces based

	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 1. Example of formal context

on the density of dataset and exploits hierarchical overlapping clusters that can be described by concepts. The experimental results show the efficiency of this algorithm.

The rest of this paper is organized as follows. The basic concepts of concept lattice are presented in the next section. Section 3 analyzes the search space of conceptual clustering. The algorithm CSC is introduced in section 4. Section 5 presents the experimental results of the algorithm. The paper ends with a short conclusion in section 6.

2 Concept lattice

The core of Formal Concept Analysis (FCA) is concept lattice. FCA provides a natural platform for data analysis and knowledge representation. FCA is different from some of the traditional, statistical means of data analysis and knowledge representation because of its focus on human-centered approaches. Concept lattice facilitates exploring, searching, recognizing, identifying, analyzing, visualizing, restructuring and memorizing conceptual structures [12]. So we can take advantage of the features of concept lattice to extract and interpret the frequent patterns and clusters of data set.

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

Definition 2.1 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 or transactions, while the elements of A are called items or attributes.*

For example, Figure 1 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.

Definition 2.2 *Two* **closure operators** *are defined as $O_1 \rightarrow O_1''$ for set O and $A_1 \rightarrow A_1''$ 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 2.3 *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 2.4 *We say that there is a hierarchical order between two formal concepts $(O_1,\ A_1)$ and $(O_2,\ A_2)$, if $O_1 \subset O_2$ (or $A_2 \subset A_1$). And we say $(O_1,\ A_1)$ is 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 formal 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**.

All concepts and concept lattice of formal context in Figure 1 are illustrated in Figure 2.

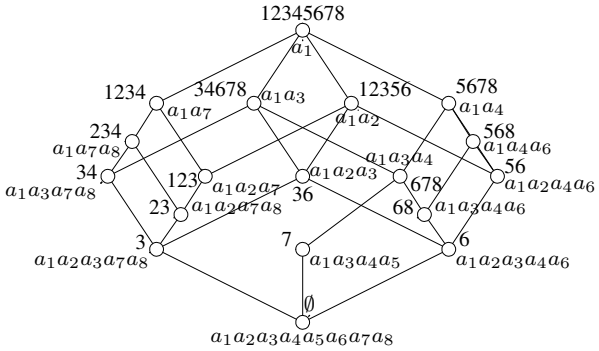


Figure 2. An example of concept lattice

3 Analysis of search space of conceptual clustering

We consider one formal concept as a cluster to analyze the search space of conceptual clustering. For one formal concept, its extent is an overlapping cluster, and the intent is the description of the cluster. Given a concept (O_i, A_i) of (O, A, R), we note sub-concept $(O_{i_{sub}}, A_{i_{sub}})$ of (O_i, A_i). We have $O_{i_{sub}} \subset O_i$ and $A_i \subset A_{i_{sub}}$ by the definition 2.4. (O_i, A_i) is more general than $(O_{i_{sub}}, A_{i_{sub}})$ in conceptual category. O_i is more dense than $O_{i_{sub}}$. We observe that the hierarchical order of cluster based on lattice structure is also hierarchical order of density of clusters.

Lemma 3.1 *$(O_{i_{sub}}, A_{i_{sub}} - A_i)$ is a concept in the sub-context $(O_i, A - A_i, R)$*

Proof : $(O_{i_{sub}}, A_{i_{sub}})$ is a sub-concept of concept (O_i, A_i), then we have $O_{i_{sub}} = (A_{i_{sub}} - A_i)'$ and $O'_{i_{sub}} = (A_{i_{sub}} - A_i)$ in the sub-context $(O_i, A - A_i, R)$. Thus, $(O_{i_{sub}}, A_{i_{sub}} - A_i)$ is a concept in the sub-context $(O_i, A - A_i, R)$.

Lemma 3.2 *All sub-concepts of (O_i, A_i) can be generated from the sub-context $(O_i, A - A_i, R)$.*

Proof : Given a concept (O_l, A_l) of the sub-context $(O_i, A - A_i, R)$, $(O_l, A_l \cup A_i)$ is a sub-concept of (O_i, A_i). Thus all sub-concepts can be generated by the concepts of the sub-context $(O_i, A - A_i, R)$.

In fact, when we analyze the the hierarchical relation between one concept and its sub-concepts, we can simplify the intent of the concept. All the sub-concepts of (O_i, A_i) have common attribute set A_i, so we can consider A_i as one merging attribute, we note $a_i = A_i$ in this case. We give the following definitions to simplify the sub-context and reduce the complexity of searching clusters.

Definition 3.3 *Given an attribute a_i of context (O, A, R), $(\{a_i\}', A - \{a_i\}'', R)$ is called* **projective context** *of a_i.*

Definition 3.4 *Given a concept (O_i, A_i) of context (O, A, R), we can merge all attributes of A_i as a new attribute a_{new}, the projective context of a_{new} is called* **projective context** *of the concept (O_i, A_i).*

Definition 3.5 *Given a context (O, A, R), an attribute a_i is called* **trunk attribute***, if attribute $a_i \in A$, for all $a_j \in A$, $i \neq j$ and $\{a_i\}' \not\subset \{a_j\}'$.*

From lemma 3.1 and the definition 3.4 and 3.5 we can deduce the following lemma.

Lemma 3.6 *Given a concept (O_i, A_i) of context (O, A, R), the projective context of (O_i, A_i) is $(O_i, A - A_i, R)$. If attribute $a_i \in A - A_i$ and a_i is a trunk attribute, then $(\{a_i\}', \{a_i\}'' \cup A_i)$ is a sub-concept of (O_i, A_i).*

Therefore, the projective context of one concept is a subspace for searching cluster. From top concept or cluster, we can search its sub-concepts or sub-clusters in the subspace of clustering. All clusters can be generated in different subspaces of clustering by the same iterative approach.

In many cases, we need not generate all clusters but we are interested in the dense clusters. Thus, we give the following definition to reduce the subspaces of clustering and generate more general concepts or dense clusters.

Definition 3.7 *Given $m =$ the minimum number of objects of cluster, a formal context is called* **ordered dense context** *if we only choose the attributes which number of objects is not less than m, and order these attributes of formal context by number of objects of each attribute from the smallest to the biggest one, and the attribute with the same objects are merged as one attribute. We note ordered dense context $(O, A^{\lhd}, R)_m$ of the formal context (O, A, R).*

Lemma 3.8 *Given ordered dense context $(O, A^{\lhd}, R)_m$, if attribute $a_i \in A^{\lhd}$ and a_i is a trunk attribute, then $(\{a_i\}', \{a_i\})$ is a concept.*

Lemma 3.8 shows us a new strategy to find concpets: we only need know which attribute is trunk attribute in an ordered dense context.

Ordered dense context can reduce the complexity of clustering high-dimensional data. For each subspace of clustering, we can deal with the projective context as a ordered dense context and then search for the clusters. By the hierarchical order, the clusters can be found from more dense subspaces to less dense subspaces of clustering. This is the key idea of the algorithm CSC.

4 Novel algorithm: CSC

In this section, we present the principle of the novel algorithm CSC (Conceptual Subspace Clustering).

Algorithm 1 Algorithm of CSC (Conceptual Subspace Clustering)

```
input: context (O, A, R)
input: m = the minimum density of clustering
output: hierarchical conceptual clusters
generate ODC = ordered dense context (O, A^◁, R)_m
D = ||O|| //D is the density of cluter
if O' ≠ ∅ then
    (O, O') is a conceptual cluster
    ODC = (O, (A − O')^◁, R)_m
end if
while D >= m do
    for all ordered dense context do
        find trunk attributes
        for all trunk attribute a_i do
            ({a_i}', {a_i}) is a conceptual cluster //generate
            sub-concept in ODC
            generate projective context of ({a_i}', {a_i})
            ODC = new ordered dense context
        end for
    end for
    D = ||O|| // in ODC
end while
```

4.1 The principle of algorithm CSC

Given a formal context and minimum density of clustering, we propose the algorithm to generate dense hierarchical conceptual clusters by following steps:

1) First of all, the algorithm needs to generate the ordered dense context $(O, A^{\lhd}, R)_m$.

2) The second step: searching for trunk attributes in the ordered dense context, every trunk attribute a_i forms a concept or conceptual cluster: $(\{a_i\}', \{a_i\})$. $\{a_i\}'$ is cluster, a set of objects that have common attribute $\{a_i\}$ to describe the cluster.

3) For each trunk attribute, we generate the projective context of the trunk attribute, then an ordered dense context. For each ordered dense context, we use the same way as step 2 to find the clusters in the new context. We can continue to generate clusters until the number of objects for all trunk attributes is less than the minimum density of clustering.

4) At the end, we can get all dense hierarchical conceptual clusters. For example, given the minimum density of clustering 2, the algorithm generates all dense hierarchical conceptual clusters(see Figure 3) from a formal context.

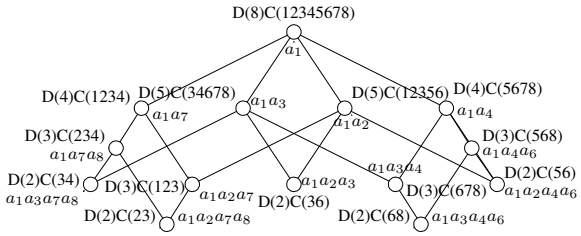


Figure 3. An example: dense hierarchical conceptual clusters mining with algorithm CSC. (minimum density of clustering is 2; for each node, D means density of cluster, C means cluster)

4.2 An example

Given the minimum density of clustering $= 2$, we show how to use the algorithm to generate all conceptual clusters.

For the first step, the algorithm generates the ordered dense context (see Figure 4) based on minimum density of clustering 2. a_5 is not included in the ordered frequent context because the number of objects of a_5 is less than 2.

From the ordered dense context, only a_1 is the trunk attribute. Hence, $\{a_1\}$ forms a concept $(\{1, 2, 3, 4, 5, 6, 7, 8)\}, \{a_1\})$. In figure 3, the first node is represented by $D(8)C(123\ 45678)$ and a_1, D(8) means the density of cluster is 8 and C(12345678) means the cluster is $\{1, 2, 3, 4, 5, 6, 7, 8\}$.

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

Figure 4. Ordered dense context

The density of a_1 is bigger than 2, so we consider the ordered dense context of the projective context of a_1 (see Figure 5). By the same way, we can generate all sub-concepts of $(\{1, 2, 3, 4, 5, 6, 7, 8)\}, \{a_1\})$: $\{D(4)C(1234)$, $a_1a_7\}$, $\{D(5)C(\ 34678)$, $a_1a_3\}$, $\{D(5)C(12356)$, $a_1a_2\}$, $\{D(4)C(\ 5678)$, $a_1a_4\}$, as a_7, a_3, a_2 and a_4 are trunk attributes in the ordered dense context of Figure 5.

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

Figure 5. Ordered dense context of the projective context of a_1

If the density of the trunk attribute is 2, this attribute ends to generate sub-concepts of next level. Otherwise for the next level, we use the same approach to generate all concepts or dense hierarchical conceptual clusters (see Figure 3).

5 Experimental results

We have implemented the algorithm CSC in Java to generate dense hierarchical conceptual clusters. We test the algorithm on some real data and simulation data. In this paper we show the experimental results on the datasets in table 1.

Figure 6 illustrates the comparison of CSC algorithm with the lattice-based algorithm without subspaces. The lattice-based algorithm without subspaces is CSC in the special case: searching hierarchical conceptual clusters in only one searching space. Figure 7 shows the comparison of CSC with the algorithm GALOIS [4]. The run time of

DataSet	ID	Objects	Attributes	Clusters
breast-cancer-wisconsin	d01	699	110	9860
house-votes-84	d02	435	18	10642
SPECT_test	d03	187	23	14532
SPECT_two	d04	267	23	21548
audiology.standardized	d05	26	110	30401
tic-tac-toe	d06	958	29	59503
nursery	d07	12960	31	147577
lung-cancer	d08	32	228	186092
agaricus-lepiota	d09	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 experiments

CSC is faster than the algorithm GALOIS and the lattice-based algorithm without subspaces.

6 Conclusion and further work

This paper integrates the techniques of subspace clustering and conceptual clustering and presents the algorithm CSC for subspace conceptual clustering. The principle of the algorithm is to generate dense ordered context to reduce the complexity of subspace clustering. And then the clusters or concepts are easy to be mined. The results of mining include the overlapping conceptual clusters and the description of clusters. The experimental results illustrate the efficiency of this algorithm.

We will compare the performance of CSC with other clustering algorithms, and develop more efficient techniques for high-dimensional data clustering.

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] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data. *Data Min. Knowl. Discov.*, 11(1):5–33, 2005.

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

[3] G. Biswas, J. B. Weinberg, and D. H. Fisher. Iterate: A conceptual clustering algorithm for data mining. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, (28):100C111, 1998.

[4] C. Carpineto and G. Romano. Galois: An order-theoretic approach to conceptual clustering. In *Proceedings of ICML'93*, pages 33–40, Amherst, Juillet 1993.

[5] C. H. Cheng, A. W.-C. Fu, and Y. Zhang. Entropy-based subspace clustering for mining numerical data. In *Knowledge Discovery and Data Mining*, pages 84–93, 1999.

[6] D. H. Fisher. Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, (2):139–172, 1987.

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

[8] A. Jain and R. Dubes. *Algorithms for Clustering Data*. Prentice Hall College Div, 1988.

[9] M. Lebowitz. Experiments with incremental concept formation: Unimem. *Machine Learning*, (2):103–138, 1987.

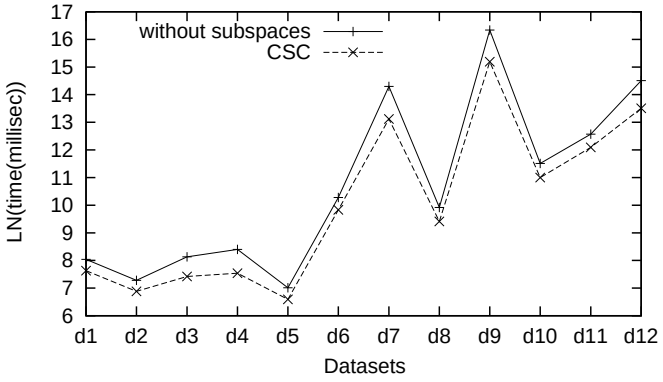
[10] E. Mephu Nguifo, M. Liquiere, and V. Duquenne. *JETAI Special Issue on Concept Lattice for KDD*. Taylor and Francis, 2002.

[11] L. Parsons, E. Haque, and H. Liu. Subspace clustering for high dimensional data: a review. *SIGKDD Explorations*, 6(1):90–105, 2004.

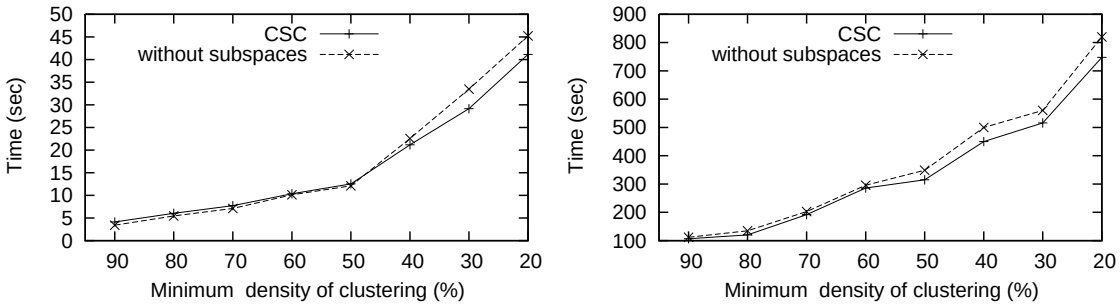
[12] U. Priss. Formal concept analysis in information science. *Cronin, Blaise (ed.),Annual Review of Information Science and Technology*, 40:521–543, 0062.

[13] R. Xu and D. Wunsch II. Survey of clustering algorithms. *Neural Networks, IEEE Transactions on*, 16:645–678, 2005.

[14] O. Zamir and O. Etzioni. Grouper: a dynamic clustering interface to Web search results. *Computer Networks (Amsterdam, Netherlands: 1999)*, 31(11–16):1361–1374, 1999.

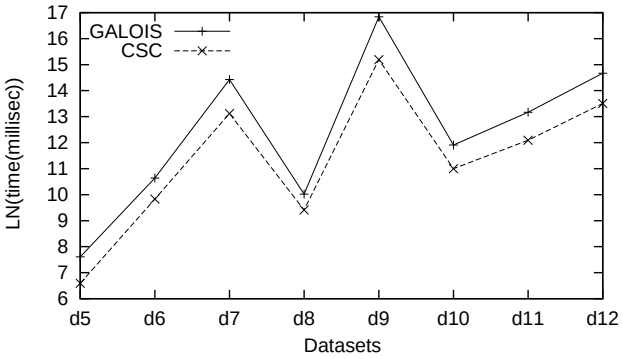


(the minimum density of clustering is 10%)



(dataset: promoters) (dataset: dermatogogy)

Figure 6. Comparison of CSC with the lattice-based algorithm without subspaces



(the minimum density of clustering is 10%)

Figure 7. Comparison of CSC with the algorithm GALOIS