Local Pattern Mining from Sequences using Rough Set Theory

Ken Kaneiwa* and Yasuo Kudo†
*National Institute of Information and Communications Technology
3-5 Hikaridai, Seika, Soraku, Kyoto 619-0289, Japan
Email: kaneiwa@nict.go.jp
†College of Information and Systems, Muroran Institute of Technology
27-1 Mizumoto, Muroran 050-8585, Japan
Email: kudo@csse.muroran-it.ac.jp

***Abstract*—Sequential pattern mining is a crucial but challenging task in many applications, e.g., analyzing the behaviors of data in transactions and discovering frequent patterns in time series data. This task becomes difficult when valuable patterns are locally or implicitly involved in noisy data. In this paper, we propose a method for mining such local patterns from sequences. Using rough set theory, we describe an algorithm for generating decision rules that take into account local patterns for arriving at a particular decision. To apply sequential data to rough set theory, the size of local patterns is specified, allowing a set of sequences to be transformed into a sequential information system. We use the discernibility of decision classes to establish evaluation criteria for the decision rules in the sequential information system.**

I. Introduction

Data mining algorithms have been developed as tools to discover valuable patterns and rules from large amounts of data. Sequential pattern mining algorithms [1], [2], [3] enable us to find frequent patterns in sequential datasets. Sequential pattern mining requires the analysis of an ordered list of itemsets (e.g., a list of actions or orders) that can be modeled by a sequence. In order to effectively carry out the task, we have to extract only valuable patterns included in sequences by skipping noisy and meaningless patterns. However, frequent data mining algorithms are not feasible when it comes to extracting local (or implicit) patterns from noisy data. This is because the algorithms may not work when valuable patterns do not appear frequently or when waste patterns appear frequently. In fact, the frequencies of such valuable patterns may be less than a user-specified threshold, but setting a lower threshold leads to the recovery of a number of meaningless patterns.

In order to solve the problem, we have to logically and combinationally analyze patterns in sequences by checking the occurrences of local patterns that consistently result in a decision. For such an analysis, rule generation in rough set theory [4], [6], [7] provides a data mining algorithm based on the notions of attribute reduction and reduced decision rules. One of the advantages of rough set data mining is that it can generate reduced and consistent decision rules by logically checking all combinations of condition and decision attributes in an information system. Thus, rough set theory can be used to generate essential attributes through attribute reduction of logical combinations. However, sequential pattern mining algorithms have not been well studied in the context of rough set theory. Extending this approach to sequential pattern mining entails a logical analysis of local patterns in granular computing, which differs from the frequency analysis of sequential patterns.

In this paper, we propose a sequential pattern mining algorithm using the rule generation from discernibility in rough set theory. This algorithm computes subsequences of a fixed size that are regarded as local patterns hidden inside sequences. A sequential information system consists of the subsequences obtained from a set of sequences so that we can apply sequential data to the rough set data mining. The decision rules generated from a sequential information system are said to be *sequential* decision rules. In each of the rules, the condition attributes represent the occurrences of local patterns in a sequence. In order to estimate the local patterns in the rules, we establish the evaluation of occurrence-based accuracy and coverage for sequential decision rules.

Our algorithm has the following interesting features.

- **Occurrences of Local Patterns:** Given a set of sequences, a sequential information system is constructed from the attributes that denote the subsequences of a fixed size, where each attribute value represents the number of occurrences of a local pattern in a sequence.
- **Granularities of Sequences:** The different sizes of local sequence patterns determine the diversity of granularities in a sequential information system. In other words, longer subsequences correspond to smaller granularities because they contain more information.
- **Reduced and Consistent Decision Rules:** In rough set theory, attribute reduction generates *reduced* decision rules. In addition, the decision rules are *consistent*, and hence, they are significantly different from the frequent association rules in traditional data mining, because logically inconsistent rules are excluded due to the discernibility of decision classes.

II. Rough Sets

An attribute a is a mapping $a\colon U \to V_a$ where U is a non-empty finite set of objects (called the universe) and V_a is the value set of a. An information system is a pair $T = (U, A)$ of the universe U and a non-empty finite set A of attributes. Let B be a subset of A. The B-indiscernibility relation is defined by an equivalence relation I_B on U such that $I_B = \{(x, y) \in U^2 \mid \forall a \in B.a(x) = a(y)\}$. The equivalence class of I_B for each object x ($\in U$) is denoted by $[x]_B$. Let X be a subset of U. We define the lower and upper approximations of X by $\underline{B}(X) = \{x \in U \mid [x]_B \subseteq X\}$ and $\overline{B}(X) = \{x \in U \mid [x]_B \cap X \neq \emptyset\}$. A subset B of A is a reduct of T if $I_B = I_A$ and there is no subset B' of B with $I_{B'} = I_A$ (i.e., B is a minimal subset of the condition attributes without losing discernibility).

A decision table is an information system $T = (U, A \cup \{d\})$ such that each $a \in A$ is a condition attribute and $d \notin A$ is a decision attribute. Let V_d be the value set $\{d_1, \ldots, d_u\}$ of the decision attribute d. For each value $d_i \in V_d$, we obtain a decision class $U_i = \{x \in U \mid d(x) = d_i\}$ where $U = U_1 \cup \cdots \cup U_{|V_d|}$ and for every $x, y \in U_i$, $d(x) = d(y)$. The B-positive region of d is defined by $P_B(d) = \underline{B}(U_1) \cup \cdots \cup \underline{B}(U_{|V_d|})$. A subset B of A is a relative reduct of T if $P_B(d) = P_A(d)$ and there is no subset B' of B with $P_{B'}(d) = P_A(d)$.

We define a formula $(a_1 = v_1) \wedge \cdots \wedge (a_n = v_n)$ in T (denoting the condition of a rule) where $a_j \in A$ and $v_j \in V_{a_j}$ ($1 \leq j \leq n$). The semantics of the formula in T is defined by $[\![(a_1 = v_1) \wedge \cdots \wedge (a_n = v_n)]\!]_T = \{x \in U \mid a_1(x) = v_1, \ldots, a_n(x) = v_n\}$. Let φ be a formula $(a_1 = v_1) \wedge \cdots \wedge (a_n = v_n)$ in T. A decision rule for T is of the form $\varphi \to (d = d_i)$, and it is true if $[\![\varphi]\!]_T \subseteq [\![(d = d_i)]\!]_T (= U_i)$. The accuracy and coverage of a decision rule r of the form $\varphi \to (d = d_i)$ are respectively defined by:

$$accuracy(T, r, U_i) = \frac{|U_i \cap [\![\varphi]\!]_T|}{|[\![\varphi]\!]_T|}$$

$$coverage(T, r, U_i) = \frac{|U_i \cap [\![\varphi]\!]_T|}{|U_i|}$$

In the evaluations, $|U_i|$ is the number of objects in a decision class U_i and $|[\![\varphi]\!]_T|$ is the number of objects in the universe $U = U_1 \cup \cdots \cup U_{|V_d|}$ that satisfy condition φ of rule r. Therefore, $|U_i \cap [\![\varphi]\!]_T|$ is the number of objects satisfying the condition φ restricted to a decision class U_i.

III. Sequential Data in Rough Sets

A. Sequential Information Systems

An itemset a_i is a non-empty set of items, and the size of a_i is the cardinality of a_i, i.e., $|a_i|$. A sequence s is an ordered list of itemsets $\langle a_1, a_2, \ldots, a_n\rangle$, simply denoted by $a_1a_2\cdots a_n$. The size of s (denoted $||s||$) is the number of elements of the list $a_1a_2\cdots a_n$, and the length of s is the total number of the sizes $|a_1|$, $|a_2|$, …, $|a_n|$. A sequence $s_1 = a_1a_2\cdots a_n$ is a subsequence of another sequence $s_2 = b_1b_2\cdots b_m$ (denoted $s_1 \sqsubseteq s_2$), if there are integers $i_1 < i_2 < \cdots < i_n$ such that $a_1 \subseteq b_{i_1}$, $a_2 \subseteq b_{i_2}$, …, $a_n \subseteq b_{i_n}$. The empty sequence ϵ is a subsequence of any sequence. A sequence s_1 is a strict subsequence of another sequence s_2 (denoted $s_1 \sqsubseteq^{st} s_2$) if there exists an integer i such that $a_1 \subseteq b_i$, $a_2 \subseteq b_{i+1}$, …, $a_n \subseteq b_{i+n-1}$.

As a practical example, an ordered list of itemsets can be used to represent a list of sequential actions of an agent where each itemset corresponds to an action, which consists of a set of operations corresponding to items. Let us consider the following four sequences:

$$\begin{aligned} s_1 &= aabcac \\ s_2 &= bcca \\ s_3 &= cba \\ s_4 &= aabca \end{aligned}$$

where $a = \{i_1, i_2\}$, $b = \{i_2, i_3, i_4\}$, and $c = \{i_2, i_3\}$ are itemsets and i_1, i_2, i_3, and i_4 are items. The sequence s_1 is the series $aabcac$ of actions of an agent and the sequence s_2 is the series $bcca$ of actions of another agent. In addition, the sequences s_3 and s_4 are the series cba and $aabca$ of actions, respectively, of two other agents.

In order to apply this sequential data to rough set theory, we characterize the local patterns of sequences in an information system that can be used to discern the sequences. In our new method, the occurrences of subsequences in each sequence are calculated to express the local features of a set of sequences by using an information system.

Definition 1 (Sequential Information System): Let $U_{sq} = \{s_1, \ldots, s_n\}$ be a set of sequences and A_{sq} be a set of subsequences of sequences $s_1, \ldots, s_n$ in U_{sq}. A *sequential* information system is an information system $T = (U_{sq}, A_{sq})$ where for each attribute $a \in A$ (named by a subsequence), $a(x)$ maps the number of occurrences of the subsequence a in each sequence $x \in U_{sq}$.

We denote the concatenation $\overbrace{s\cdots s}^{n}$ of sequence s by s^n (in particular, s^0 denotes the empty sequence ϵ). We can precisely define the number n of occurrences of subsequence s_1 in sequence s_2 as follows:

$$\Omega_{s_1}(s_2) = n$$

if the concatenation s_1^n is a subsequence of s_2 but the concatenation s_1^{n+1} is not a subsequence of s_2. For example, $\Omega_{ac}(caac) = 1$ and $\Omega_{ac}(abcbacc) = 2$, i.e., ac appears once in the sequence $caac$ and twice in the sequence $abcbacc$.

Definition 2 (Sequential Decision Table): A sequential decision table is a decision table $T' = (U_{sq},\ A_{sq} \cup \{d\})$ such that $T = (U_{sq}, A_{sq})$ is a sequential information system and $d \notin A_{sq}$ is a decision attribute.

B. Granularities of Sequences

The local size of valuable patterns varies depending on the property of sequential data in many application domains. To deal with the diversity of sequential data, we consider the different sizes of subsequences in a sequential information system that set granularities for the features of sequences in rough set theory. As a result of this method, the size k subsequences of a sequence have a smaller granularity than the size $k-1$ subsequences of that.

In order to capture local patterns from a sequence s, we define the set of subsequences of a size occurring in the sequence s as follows.

Definition 3 (Size k Subsequences): The set of size k subsequences of s is defined by

$$Sub_k(s) = \{s' \mid s' \sqsubseteq s \ \& \ ||s'|| = k\}$$

For sequences s_1, s_2, s_3, and s_4 shown in Section III-A, we obtain the following sets of size 2 subsequences:

$$\begin{aligned} Sub_2(s_1) &= \{aa, ab, ac, ba, bc, ca, cc\} \\ Sub_2(s_2) &= \{ba, bc, ca, cc\} \\ Sub_2(s_3) &= \{ba, ca, cb, cc\} \\ Sub_2(s_4) &= \{aa, ab, ac, ba, bc, ca, cc\} \end{aligned}$$

In $Sub_2(s_1)$ with $s_1 = aabcac$, subsequence aa consists of the first and second itemsets in s_1; subsequence ab consists of the second and third itemsets in s_1. In this example, we can intuitively interpret the size 2 subsequences as changes from one action to another when the sequences describe agents' actions. Therefore, the size 2 subsequence sets $Sub_2(s_1), \ldots, Sub_2(s_4)$ indicate the local changes in actions of the four agents.

Furthermore, local patterns in a sequential information system are analyzed more strictly as follows. By limiting the definition of subsequences, we obtain the set of strict subsequences occurring in the sequence s.

Definition 4 (Size k Strict Subsequences): The set of strict size k subsequences of s is defined by

$$Sub_k^{st}(s) = \{s' \mid s' \sqsubseteq^{st} s \ \& \ ||s'|| = k\}.$$

For example, we have the following sets of strict size 2 subsequences in the sequences s_1, s_2, s_3, and s_4:

$$\begin{aligned} Sub_2^{st}(s_1) &= \{aa, ab, ac, bc, ca, cc\} \\ Sub_2^{st}(s_2) &= \{bc, ca, cc\} \\ Sub_2^{st}(s_3) &= \{ba, cb, ca, cc\} \\ Sub_2^{st}(s_4) &= \{aa, ab, ac, bc, ca, cc\} \end{aligned}$$

In $Sub_2^{st}(s_1)$ with $s_1 = aabcac$, the local pattern ba in $Sub_2(s_1)$ is not a strict subsequence of s_1, but it is nevertheless a subsequence of s_1. This is because there is an itemset c between b and a (i.e., bca) in the sequence s_1. That is, we can use Sub_k to generate lazy local patterns by skipping itemset c in sequence bca.

	aa	ab	ac	ba	bc	ca	cb	cc	d
s_1	1	1	2	1	1	1	0	1	1
s_2	0	0	0	1	1	1	0	1	0
s_3	0	0	0	1	0	1	1	1	0
s_4	1	1	1	1	1	1	0	1	1

Table I
SIZE 2 SEQUENTIAL INFORMATION SYSTEM T_1

Another granularity can be analyzed by extracting size 3 subsequences from the sequences s_1, s_2, s_3, and s_4. Intuitively, in the analysis of actions, the size 3 subsequences imply more complex combinations of action changes than the size 2 subsequences. Similar to the above example, the sets of size 3 subsequences are captured from the sequences s_1, s_2, s_3, and s_4 as follows.

$$\begin{aligned} Sub_3(s_1) &= \{aaa, aab, aac, aba, abc, aca, acc, bac, bca, \\ &\qquad bcc, cac, cca, ccc\} \\ Sub_3(s_2) &= \{bca, bcc, cca, ccc\} \\ Sub_3(s_3) &= \{cba, cca\} \\ Sub_3(s_4) &= \{aaa, aab, aac, aba, abc, aca, acc, bca, cca\} \end{aligned}$$

The combinations of itemsets occurring in the size 3 subsequences are more complex (e.g., $Sub_3(s_1)$ contains 12 local patterns) but those in the strict size 3 subsequences are not very complex, as can be seen in the following:

$$\begin{aligned} Sub_3^{st}(s_1) &= \{aab, aac, abc, acc, bca, cac\} \\ Sub_3^{st}(s_2) &= \{bcc, cca, ccc\} \\ Sub_3^{st}(s_3) &= \{cba, cca\} \\ Sub_3^{st}(s_4) &= \{aab, aac, abc, acc, bca, cca\} \end{aligned}$$

Let S be a set of sequences. We denote $Sub_k(S) = \bigcup_{s \in S} Sub_k(s)$ (resp. $Sub_k^{st}(S) = \bigcup_{s \in S} Sub_k^{st}(s)$).

C. Transformation of Sequences

We define a transformation from a finite set of sequences into a sequential information system with respect to size k.

Definition 5 (Transformation): Let $k > 0$ be a non-negative integer, and let $S = \{s_1, \ldots, s_j\}$ be a finite set of sequences. The size k sequential information system is defined as a sequential information system $T = (U_{sq}, A_{sq})$:

$$U_{sq} = S \text{ and } A_{sq} = Sub_k(S)$$

In addition, if A_{sq} is defined by $Sub_k^{st}(S)$, then T is the strict size k sequential information system.

After a finite set of sequences is transformed into a sequential information system $T = (U_{sq}, A_{sq})$, the information system is extended to a decision table $T' = (U_{sq}, A_{sq} \cup \{d\})$ by adding decision attribute d. For example, we can set a decision attribute such that the sequences s_1 and s_4 result in a success (denoted value 1), but the sequences s_2 and s_3 cause a failure (denoted value 0). This setting is modeled by supplementing the decision attribute d to the

	aa	ab	ac	ba	bc	ca	cb	cc	d
s_1	1	1	1	0	1	1	0	1	1
s_2	0	0	0	0	1	1	0	1	0
s_3	0	0	0	1	0	1	1	1	0
s_4	1	1	1	0	1	1	0	1	1

Table II
STRICT SIZE 2 SEQUENTIAL INFORMATION SYSTEM T_2

information system $T = (U_{sq}, A_{sq})$ with $d(s_1) = d(s_4) = 1$ and $d(s_2) = d(s_3) = 0$. In Table I, we show a sequential decision table that is obtained from the transformation from the sequences s_1, s_2, s_3, and s_4 into the size 2 sequential information system T_1, and the decision attribute d. In the table, the attributes are labeled by the size 2 subsequences

$$aa,\ ab,\ ac,\ ba,\ bc,\ ca,\ cb, \text{ and } cc$$

in $Sub_2(s_1) \cup Sub_2(s_2) \cup Sub_2(s_3) \cup Sub_2(s_4)$. For example, $aa(s_1) = 1$ and $ac(s_1) = 2$ indicate that the local patterns aa and ac occur in s_1 once and twice, respectively, and $cb(s_1) = 0$ indicates that cb does not occur in s_1.

Table II shows a sequential decision table of a strict size 2 sequential information system transformed from the sequences s_1, s_2, s_3, and s_4 along with the decision attribute d. The size 2 subsequences in Table I contain some discontinuous ordered patterns but the strict size 2 subsequences in Table II do not include them. For example, $ba(s_1) = 0$ means that the strict pattern ba does not occur in sequence s_1, but the lazy pattern ba does occur in the sequence.

From the sets of subsequences in $Sub_3(s_1)$, $Sub_3(s_2)$, $Sub_3(s_3)$, and $Sub_3(s_4)$, the size 3 sequential and strict size 3 sequential information systems T_1 and T_2 in Tables III and IV, respectively, are transformed from the sequences s_1, s_2, s_3, and s_4. Consequently, the number of subsequences increases in comparison with the size 2 sequential information systems.

D. Accuracy and Coverage

Using the transformation discussed in Section III-C, we can obtain a size k sequential information system $T_k = (U_{sq}, A_{sq})$ from a set of sequences. The sequential decision table $T'_k = (U_{sq}, A_{sq} \cup \{d\})$ is constructed by adding a decision attribute d for the sequences in U_{sq} to the information system T_k. This decision table is used to generate decision rules for T_k of the form:

$$(a_1 = n_1) \wedge \cdots \wedge (a_n = n_n) \Rightarrow (d = v)$$

where each a_i denotes a subsequence and each n_i expresses the number of occurrences of subsequence a_i by a non-negative integer. Let T be a sequential decision table. A decision rule for T can be called a sequential decision rule if there is an attribute condition $a_i = n_i$ in the rule such that $n_i \neq 0$.

Here, we discuss the interpretation of such a sequential decision rule. From the sequential decision table $T = (U_{sq}, A_{sq} \cup \{d\})$, we can generate sequential decision rules as follows:

$$(cca = 1) \wedge (acc = 1) \Rightarrow (d = 1)$$

This rule implies that if a sequence contains the local patterns cca and acc, then it results in $d = 1$. However, the following decision rule is not valuable for our purpose.

$$(cba = 0) \wedge (bcc = 0) \Rightarrow (d = 1)$$

This is because the condition attributes indicate that no occurrence of the local patterns cba and bcc in a sequence results in the derivation of the decision attribute $d = 1$. In order to analyze agents' behaviors, some patterns that actually occur have to be mined from the sequential data. However, we do not exclude decision rules if they indicate the occurrence and non-occurrence of local patterns:

$$(cac = 1) \wedge (acc = 0) \Rightarrow (d = 1)$$

This rule means that the occurrence of local pattern cac results in the decision attribute $d = 1$ as long as the local pattern acc does not appear.

We define an evaluation function for sequential decision rules that determines whether each size k sequential information system is well represented when it comes to classifying the decision class. To measure varieties of local sequence patterns for each sequence, we calculate the sum of numbers of the occurring patterns as follows.

Definition 6 (Sum of Occurring Local Patterns): Let S be a set of sequences and let $A' \subseteq A_{sq}$. The sum of numbers of occurring local patterns $o(s, A')$ in each sequence $s \in S$ is defined by

$$o(s, A') = \sum_{a \in A'} sign(a(s))$$

where the sign function $sign(n)$ is defined by $sign(n) = 1$ if $n > 0$ and $sign(n) = 0$ if $n = 0$.

We extend the function $o(s, A')$ to a set of sequences S by defining $o(S, A') = \sum_{s \in S} o(s, A')$.

Definition 7 (Occurrence-Based Accuracy and Coverage): Let S_i be a decision class in S and let $s \in S_i$. The occurrence-based accuracy $o\_accuracy$ and occurrence-based coverage $o\_coverage$ of a sequential decision rule r of the form $\varphi \rightarrow (d = d(s))$ with $d(s) \in V_d$ are defined as follows.

$$o\_accuracy(T, r, S_i) = \frac{o(S_i \cap [\![\varphi]\!]_T, A_\varphi)}{o([\![\varphi]\!]_T, A_\varphi)}$$

$$o\_coverage(T, r, S_i) = \frac{o(S_i \cap [\![\varphi]\!]_T, A_\varphi)}{o(S_i, A_\varphi)}$$

where $A_\varphi = \{a \in A \mid a = v \text{ occurs in } \varphi\}$.

We can define another measurement of the occurrence-based coverage by replacing $o(S_i, A_\varphi)$ with $|S_i| \cdot |A_\varphi|$.

Definition 8 (Variant of Occurrence-Based Coverage): Let S_i be a decision class and let $s \in S_i$. A variant

	aaa	aab	aac	aba	abc	aca	acc	bac	bca	bcc	cac	cba	cca	ccc	d
s_1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
s_2	0	0	0	0	0	0	0	0	1	1	0	0	1	1	0
s_3	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
s_4	1	1	1	1	1	1	1	0	1	0	0	0	1	0	1

Table III
SIZE 3 SEQUENTIAL INFORMATION SYSTEM T_3

	aab	aac	abc	acc	bca	bcc	cac	cba	cca	ccc	d
s_1	1	1	1	1	1	0	1	0	1	0	1
s_2	0	0	0	0	0	1	0	0	1	1	0
s_3	0	0	0	0	0	0	0	1	1	0	0
s_4	1	1	1	1	1	0	0	0	1	0	1

Table IV
STRICT SIZE 3 SEQUENTIAL INFORMATION SYSTEM T_4

Algorithm *sq_mining*

input: set of sequences $S = \{s_1, \ldots, s_n\}$,
maximum subsequence size m,
decision attribute d, bool b
output: list of sets of decision rules $(R_2, \ldots, R_m)$
1: **begin**
2: **for** $k = 2$ **to** m **do**
3: $R_k = \emptyset$;
4: $A_k = subsq(s_1, k, b) \cup \cdots \cup subsq(s_n, k, b)$;
5: **for** $s \in S$ and $a \in A_k$ **do**
6: $a(s) = subsq\_count(s, a, b)$
7: **rof**
8: $T_k = (S, A_k \cup \{d\})$;
9: $\mathcal{R} = reducts(T_k)$;
10: **for** $B \in \mathcal{R}$ **do**
11: **for** $i = 1$ **to** $|V_d|$ **do**
12: **for** $s \in S_i$ **do**
13: $R_k = R_k \cup \{rule(s, B, T_k)\}$;
14: **rof**
15: **rof**
16: **rof**
17: **rof**
18: **return** $(R_2, \ldots, R_m)$;
19: **end**;

Figure 1. Sequential pattern mining algorithm.

$vo\_coverage$ of the occurrence-based coverage of a sequential decision rule r of the form $\varphi \to (d = d(s))$ with $d(s) \in V_d$ is defined as follows.

$$vo\_coverage(T, r, S_i) = \frac{o(S_i \cap [\![\varphi]\!]_T, A_\varphi)}{|S_i| \cdot |A_\varphi|}$$

where $A_\varphi = \{a \in A \mid a = v \text{ occurs in } \varphi\}$.

IV. SEQUENTIAL PATTERN MINING ALGORITHM

This section describes a sequential pattern mining algorithm $sq\_mining(S, m, d)$ for a set of sequences S, a maximum subsequence size m, and a decision attribute d. In Fig.1, we show a sequential data mining algorithm that returns a list of sets of (sequential) decision rules $R_2, \ldots, R_m$ (from size 2 to m), such that the condition attributes in each rule indicate the occurrences of subsequences. This algorithm is outlined as follows.

1) **Transformation:** For each size k from 2 to m, a set of sequences is transformed into size k (resp. strict size k) sequential information systems if $b = 0$ (resp. $b = 1$) by calling the following subroutines.
   a) **Subsequence generation:** The set of size k subsequences $Sub(S)$ or strict subsequences $Sub^{st}(S)$ is generated by checking all the partial patterns of given sequences. These subsequences are used to represent attribute names in the sequential information system.
   b) **Subsequence counting:** The occurrences of subsequences are counted to set the values of attributes in the sequential information system.
2) **Rule generation:** By a rough set rule generation method, reduced decision rules are generated from the sequential decision table where condition attributes are represented by the occurrences of size k subsequences.

A. Transformation

In lines 2 - 17 of the mining algorithm sq_mining, for each size k from 2 to m, the set of size k subsequences $Sub(S)$ or strict size k subsequences $Sub^{st}(S)$ is extracted from sequences in order to construct the size k or the strict size k sequential information system. In line 4, all the subsequences of size k in S are generated as attribute names, which are in $A_k = subsq(s_1, k, b) \cup \cdots \cup subsq(s_n, k, b)$.

As shown in Fig.2, the subsequence generation algorithm $subsq(s, k, b)$ for sequence s, subsequence size k, and bool value b. This algorithm computes $Sub_k(s)$ if $b = 0$ and $Sub^{st}_k(s)$ if $b = 1$. In $subsq(s, k, b)$, we use some operations for sequences. Let $s = a_1 a_2 \cdots a_n$ be a sequence. Then, $start(s)$ and $other(s)$ return the first itemset a_1 and the sequence of the other itemsets $a_2 \cdots a_n$. Let s_1 and s_2 be two sequences. Then, $concat(s_1, s_2)$ is the concatenation of

Algorithm *subsq*

input: sequence s, subsequence size k, bool b
output: a set of sequences S

```
1:  begin
2:      Δ = ∅;
3:      if size(s) < k then return ∅
4:      else if k = 0 then return {ϵ}
5:      else if b = 0 then
6:          Δ = {concat(start(s), s′) |
7:                  s′ ∈ subsq(other(s), k − 1, 0)}
8:              ∪subsq(other(s), k, 0);
9:      else if b = 1 then
10:         Δ = {concat(start(s), s′) |
11:                 s′ ∈ subsq(other(s) ↑ k − 1, k − 1, 1)}
12:             ∪subsq(other(s), k, 1)
13:     for x ⊆ start(s) do
14:         Δ = Δ ∪ subsq(concat(x, other(s)), k, b);
15:     rof
16:     return Δ;
17: end;
```

Figure 2. Subsequence generation algorithm.

s_1 and s_2, i.e., $concat(s_1, s_2) = s_1 s_2$. In lines 13 - 15 of algorithm $subsq(s, k, b)$, for every subset x of the first itemset $start(s)$, this algorithm is recursively called in order to generate the set of subsequences $subsq(concat(x, other(s))$. This is because the subsequences of s contain subsets x of the itemsets of s, i.e., the sequence ab is a subsequence of the sequence ac if $b \subseteq c$ where a, b, and c are itemsets.

After generating the subsequences, in lines 5 - 7 of the mining algorithm, it calculates the numbers of occurrences $a(s)$ of local patterns denoted by the attributes a in A_k and the sequences s in S, which become their attribute values in a sequential information system $T_k = (S, A_k \cup \{d\})$ (in line 8). As a subroutine, the subsequence counting algorithm $subs\_count(s_1, s_2, b)$ shown in Fig.3 counts the number of occurrences of subsequence pattern s_2 in sequence s_1.

B. Rule Generation

In line 9, the set $\mathcal{R} = reducts(T_k)$ [6] of all the relative reducts of size k sequential information system $T_k = (U_{sq}, A_{sq})$ is computed by the standard reduct set computation in [7]. Each $B \in \mathcal{R}$ is a minimal subset of the condition attributes that are the attributes $a_1, \ldots, a_l$ expressed by subsequences. This means that the subsequences denoted by $a_1, \ldots, a_l$ are essential to discern the decision classes $S_1, \ldots, S_{|V_d|}$. In lines 10 - 16, the reduced decision rules generated by $rule(s, B, T_k)$ are added to the set R_k of decision rules for size k for each relative reduct $B \in \mathcal{R}$ where i is a natural number from 1 to $|V_d|$ and S_i is a decision class of S.

Algorithm *subsq_count*

input: sequence s_1, sequence s_2, bool b
output: number of subsequences ct

```
1:  begin
2:      π = s2; ct = 0;
3:      while π ∈ subsq(s1, |π|, b) do
4:          π = concat(π, s2);
5:          ct = ct + 1;
6:      elihw
7:      return ct;
8:  end;
```

Figure 3. Subsequence counting algorithm.

V. Conclusion

We have proposed an alternative method for sequential pattern mining using rough set theory. In our method, we represent the local features of sequences by using a sequential information system where attributes correspond to the occurrence of size k subsequences as local patterns. The proposed mining algorithm computes sequential decision rules according to the size of subsequences by changing the size from 2 to a maximal number in order to check different granulates for sequential data.

Acknowledgment

This research has been partially supported by the Japanese Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research (20700147).

References

[1] R. Agrawal and R. Srikant, "Mining sequential patterns," in *Proceedings of the 11th international conference on data engineering (ICDE'95)*, 1995, pp. 3–14.

[2] Q. Zhao and S. S. Bhowmick, "Sequential pattern mining: A survey," Nanyang Technological University, Singapore, Tech. Rep. 2003118, 2003.

[3] J. Ayres, J. Gehrke, T. Yiu, and J. Flannick, "Sequential pattern mining using a bitmap representation," in *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002, pp. 429–435.

[4] Z. Pawlak, *Rough Sets: Theoretical Aspects of Reasoning about Data*. Kluwer Academic Publishers, 1992.

[5] K. Kaneiwa, "A rough set approach to mining connections from information systems," in *In Proceedings of the 25th ACM Symposium on Applied Computing*, 2010, pp. 990–996.

[6] S. K. Pal and P. Mitra, *Pattern Recognition Algorithms for Data Mining: Scalability, Knowledge Discovery, and Soft Granular Computing*. Chapman & Hall, Ltd., 2004.

[7] J. G. Bazan, H. S. Nguyen, S. H. Nguyen, P. Synak, and J. Wróblewski, "Rough set algorithms in classification problem," in *Rough set methods and applications*, L. Polkowski, S. Tsumoto, and T. Y. Lin. Physica-Verlag, 2000, pp. 49–88.