International Journal of Systems Science, vol. 45, no. 5, pp. 1170-1179, May 2014.

An Outlier Mining Algorithm Based on Constrained Concept Lattice

Jifu Zhang [a], Sulan Zhang [a], Kai H. Chang [b], and Xiao Qin[b]

[a] School of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan, P. R. China 030024

[b] Department of Computer Science and Software Engineering, Auburn University, Auburn, AL, USA 36849-5347

jifuzh@sina.com.cn

Abstract: Traditional outlier mining methods identify outliers from a global point of view. These methods are inefficient to find locally-biased data points (outliers) in low dimensional subspaces. Constrained concept lattices can be used as an effective formal tool for data analysis because constrained concept lattices have the characteristics of high constructing efficiency, practicability, and pertinency,. In this paper, we propose an outlier mining algorithm that by treats the intent of any constrained concept lattice node as a subspace. We introduce sparsity and density coefficientsto measure outliers in low dimensional subspaces. The intent of any constrained concept lattice node is regarded as a subspace, and sparsity subspaces are searched by traversing the constrained concept lattice according to a sparsity coefficient threshold. If the intent of any father node of the sparsity subspace is a density subspace according to a density coefficient threshold, then objects contained in the extent of the sparsity subspace node are considered as bias data points or outliers. Our experimental results show that the proposed algorithm performs very well for high red-shift spectral data sets.

Keywords: Constrained Concept Lattice；Outliers；Sparsity Subspace；Density Coefficient

1. Introduction

Concept lattice is also known as Galois lattice and formal concept analysis, which was proposed by Wille in 1982 [1]. Concept lattice is a hierarchical data structure constructed by binary relationships between objects and attributes in a data set, reflecting the relations of generalization and specialization among concepts through Hasse diagrams [1]. Concept lattice - an effective tool for data analysis and knowledge representation - has been widely utilized in the fields of knowledge engineering, data mining, and information retrieval. Currently, research on knowledge discovery using concept lattice includes association rules [3,4,13] and classification rules [5]. However, using concept lattice to discover outliers receives little attention in the literature.

Outliers have the characteristics of grossly deviating from other data points and not satisfying general patterns or behaviors of a data set [6]. Outliers generally come from errors of measurements, inputs, and/or human actions. These data points need to be deleted or revised; otherwise they could adversely affect data analysis results. On the other hand, outliers may possibly reflect true natures of data sets thereby being more valuable than normal data.. Evidence shows that outliers should be safeguarded, because outliers usually help people to discover intriguing and unexpected knowledge. Sample application domains where outliers are widely applied include credit card management, network invasion detection, and medical research. Practically speaking, outliers in high dimensional data sets

exhibit obvious deviations in certain dimensions while no marked departure in other dimensions. Traditional outlier mining methods include distance-based [6], statistical-based [10], local density based [11] and deviation-model-based approaches [12]. These approaches discover outliers from a global point of view. The existing mining methods are inadequate in identifying deviation data points in a low dimensional subspace, especially in the presence of noise. In 2005, Agarwal and Yu put forward a new method of outlier detection for high dimensional data [7]. Their scheme projects high dimensional data into low subspaces, adopts a genetic algorithm to search for sparsity subspaces, and judges whether outliers exist by using sparsity coefficient. The downside of their approach lies in accuracy in outlier discovery. For example, (1) a low dimensional sparsity subspace, decided by the sparsity coefficient, may comprise many normal sparse data points; hence, it is not appropriate to use the sparsity coefficient to determine the sparsity subspace; (2) there is no guarantee that a subspace of the smallest sparsity coefficient can be discovered (see, for example, [7]); (3) there is no guarantee that all data points satisfying outlier conditions can be discovered, meaning that the completeness of results cannot be ensured. Because outliers are often apt to occur in some dimensions, a key problem of discovering outliers in high dimensional data sets is how to efficiently and accurately identify sparsity subspaces in which the outliers are contained.

Each node in a concept lattice is made up of two parts: intent (attribute set) and extent (object set possessing its attribute set). Thanks to the completeness and accuracy of concept lattice, concept lattice provides a complete and accurate data structure to express and describe subspaces containing outliers. Constrained concept lattice [9] adopts the predicate logic to describe background knowledge that users are interested in. Constrained concept lattice has a few salient features. First, it introduces background knowledge into the process of constructing concept lattice.Second, it improves the constructing efficiency of concept lattice by reducing space-time complexity. Third, it enhances practicability and pertinency of extracting knowledge from the concept lattice.

In low dimensional subspaces, a constrained-concept-lattice-based outlier mining algorithm treats the intent of any constrained concept lattice node as a subspace. We define and discuss sparsity and density coefficients, which measure outliers in low dimensional subspaces. Sparsity subspaces are searched according to a sparsity coefficient threshold. If the intent of any father node of the sparsity subspace is a density subspace according to a density coefficient threshold, then objects contained in the extent of the sparsity subspace node are considered as bias data points or outliers in low dimensional subspace. We conduct experiment to show that our algorithm is effective in mining outliers in the high red-shift spectral data sets.

2. Concept lattice and sparsity subspace

2.1. Concept lattice

Concept lattice is a formal context containing a set of objects, a set of attributes and a relation between objects and attributes [1]. The relationship in concept lattice describes attributes possessed in each objects.. Now we formally present the formal context as a triple $K = (G, M, I)$, where G is an object set and M is an attribute set and I is a binary relation between G and M, i.e., $I \subseteq G \times M$. Given $g \in G$, $m \in M$, and $(g, m) \in I$, we denote this relation as "g I m", meaning that object *g* has attribute m. Table 1

shows an example formal context, where objects G set is O = {1, 2, 3, 4, 5, 6}, attributes M set is A ={H, C, N, I}, and incidence relation I is marked as Y in the table.

Let K = (G, M, I) be a formal context. A formal concept J is a pair of elements (A, B), where $A \subseteq G$ and $B \subseteq M$. Elements A and B are the extent and intent of formal concept J. And the two elements A and B satisfy the following conditions.

(1) $A=B' = \{a \in G | \forall b \in B, a\ I\ b\}$

(2) $B=A' = \{b \in M | \forall a \in A, a\ I\ b\}$

In general, concept lattice provides a way to discover sensible groups of objects that have common attributes in a certain context [1]. A concept is a collection of all objects sharing a set of attributes in a given context. The partial-order relation $\leq$ among all formal concepts is expressed as $(A_1, B_1) \leq (A_2, B_2) \Leftrightarrow A_1 \subseteq A_2 \Leftrightarrow B_2 \subseteq B_1$. All formal concepts and a partial-order relation in the formal context K form a complete lattice called concept lattice or general concept lattice. Let us , denote a complete lattice as L(G, M, I).

Table 1 Formal context

patient ID	H (headache)	C (chest distress)	N (nausea)	I (insolation)
1	Y	N	N	Y
2	Y	Y	N	Y
3	Y	Y	Y	N
4	N	N	Y	N
5	N	N	Y	N
6	N	Y	N	Y

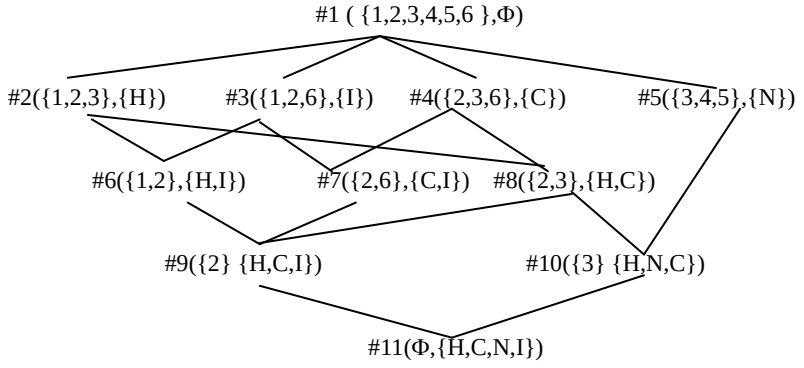

Figure 1 A example of a concept lattice constructed from the formal context presented in Table 1.

There are general approaches to constructing concept lattices. The first one is the batch algorithm [14] and another one is the incremental algorithm [2]. The Godin algorithm is a representative incremental construction algorithm creating concept lattices [2]. Figure 1 is a concept lattice constructed from the formal context in Table1. In this figure, H denotes headache, C denotes chest distress, N denotes nausea, I denotes insolation, Y denotes “YES”, and N denotes “NO”.

2.2. Constrained concept lattice

During the process of constructing concept lattice, users may not be interested in certain combinations of attributes contained in concept lattice intents. In addition, the combinations may be insignificant from applications' perspective. Thus, we can construct a concept lattice according to users' interest and understanding of a data set. This approach allows us to create the structure of a concept lattice in a pertinent and practical way**.** Before adopting the predicate logic to express background knowledge representing users' interest and understanding, we first define predicates, which are used to form a predicate formula P to express background knowledge [9].

Definition 1 Let us considera formal concept h(A, B). If h(A, B) is a user's concerned node andP(h(A, B))=.T. indicates the attributes in intent B satisfy the constrained condition, P. , then we define Care(B)= .T.. If h (A, B) is not a concerned node for user (i.e., P (h(A, B)) =.F.), then we define Care (B) = .F..

Definition 2 A formal concept is described as h = ((A, B), P), where P is a constrained condition, $A \subseteq G$ is the extent of h, $B \subseteq M$ is the intent of h. A and B in formal concept h satisfy the following three conditions.

(1) $f(B)= A=B'=\{A \in G| \forall B \in M, A\ I\ B\}$

(2) $f'(A)=B=A'=\{B \in M| \forall A \in G, A\ I\ B\}$

(3) care(B)=.T.

The concept lattice with the above structure is called constrained concept lattice, denoted as < L(G, M, I, P), ≤ >, where L(G, M, I, P) is a concept (node) set in which any intent satisfies constrained condition P and ≤ is a partial-order relation. The ordered pair h=(A, B) satisfies upper three conditions and $h \in L$ (G, M, I, P).

Definition 3 Let $h_1 = ((A_1, B_1), P)$ and $h_2 = ((A_2, B_2), P)$ be two nodes in a constrained concept lattice, where h_1 is a sub-concept of h_2 (or h_2 is a super-concept of h_1). We define $h_1 \leq h_2 \Leftrightarrow B_2 \subseteq B_1 \Leftrightarrow A_1 \subseteq A_2$. If there is not $h_3 = ((A_3, B_3), P)$, satisfying $h_1 \leq h_3 \leq h_2$, then h_2 is called a father concept of h_1, h_1 is a son concept of h_2. Otherwise, h_2 is called a forefather node of h_1.

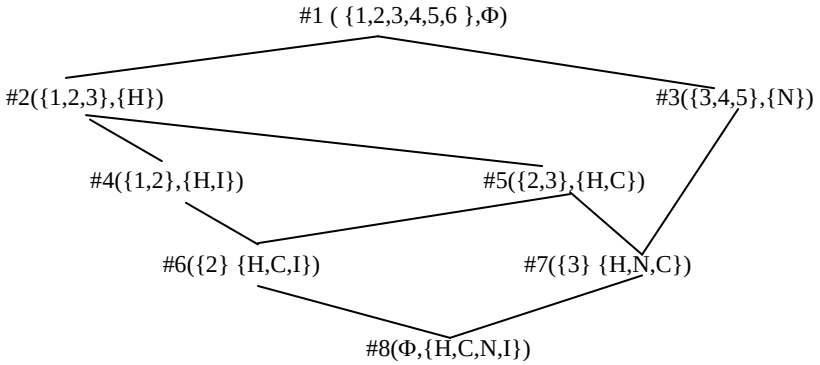

Figure 2 Constrained concept lattice with background knowledge H∨N

An incremental constructing algorithm CCLA was proposed to create constrained concept lattice [9]. Any constrained concept lattice constructed by CCLA equals to a general concept lattice when there is no background knowledge (P=Φ). Figure 2 is a constrained concept lattice constructed from the formal context in Table1 with background knowledge P=H∨N. From Figures 1 and 2, it can be seen that

constrained concept lattice is a sub-structure of concept lattice in the same formal context, and all combinations of the attributes contained in concept lattice intent satisfy constrained condition defined by background knowledge. Therefore, a constrained concept lattice has more practicability and more pertinent than concept lattice. Constructing a constrained concept lattice is more efficient than building the corresponding concept lattice.

2.3. Sparsity coefficient and sparsity subspace

Suppose there are N records in a high dimensional data set DB. Each dimension is divided into θ discrete intervals; every record is independent of each other. Constructing a k-dimensional cube by selecting k attributes, one can randomly distribute N records in the cube with the probability $(1/\theta)^k$ according to the Bernoulli probability. The record number of every interval is its mathematic expectation $N*(1/\theta)^k$. Agarwal and Yu introduce sparsity coefficient S(D) (see the following formula)to measure the deviation degree of data points in a subspace [7], where f = 1/θ, n (D) is the number of records in the k-dimensional cube D.

$$S(D) = (n(D) - N*f^k)/\sqrt{N*f^k(1-f^k)}$$

S(D) < 0 means the number of data in D is lower than expectation. A small value of S(D) indicates that data in k-dimensional cube D are sparse.

Definition 4 Let us consider a data set DB, its attribute set M, its object set G and background knowledge P. D_1 is a subspace composed of attribute subset B_1, ($B_1 \subseteq M$) with object set A_1, ($A_1 \subseteq G$). For a subspace, D_2, composed of attribute set B_2, ($B_2 \subset B_1$) with object set A_2. If $A_2 \neq A_1$, then D_1 is called a reduction subspace, B_1 is called a reduction attribute set. If Care(B_1)=.T., then D_1 is a constrained reduction subspace, B_1 is a constrained reduction attribute set.

Definition 5 Suppose we have data set DB, attribute set M, object set G, and a sparsity threshold TS. For any reduction subspace, D, composed of reduction attribute set B, ($B \subseteq M$), if its sparsity coefficient S(D) ≤ TS, D is called a sparsity subspace.

For a k-dimensional sparsity subspace D and the object number n(D) ≥0, from $(n(D) - |G|*f^k)/\sqrt{|G|*f^k(1-f^k)} \leq$TS, the following expression holds: $0<k\leq \log_{\phi}(|G|/TS^2+1)$ [7]. This expression shows that maximal dimension of bias data points or outliers measured by S(D) formula is $\lfloor \log_{\theta}(|G|/TS^2+1) \rfloor$.

3. A constrained-concept-lattice-based outlier mining

Adopting constrained concept lattices as a tool to express subspaces, we improve the constructing efficiency of concept lattice and reduce the space-time complexity of searching subspaces. Our approach makes outlier mining results practicable and pertinent. All sparsity subspaces can be searched by traversing a contained concept lattice. To improve accuracy of the discovered outliers, we focus on an effective sparsity metric to measure outliers in low dimensional subspaces.

In a sparsity subspace, sparsity coefficient merely indicates that the number of objects contained in

the subspace is far less than the expected value. Sparsity coefficient is unable to effectively represent the deviation degree of a data point in the sparsity subspace. Hence, the sparsity subspace determined only by S(D) is not necessarily a true sparsity subspace.

Now let us take a look at an example. where there are many young patients in a hospital and the patients suffer from hypertension. If very few young people suffer from hypertension, then a valuable outlier emerges: “young people suffer from hypertension”. On the other hand, if young patients in the hospital are few, a discovered outlier by sparsity coefficient S(D) shows that young people suffering from hypertension is obviously not reasonable. This is because the small number of young people suffering from this disease is chiefly due to the small number of young patients. To avoid such irrational results caused by the sparseness of objects in the lower dimensional subspaces, we have to examine the true subspaces of a sparsity subspace. Only when the number of data objects contained in the true subspaces of a sparsity subspace reaches a certain level, objects included in the sparsity subspace can be considered as outliers. This allows us to identify valuable deviation data or true outliers. For example, only if the number of young patients and the number of people suffering hypertension are both sufficiently large, the conclusion "young people suffering from hypertension" is a reasonable conclusion.

As mathematical expectation can be considered as the average number of objects in a subspace, we enable a user to provide a new coefficient(density coefficient). The pair of the expectation and the user supplied coefficiency are then used evaluate the dense degree of subspaces.

Definition 6 Suppose we have a data set DB, attribute set M, and object set G. Each attribute is divided into θ ranges, DENSE is a density coefficient given by users, D is a subspace composed of attribute set B, ($B \subseteq M$) with object set A, ($A \subseteq G$). D is a density subspace if $|A| \geq DENSE * |G| * (1/\theta)^{|B|}$.

Because $|G| * (1/\theta)^{|B|}$ is the average number (expected value) of objects in a subspace, a large value of DENSE leads to a large number of objects contained in density subspace D. A large DENSE value also results in high dense degrees of density subspace D. The minimal number of objects contained in the density subspace equals to the expected value when DENSE=1. Such a minimal number is smaller than the expected value when 0<DENSE<1; the minimal number is bigger than the expected value when DENSE>1.

Definition 7 Suppose we have a data set DB, attribute set M, and object set G. Let D be a reduction subspace constructed by any reduction attribute set B, ($B \subseteq M$) with object set A ($A \subseteq G$) and D be a sparsity subspace. If reduction subspace D_1 constructed by attribute set B_1($B_1 \subset B$ and $|B_1| = |B| - 1$) and D_1 is a density subspace, then D is an outlier subspace and objects in A are outliers.

Definition 8 Let K = (G, M, I) be any formal context and P be background knowledge. For $\forall h = ((A, B), P) \in L(G, M, I, P)$, if $\exists B_1 \subseteq B$ and following conditions are satisfied, then B_1 is a constrained intent reduction of h.

(1) $B_1' = B' = A$

(2) $B_2' \supset B_1' = A$ (for any $B_2 \subset B_1$)

(3) $Care(B_1) = .T.$

Definition 8 shows that for constrained intent h, there are not redundant attributes in its every constrained intent reduction in which the attributes satisfy P. The details on the intent reduction

algorithm can be found in [8].

Definition 9 Let K = (G, M, I) be any formal context, P be background knowledge. For $\forall h=((A, B), P)\in L(G, M, I, P)$, the constrained intent reduction set RED of h is $\{B_i| B_i$ is a constrained intent reduction of h$\}$. If $\exists B_i\in$ RED , the subspace D constructed by attribute set B_i is a sparsity subspace and a subspace D_1 constructed by any attribute set S ($S\subset B_i$ and $|S|=|B_i|-1$) is a density subspace, then concept h is a constrained outlier concept and objects in A are outliers.

Theorem 1 Let K = (G, M, I) be a formal context and P be its background knowledge. For $\forall h=((A, B), P)\in L(G, M, I, P)$, any father concept of h is $h_i=((A_i, B_i),P)\in L(G, M, I, P)$. If a constrained reduction attribute set B_1 ($B_1\subset B$) satisfies $B_1\cap (B-B_i)\neq \Phi$, then B_1 is a constrained intent reduction of h.

Proof: $B\supset B_1\Rightarrow B_1'\supseteq A$. ($B_1'$, B_1'') satisfies the completeness of concept lattice [1]; therefore, we have $(B_1', B_1'')\in L(G, M, I)$. According to Definitions 1, 2 and 4, $Care(B_1)\Rightarrow((B_1', B_1''),P)\in L(G,M,I,P)$. From $B_1'\supset A$ and $B_1''\subset B$, we show that B_1 isn't a constrained intent reduction of h. According to Definition 3, ((B_1', B_1''), P) is the father or the forefather concept of h $\Rightarrow \exists$ father concept $h_j = ((A_j, B_j), P)\in L(G, M, I, P)$ of h, $A_j\subseteq B_1'$ and $B_j\supseteq B_1''$. $\forall$ father concept $h_i =((A_i, B_i),P)\in L(G, M, I, P)$ of h satisfy $B_1\cap (B-B_i)\neq \Phi\Rightarrow B_1\cap (B-B_j)\neq \Phi \Rightarrow B\cap B_1\cap \overline{B_j}\neq\varnothing$. Hence, $B_j\supseteq B_1\Rightarrow B_1\cap \overline{B_j}=\varnothing\Rightarrow B\cap B_1\cap \overline{B_j}=\varnothing$. It is inconsistent with $B\cap B_1\cap \overline{B_j}\neq\varnothing$, so the hypothesis is wrong. □

Theorem 2 Let K = (G, M, I) be a formal context and P be its background knowledge. For $\forall h=((A, B), P)\in L(G, M, I, P)$, if B_1 is a constrained intent reduction of h and a father concept of h is $h_i=((A_i, B_i),P)\in L(G, M, I, P)$, then $B_1\cap(B-B_i)\neq\Phi$.

Proof: Suppose a father concept $h_j=((A_j,B_j),P)\in L(G,M,I,P)$ of h satisfies $B_1\cap(B-B_j)=\Phi$. Because $B_1\cap (B-B_j) = \Phi\Rightarrow B_1\cap B\cap \overline{B_j}=\varnothing$, according to Definition 8, $B_1\subseteq B\Rightarrow B_1\cap B=B_1$, $B_1\cap B\cap \overline{B_j}=\varnothing\Rightarrow B_1\cap \overline{B_j}=\varnothing\Rightarrow B_1-B_j=\Phi\Rightarrow B_1\subseteq B_j\Rightarrow B_1'\supseteq B_j'$. Since h_j is a father concept of h, hence $B_j'\supset A$. $B_1'=A$ is inconsistent with $B_1'\supseteq B_j'$; therefore, the hypothesis is wrong. □

Theorems 1 and 2 suggest that the constrained intent reduction set of any concept lattice node can be derived by computing a difference set between the node's intent and father concept intents.

Theorem 3 Let K = (G, M, I) be any formal context and P be the background knowledge. For outlier subspace S constructed by constrained reduction attribute set D($D\subseteq M$) and object set $O\neq\Phi$ ($O= D'$) contained in S, then $\exists h=((A, B), P)\in L(G, M, I, P)$ and the constrained intent reduction set $RED=\{B_i| B_i$ is the constrained intent reduction of h $\}$ satisfy $D\in RED$ and $O= A$.

Proof: According to [1], (D′, D″) satisfies the completeness of concept lattice $\Rightarrow$(D′, D″)$\in$ L(G, M, I)$\Rightarrow D\subseteq D''$. According to Definitions 3 and 9, Care(D) $\Rightarrow$ ((D′, D″), P)$\in$L(G, M, I, P), O=D′$\Rightarrow\exists$h= ((O, D″), P)$\in$ L(G, M, I, P). O=D′ $\Rightarrow$ constrained reduction attribute set D. According to Definitions 4 and 8, we have D$\in$RED. □

Theorem 4 Let K = (G, M, I) be a formal context and P be K's background knowledge. If $\exists$ outlier subspace S constructed by constrained reduction attribute set D($D\subseteq M$) and object set $O\neq\Phi$ ($O=D'$) contained in S, then $\exists h=((A,B),P)\in L(G,M,I,P)$ satisfies A=O and h is an outlier constrained concept.

Proof：According to Theorem 3, ∃h=((A, D′), P)∈ L(G, M, I, P) and D∈RED={B_i| B_i is the constrained intent reduction of h}. Definition 7 shows that subspace S_1 constructed by attribute set D_1, ($D_1 \subset D$ and $|D_1|=|D|-1$), is a density subspace if S is a sparsity subspace. According to Definition 9, h=((A, D′), P) is an outlier constrained concept. □

Theorem 3 and 4 suggest that the outlier mining method using constrained concept lattices has the characteristics of completeness, i.e., for an outlier subspace, we can identify its corresponding outlier concept in the constrained concept lattice. This approach guarantees that objects contained in the outlier subspace are the same as in the extent of the constrained outlier concept node.

4. **A constrained-concept-lattice-based outlier mining algorithm**

The above analysis allows us to compute a constrained intent reduction set by (1) generating a difference set between a concept and its father concept intents and (2) identify all the outliers according to Definition 9. Our constrained-concept-lattice-based outlier mining algorithm is described as follows：

Algorithm OMACCL (outlier mining algorithm based on constrained concept lattices)

input： constrained concept lattice, sparsity coefficient threshold TS, density coefficient DENSE, background knowledge $P=P_1 \vee P_2 \vee ... \vee P_n$

output：Outlier

1 Outlier=$\varnothing$；

2 sort(RCL)； /* arrange the concept in ascending order according to the number of attributes in the concept intent */

3 k = $\left\lfloor \log_{\theta}(|G|/TS^2+1) \right\rfloor$； /* k is maximal dimensional number of the outlier , according to [7]

4 N= $|G|*f^k + TS*\sqrt{|G|*f^k(1-f^k)}$； /*maximum objects contained in outlier concept extent

5 FOR each concept C=((A, B), P) in RCL and |B|≥K and |A|≤N DO

6 IF C= (A, B) is not NC THEN

7 Red = B∩{{P_1},{P_2}...{P_n}}, and delete {$\varnothing$} element in Red；

8 compute the constrained intent reduction set Red of C, according to [8]；

9 FOR each R in Red DO

10 IF |R| = K THEN

11 R_1={R'_i| $R'_i \subset$R 且| R'_i |=|R|-1}

12 IF all constructing subspaces for $\forall R'_i \in R_1$ satisfy the condition of density subspace THEN

13 Outlier=Outlier ∪ A；

14 Mark every child of C as NC；

15 BREAK； /*exit loop*/

16 END IF

17 END IF

18 END FOR

16 ELSE Mark every child of C as NC；

17 END IF
18 END FOR
19 END.

During the process of computing the intent reduction (see line 8), there is no need to consider constrained intent reductions whose attribute number is larger than K (see line 10). The OMACCL algorithmfirst computes a difference set between the intent of a concept and the intent of its father concept. In the worst case, a concept has |C|-3 father concepts (|C| is the number of concepts in a concept lattice). Thus, the time complexity of computing the difference set is O(|C|). Then, each difference set needs to perform an intersection operation with the intent reduction set, and the maximal number of constrained intent reductions whose attribute number is not bigger than K, is $\theta * C_m^{\lceil k/2 \rceil}$ for any concept (m is the attribute number of concept intent and θ is the number of discrete intervals.) Therefore, the time complexity of computing the constrained intent reduction is O($\theta * C_m^{\lceil k/2 \rceil} *$|C|). When the OMACCL algorithm processes density subspaces, OMACCL enumerates the |R|-1-dimension subset of R and decides whether it is a density subspace using Definition 6. Therefore, the time complexity of processing the density subspace is O(|R|). The total time complexity of the OMACCL algorithm is O($\theta * C_m^{\lceil k/2 \rceil} *$|C|^2). Since the father concept number of a concept is far smaller than |C|-3, the number of intent reductions is much smaller than $\theta * C_m^{\lceil k/2 \rceil}$, and the number of concept nodes whose attribute number is equal to or smaller than K is much smaller than |C|. Therefore, the time complexity of the algorithm is below O($\theta * C_m^{\lceil k/2 \rceil} *$|C|^2).

5. **An example.**

We show how the OMACCL algorithm works using an example of contained concept lattice plotted in Figure 2. Let sparsity coefficient threshold TS be –0.1 and density coefficient DENSE be 1. In this example, the maximal dimension of all the outlier subspaces is k=2, meaning that the maximal attribute number contained in an intent is not more than 2. The maximal object number contained in an extent is N =1. The constrained intent reduction set of concept #6 is Red={{H, C, I}} (see [8]). Because we have |{H,C,I}|>2, concept #6 is not an outlier concept. The constrained intent reduction attribute set of concept #7 is Red={{H, N},{N, C}}. Now we consider constrained intent reduction {H, N}. Subspaces constructed by subsets {H} and {N} of {H, N} are density subspaces and; therefore, concept #7 is an outlier concept. Object id3 in concept #7 is an outlier, which suggests that patient id3 has the symptom of headache, chest distress, nausea, but s/he does not have the symptom of isolation. Figure 2 shows the structure of constrained concept lattice. Our OMACCL algorithm reduces the space-time complexity and the searching range of concepts. Because the attributes of the constrained intent reduction is of users' interests and outlier information extracted is of the users' concern, OMACCL improves the practicability and pertinency of outlier-mining results.

6. **Experiments and analysis**

We implement the OMACCL algorithm and conduct experiments using a server that contains a Pentium III-1.0G CPU, 256M memory, Windows XP operating system and ORACLE 9i DBMS. For comparison purpose, we also implement two existing solutions:the CCLA [9] and Godin [2] algorithms Visual C++6.0. Input data sets for all the tested algorithms are derived from a high red-shifted spectrum database provided by the National Observatory in Beijing, China. The high red-shifted spectra preprocessed in the following two steps are used as formal context of our experiments. 1) For each high red-shifted spectrum, we choose 44 characteristic lines as attributes. 2) For each characteristic line, flux, peak-width and shape information of the absorption line are divided into 7 discrete values. Tables 2, 3 and 4 show the efficiency and correctness of the OMACCL algorithm. In the three tables, NII-1, OI-2 and H\d\ga denote the characteristic lines of the high red-shifted spectra.

The experiment results summarized in Tables 2，3 and 4 validate that all outliers mined from the constrained concept lattice are included in the outliers derived from the concept lattice. The mining results also satisfy the constrained conditions specified as the background knowledge. This result proves that the OMACCL algorithm is correct. Because the time and space complexity of constructing a constrained concept lattice is less than building a general concept lattice, the outlier-mining efficiency of OMACCL is higher than those of the two existing schemes. Using the SQL statements, we validate that all the outliers mined from the constrained concept lattice satisfy the sparsity coefficient threshold TS, the density coefficiency DENSE, and the background knowledge P. The results show that OMACCL is feasible and highly effective.

In Table 2 (see also lines 3 and 4 of OMACCL), the outliers are mined in a three dimensional subspace when $-4.8 \le TS \le -4.3$ and in a four dimensional subspace when $-1.4 \le TS \le -0.48$. Therefore, the outlier number is 10 when TS=-4.3 and is 3 when TS=-1.4. The number of records, the density coefficient, and the background knowledge remain unchanged. When the TS value is decreased, some of the original sparsity subspaces no longer satisfy the condition of sparsity subspace; some sparsity concepts no longer satisfy the condition of sparsity concept. Thus, the number of the extracted outliers drops. When the number of sparsity concepts decreases, only the intent reduction densities of the forefather concept nodes of those sparsity concepts need to be checked; thus, the time spent in outlier mining is reduced. Sparsity coefficient reflects the sparse degree of a subspace. The density excludes the possibility of data deviation caused by the small number of samples in a subspace. Hence, with the unchanged density degree, decreasing TS and increasing the deviation degree of data objects in outlier subspace can improve the accuracy of discovered results.

Table2 Outlier mining time and outlier numbers of different TS values

(DENSE=1，|DB|=5412，background knowledge：NII-1)

TS	general concept lattice		constrained concept lattice	
	Mining Time (sec)	Outlier Number	Mining Time (sec)	Outlier Number
–4.8	543	3	117	3
–4.5	856	6	134	4
–4.3	1158	16	242	10
–1.4	1154	3	561	3
–0.9	1581	11	816	11
–0.48	2022	19	983	14

In Table 3, we fix the number of records, TS, and the background knowledge. With the increasing value of the density coefficient, the number of objects included in a density subspace increases, and some of the original density subspaces would no longer satisfy the condition of density subspace. As a result, the number of outliers decreases. If a sparsity concept is not an outlier concept with respect to its intent reductions, we need to determine real subset in its forefather concept node to figure out whether it is a density subspace. When it comes to outlier concepts, we simply need to find out some intent reductions that satisfy formula S(D) without checking the rest of the intent reductions. With the decreasing number of outliers, the time spent in mining outliers is substantially reduced. In summary, the number of records or the TS value affect the number of sparsity concept nodes, which in turn make significant impacts on on the efficiency of outlier mining. In contrast, the density coefficient only decides whether an entire or part of the intent of a sparsity concept should be reduced; hence, the density coefficient has less impact on the efficiency of outlier mining. The density coefficient along with TS affects the performance of the outlier-mining algorithm.

Table3 Outlier mining time and outlier numbers of different DENSE values

(TS= –1.4，|DB|=5412，background knowledge：OI-2)

DENSE	general concept lattice		constrained concept lattice	
	Mining Time (sec)	Outlier Number	Mining Time (sec)	Outlier Number
1	1154	3	654	2
0.9	1152	4	653	3
0.8	1145	8	649	5
0.7	1139	12	647	6
0.6	1124	19	646	7

In Table 4, the number of records, TS, and DENSE remain unchanged. When the constrained condition is increased, node number of the constrained concept lattice is decreased accordingly. Consequently, time spent in building the lattice is also decreased. Therefore, the number of outliers and mining time are both decreased due to the decrease in the number of subspaces satisfying TS and DS. Since the outliers mined from a constrained concept lattice satisfy the constraining condition given by users, the outliers are more pertinent and practical. Our OMACCL algorithm avoids inefficient outlier

minings.

Table 4 Outlier mining time and outlier numbers of different background knowledge

(TS= –1.4，DENSE =0.8，|DB|=5412)

background knowledge	Time of building Constrained concept Lattice (sec)	Node number of Constrained concept lattice	Mining Time (sec)	Outlier Number
OI-2∨H\d\ga	1103	88242	798	7
H\d\ga	424	48962	568	5
OI-2	853	69409	649	5
(OI-2∧H\d\ga)∨ (SII-3∧CaII-2)	340	38340	350	4
OI-2∧H\d\ga	252	30129	170	3
SII-3∧CaII-2	46	11801	137	2

Using the SQL statements in the above experiments, all outliers mined in Tables 2, 3, and 4 meet the requirements of sparsity coefficient, density coefficient thresholds and constrained condition. Furthermore, the astronomers from the National Observatory in Beijing, China manually validated the correctness of these outliers. In the case where the number of data points is 5412, constrained condition P=(OI-2 ∨H\ga) ∨(SII -3∧CaII-2), DEN=0.8 and TS= –1.4, our OMACCL discovered four outliers (see Figure 3). The evidence shows that OMACCL can identify the smallest sparsity coefficient subspace, thereby guaranteeing the completeness of the outlier mining results.

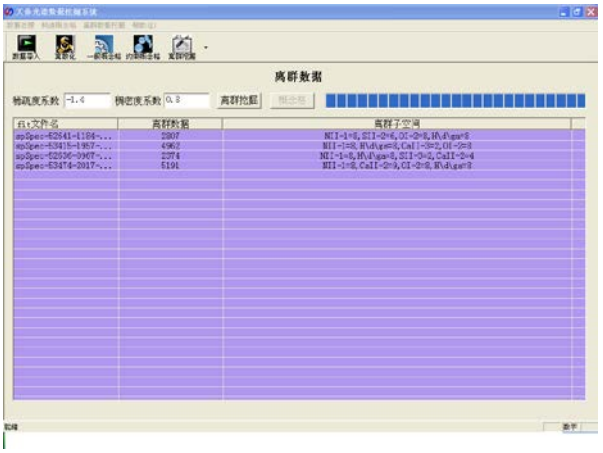

Figure 3 Outliers

7. Related work and comparisons

Concept lattice, being a useful tool for data analysis and knowledge processing, can concisely express the relations of generalization and specialization among concepts through the Hasse diagram [1]. With its straight-forwardness, conciseness and completeness of knowledge expression, concept lattice has been successfully applied to data mining and knowledge discovery, digital library, literature retrieval,

software engineering, medical data analysis, and CBR（case-based reasoning). Research on knowledge discovery using concept lattice includes association rules and classification rules [3, 4, 5, 13]. However, little attention has been paid to outlier discovery suing concept lattice.

Nowadays, there are mainly four types of outlier-mining approaches. (1) Statistical methods [10,19] learn a distributive or probable model for a given data set and adopt a discordance test to decide outliers according to a model. These methods demand parameters of the data set, the distribution parameters and the expected outliers, which are in most cases unknown. The common problem is that most tests aim at a single attribute. However, data mining normally requires discovering outliers in a high dimensional space. (2) Distance-based methods [6,18] discover outliers by computing the distances between data points; their computational complexity is usually high; this is especially true for high dimensional data sets. It is difficult to find data deviations appearing in a low dimensional space. Typical algorithms of these methods are either index-based or unit-based. (3) Deviation-based methods [12] imitate reasoning patterns of human thoughts. In these approaches, an apparent difference of one data point to others is rapidly discovered after observing a consecutive sequence. The main downside of such methods is that dissimilarity function is difficult to define if there is no priori knowledge about the data. The commonly used techniques include the sequence abnormal technique and the OLAP data cube technique [15]. (4) In the density methods [11], object distances for all dimensions of data points are first computed. Then, the reachable density of each point is computed. Finally, outliers are mined by local deviations. Such methods are very time consuming, because the LOF (Local Outlier Factor) value of each data point must be calculated. Since most outlier mining methods treat outliers from a global point of view, these methods have a stumbling block in discovering deviated data points in a low-dimensional subspace. Thus, the existing solutions are inadequate for high-dimensional data.

Recently, a novel robust outlier detection method was proposed for high dimensional datasets [16]. This scheme detects high dimensional outliers based on user examples and tolerates incorrect inputs. Another strategy was designed to detect outlying subspaces in high-dimensional databases using the genetic algorithm paradigm for searching outlying subspaces [17]. The subspace-based high-dimensional outlier mining method presented in [7] utilizes the genetic algorithm to identifyoutliers and improves mining efficiency and result rationality. Since genetic algorithms cannot ensure the completeness of search results, the above schemes may miss or fail to discover data points with the smallest sparsity coefficient. If sparsity coefficient is only used to decide data behaviors in a subspace, the algorithms cannot effectively deal with subspace sparsity from a normal data set's sparsity; consequently, mining results are inaccurate and incomplete.

OMACCL treats each concept node of a constrained concept lattice as a subspace, the sparsity coefficient of which can be efficiently calculated. All sparsity subspaces are searched by traversing constrained concept lattice nodes according to a sparsity coefficient threshold. Objects contained in the extent of the sparsity subspace node are considered as outliers if the intent of any father node of the sparsity subspace is a density subspace. Making use of the density coefficient, OMACCL effectively overcomes the problem of normal data deviations in the low-dimensional subspace due to data sparsity. OMACCL also improves the accuracy of mining results. Because outliers mined from a constrained concept lattice satisfy constraining conditions specified by users, the outliers are more pertinent and

practical. Thus, mining results yielded by OMACCL can be easily understood and further processed by users.

8.Conclusions

In this paper, we propose a constrained concept lattice based outlier mining algorithm OMACCL that deals the intent of a constrained concept lattice node as a subspace. Using background knowledge offered by users to construct a constrained concept lattice, our OMACCL algorithm reduces the time and space complexity of constructing lattices. Using the real-world spectral data sets, we demonstrate that OMACCL can extract highly pertinent and accurate outlier knowledge. As a future research direction, we plan to integrate the implemented OMACCL algorithm into an outlier mining system for astronomical spectrum data. The integrated system will be operated in the National Observatory in Beijing, China.

Acknowledgment. This work is partially supported by the National Natural Science Foundation of P. R. China (61073145). Xiao Qin's work was made possible thanks to NSF awards CCF-0845257 (CAREER), CNS-0757778 (CSR), CCF-0742187 (CPA), CNS-0831502 (CyberTrust), OCI-0753305 (CI-TEAM), DUE-0837341 (CCLI), and DUE-0830831 (SFS).

REFERENCES

[1] Wille R. Restructuring Lattice Theory: an Approach based on Hierarchies of Concepts. In : Rival I ed. Ordered Sets. Dordrecht: reidel,1982, 415-470

[2] Robert Godin Etc. Incremental Concept Formation Algorithms based on Galois (concept) lattices, Computational Intelligence,1995, 11(2), 246- 267

[3] Ohbyung Kwon, ihoon Kim. Conceptlattices for visualizing and generating user profiles for context-aware service recommendations. Expert Systems with Applications,2009,36(2): 1893–1902

[4] Sadok Ben Yahia, Ghada Gasmi, Engelbert Mephu Nguifo. A new generic basis of "factual" and "implicative" association rules. Intelligent Data Analysis,2009, 13(4): 633-656

[5] Jonas Poelmans, Paul Elzinga, Stijn Viaene and Guido Dedene. Formal Concept Analysis in Knowledge Discovery: A Survey. Lecture Notes in Computer Science, 2010, Volume 6208/2010, PP:139-153

[6] Knorr E, Ng R. Algorithms For mining Distance-based Outliers in Large Datasets, (C). In: Proc. of the 24th VLDB Conference. New York USA ,1998, PP：392-403

[7] C C.Agarwal,P S.Yu. An Effective and Efficient Algorithm for High-dimensional Outlier Detection, The International Journal on Very Large Data Bases,2005, 14 (2), 211 – 221

[8] Xie Zhipeng, Liu Zongtian. Intent Reduct of Concept Lattice Node and Its Computing. Computer Engineering,2001, 27(3), 9-10,39

[9] Zhang jifu Etc. Constrained Concept Lattice and its Construction Method, CAAI Transactions on Intelligent Systems, 2006, l(2), 31-38

[10] Barnett V and Lew T. Outliers in Statistical Data. New York: JohnWiley &Sons, 1994

[11] Hui Cao. Enhancing effectiveness of density-based outlier mining scheme with density similarity

neighbor based outlier factor. Expert Systems with Applications,2010,37(12)：8090-8101

[12] Arning A, Agrawal R, Raghavan P. A linear method for deviation in large database . In : Proc. of Int. Conf. Data Mining and Knowledge Discovery, 1996,164-169

[13] Jhieh-Yu Shyng, How-Ming Shieh,Gwo-Hshiung Tzeng. An integration method combining Rough Set Theory with formal concept analysis for personal investment portfolios. Knowledge-Based Systems,2010,23(6): 586–597

[14] Nourine L, Raynaud O. A fast algorithm for building lattices. Information processing letters, 1999,71(5), 199-204.

[15] Han J, Kambr M. Data mining concepts and techniques. Morgan Kaufmann Publishers, San Franciso, 2001

[16] Cui Zhu , Hiroyuki Kitagawa , Christos Faloutsos. Example-Based Robust Outlier Detection in High Dimensional Datasets. Proceedings of the Fifth IEEE International Conference on Data Mining, 2005 , p.829-832

[17] Ji Zhang, Qigang Gao, Hai Wang. A Novel Method for Detecting Outlying Subspaces in High-dim ensional Databases Using Genetic Algorithm, Proceedings of the Sixth IEEE International Conference on Data Mining, 2006, p.731-740

[18] Chenglong Tang,Shigang Wang ,Wei Xu . New fuzzy c-means clustering model based on the data weighted approach. Data & Knowledge Engineering, 2010, 69(9):881–900

[19] Hido Shohei , Tsuboi Yuta , Kashima Hisashi,ect. Statistical outlier detection using direct density ratio estimation. Knowledge and information systems, 2011, 26(2): 309-336