Expert Systems with Applications 37 (2010) 7419–7426

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Two novel feature selection methods based on decomposition and composition

Na Jiao [a,b,*], Duoqian Miao [b], Jie Zhou [b]

[a] Department of Information Science and Technology, East China University of Political Science and Law, Shanghai 201620, PR China
[b] Department of Computer Science and Technology, Tongji University, Shanghai 201804, PR China

ARTICLE INFO

Keywords:
Feature selection
Decomposition
Composition
Master-table
Sub-table

ABSTRACT

Feature selection is a key issue in the research on rough set theory. However, when handling large-scale data, many current feature selection methods based on rough set theory are incapable. In this paper, two novel feature selection methods are put forward based on decomposition and composition principles. The idea of decomposition and composition is to break a complex table down into a master-table and several sub-tables that are simpler, more manageable and more solvable by using existing induction methods, then joining them together in order to solve the original table. Compared with some traditional methods, the efficiency of the proposed algorithms can be illustrated by experiments with standard datasets from UCI database.

1. Introduction

As the capability of acquiring and storing information increases, more and more features (attributes) and objects (instances) are involved in pattern recognition, machine learning and data mining. Thus, there are a lot of irrelevant or redundant features for a target task. It is known that superfluous features will confuse the learning algorithms and deteriorate mining performance (Guyon, 2003; Yu & Liu, 2004). Hence feature selection becomes increasingly essential in practical applications. The motivation of feature selection is to reduce the cost of acquiring and storing features and speed up learning algorithm.

Rough set theory, proposed by Greco, Inuiguchi, and Slowinski (2006), Hu, Liu, and Yu (2008), Parthalcin and Shen (2009), Pawlak (1982), and Yang and Yang (2008, 2009), is a valid mathematical tool to deal with imprecise, uncertain, and vague information. It has been widely applied in many fields such as machine learning (Swiniarski & Skowron, 2003), data mining (Liu & Motoda, 1998), and pattern recognition (Pawlak, 1982). Feature selection is one of the most fundamental problems in rough set theory (also called attribute reduction). Researchers have proposed various approaches for feature selection (Liu & Motoda, 1998; Miao & Wang, 1997; Oh, Lee, & Moon, 2004; Wang, 2001). These approaches can be generally divided into three categories which are methods based on discernibility matrix (Wang, 2001), methods based on positive region (Wang, 2001) and methods based on information entropy (Miao & Wang, 1997). All feature selection methods are available for smaller tables. However, in the information age, data is automatically collected and therefore the database can be quite large, such as medicine data, astronomy data, the stock market data and many other areas. The growth of the size of data and number of existing databases far exceed the ability of humans to analyze this data. We may gain worse performance even get no result when dealing with large-scale data with traditional feature selection methods based on rough set theory.

The main motivation of this study is to design a method that can deal with massive and complicated real-world problems, we present a decomposition and composition method. The idea of decomposition and composition (Cheng & Wang, 2009; Maimon & Rokach, 2005; Rokach, 2006) is to break down a large and complex task into several simpler and more manageable sub-tasks that can be solved by using existing induction methods. Their results will be jointed together in the sequel in order to solve the original problem. The decomposition and composition approach can make the original task easier and less time consuming. And it is frequently used in statistics (Fischer et al., 1995), operations research (He, Strege, Tolle, & Kusiak, 2000) and engineering (Kusiak, 2000). There are a few works in data mining using decomposition and composition methodology such as decomposition and composition of incomplete information systems (Bazan, Latkowski, & Szczuka, 2006; Zhang, 2007), and decomposition and composition in multi-agent systems (Nguyen, Nguyen, & Skowron, 1999). However, some decomposition and composition methods may result in the loss of information or distortion of original data and knowledge, and can even lead to the original data mining system un-minable.

To avoid these shortcomings of decomposition and composition in data mining, we should choose the appropriate decomposition and composition method. Han and Kamber (2006) introduces multirelational data mining using keys to link multiple tables, furthermore, there is the same expression in database. There is no any the loss of information or distortion of original data and knowledge

* Corresponding author.
E-mail address: zdx.jn@163.com (N. Jiao).

0957-4174/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2010.03.039

when we convert a single table into multirelational tables. Therefore we break down a large-scale decision table into a master-table and several sub-tables. The master-table is composed of a set of decision features and several joint features which are the key words in sub-tables. The sub-table is made up of a subset of condition features. Then we join their solutions together in order to solve the problem with the original table. In order to compare with classical methods, numerous experiments can be done using some standard datasets from UCI database. Experimental results show that the proposed algorithms in this paper can improve the computational efficiency, especially to large-scale database. Finally, we discuss the complexity and the suit number of sub-tables.

The rest of the paper is organized as follows. In Section 2, basic definitions and properties are shown. In Section 3, two novel feature selection methods and an algorithm to compute the core based on decomposition and composition are introduced respectively. Some experiments and analysis are presented in Section 4. Finally, conclusions and future works are given in Section 5.

2. Basic notions

For the convenience of description, some basic definitions and properties are introduced here at first.

2.1. Basic definitions

We assume that feature selection discussed in this paper is performed in a decision table.

Definition 1 (*Decision table*). A decision table is defined as $T = \langle U, C \cup D, V, f\rangle$, where U is a non-empty finite set of objects, called universe; C is a set of all condition features (also called conditional attributes) and D is a set of decision features (also called decision attributes); $V = \bigcup_{a\in C\cup D} V_a$, V_a is a set of feature values of feature a; and $f : U \times (C \cup D) \to V$ is an information function such that $f(x, a) \in V_a$ for every $x \in U,\ a \in C \cup D$.

Definition 2 (*Equivalence relation*). Let $B \subseteq C \cup D$, B induces an equivalence (indiscernibility) relation on U as shown:

$$IND(B) = \{(x, y) \in U \times U | \forall b \in B, b(x) = b(y)\}. \quad (1)$$

Definition 3 (*Partition*). The family of all equivalence classes of $IND(B)$, i.e., the partition induced by B, is denoted as:

$$U/IND(B) = \{[x_i]_B : x_i \in U\}, \quad (2)$$

where $[x_i]_B$ is the equivalence class containing x_i. All the elements in $[x_i]_B$ are equivalent (indiscernible) with respect to B. Equivalence classes are elementary sets in rough set theory.

Definition 4 (*Lower approximation and upper approximation*). Let $X \subseteq U$ and $B \subseteq C$, the lower and upper approximations of X with respect to B, denoted by $\underline{B}X$ and $\overline{B}X$, respectively, are defined as:

$$\underline{B}X = \cup\{[x_i]_B | [x_i]_B \subseteq X\}, \quad (3)$$

$$\overline{B}X = \cup\{[x_i]_B | [x_i]_B \cap X \neq \emptyset\}. \quad (4)$$

Definition 5 (*Degree of dependency of feature*). The degree of dependency of D on C can be defined as:

$$\gamma_C(D) = |POS_C(D)|/|U|, \quad (5)$$

where $POS_C(D) = \cup_{X\in U/D} \underline{C}X$ is called the positive region of the partition U/D with respect to C, and it is the set of all samples that can be certainly classified as belonging to blocks of U/D using C.

Definition 6 (*Significance of feature*). The significance of $a \in C - B$ on the basis of B with respect to D is defined as:

$$SIG_\gamma(a, B, D) = \gamma_{B\cup\{a\}}(D) - \gamma_B(D). \quad (6)$$

Definition 7. Core$\{W_j | j \leqslant r\}$ is the set of reducts, *Core* is defined as:

$$Core = \cap_{j\leqslant r} W_j. \quad (7)$$

We break down a decision table into a master-table and several sub-tables. The master-table consists of a set of decision features and several joint features which are the keywords in sub-tables. The sub-table is composed of a subset of condition features.

Definition 8 (*Sub-table, master-table and mid-table*). Given a decision table $T = \langle U, C \cup D, V, f\rangle$.

- A sub-table is defined as $T^{B_i} = \langle U^{B_i}, B_i \cup \{b_i\}, V^{B_i}, f^{B_i}\rangle$, where U is a non-empty unique finite set of objects, called universe; $B_i \subseteq C,\ i = 1, 2, \ldots, m,\ C = \cup_{i=1}^m B_i$ and $B_i \cap B_j = \emptyset,\ i \neq j$. b_i is a joint feature which join the sub-table to the master-table and it is a keyword in T^{B_i}, $V_{b_i}^{B_i} = b_i^k,\ k = 1, 2, \ldots, p$; $V^{B_i} = \bigcup_{a\in B_i} V_a^{B_i}$, $V_a^{B_i}$ is a set of feature values of feature a; and $f^{B_i} : U^{B_i} \times B_i \to V^{B_i}$ is an information function such that $f^{B_i}(x, a) \in V_a^{B_i}$ for every $x \in U^{B_i},\ a \in B_i$.
- A master-table is defined as $T^S = \langle U, S \cup D, V^S, f^S\rangle$, where U is a non-empty finite set of objects, called universe; $S = \cup_{i=1}^m \{b_i\}$ is a set of all joint features and D is a set of decision features; $V^S = \bigcup_{a\in S\cup D} V_a^S$, V_a^S is a set of feature values of feature a; and $f^S : U \times (S \cup D) \to V^S$ is an information function such that $f^S(x, a) \in V_a^S$ for every $x \in U,\ a \in S \cup D$.
- A mid-table is defined as $T^{M_i} = \langle U, M_i \cup D, V^{M_i}, f^{M_i}\rangle,\ i = 1, 2, \ldots, m$, where U is a non-empty finite set of objects, called universe; $M_i = (S \setminus \{b_i\}) \cup B_i$ and D is a set of decision features; $V^{M_i} = \bigcup_{a\in M_i\cup D} V_a^{M_i},\ V_a^{M_i}$ is a set of feature values of feature a; and $f^{M_i} : U \times (M_i \cup D) \to V^{M_i}$ is an information function such that $f^{M_i}(x, a) \in V_a^{M_i}$ for every $x \in U,\ a \in M_i \cup D$.

Example 1. Table 1 is a decision table, we decompose it into one master-table (Table 2) and two sub-tables (Tables 3 and 4). Combine Tables 2 and 3 to compose a mid-table Table 5. Similarly, Table 6 comes from Tables 2 and 4. Their relationship is shown in Fig. 1. A decision table (Table 1), one master-table (Table 2) and two mid-tables (Tables 5 and 6) can be converted by two sub-tables (Tables 3 and 4).

Table 1
A original decision table.

U	a_1	a_2	a_3	a_4	d
1	1	1	1	0	1
2	1	0	1	1	1
3	0	0	0	1	0
4	1	0	1	0	1

Table 2
A master-table.

U	b_1	b_2	d
1	b_1^1	b_2^1	1
2	b_1^2	b_2^2	1
3	b_1^3	b_2^3	0
4	b_1^2	b_2^1	1

Table 3
The first sub-table.

b_1	a_1	a_2
b_1^1	1	1
b_1^2	1	0
b_1^3	0	0

Table 4
The second sub-table.

b_2	a_3	a_4
b_2^1	1	0
b_2^2	1	1
b_2^3	0	1

Table 5
The first mid-table.

U	a_1	a_2	b_2	d
1	1	1	b_2^1	1
2	1	0	b_2^2	1
3	0	0	b_2^3	0
4	1	0	b_2^1	1

Table 6
The second mid-table.

U	b_1	a_3	a_4	d
1	b_1^1	1	0	1
2	b_1^2	1	1	1
3	b_1^3	0	1	0
4	b_1^2	1	0	1

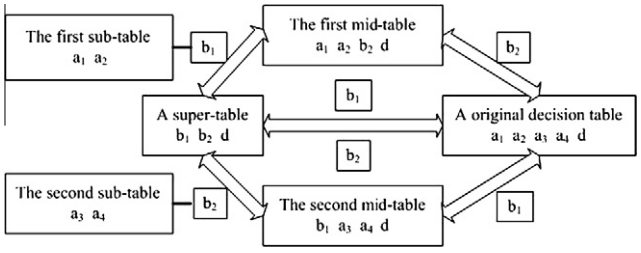

Fig. 1. The relationship of the original decision table, the master-table, the sub-tables and the mid-tables.

2.2. Basic properties

The following are some properties according to the above definition. Assume a decision system $T = \langle U, C \cup D, V, f\rangle$, sub-tables $T^{B_i} = \langle U^{B_i}, B_i \cup \{b_i\}, V^{B_i}, f^{B_i}\rangle$, $i = 1, 2, \ldots, m$, a master-table $T^S = \langle U, S \cup D, V^S, f^S\rangle$, mid-tables $T^{M_i} = \langle U, M_i \cup D, V^{M_i}, f^{M_i}\rangle$, $i = 1, 2, \ldots, m$. Some properties are described as follows.

Property 1. *The significance of $a \in C - B$ on the basis of $B \subseteq C$ with respect to D in the original decision table T is greater than zero, that is $SIG_\gamma(a, B, D) > 0$, iff the condition feature a in T is indispensable, that is $POS_{(B\cup\{a\})}(D) \neq POS_{(B)}(D)$.*

Property 2. *The significance of $a \in C - B$ on the basis of $B \subseteq C$ with respect to D in the original decision table T is greater than zero, that is $SIG_\gamma(a, B, D) > 0$, iff the condition feature a in T belongs to Core, that is $a \in Core$.*

Property 3. *The positive region in the master-table T^S is equivalent to the positive region in the original decision table T, that is $POS_{(S)}(D) = POS_{(C)}(D)$.*

Proof. The intersection of an equivalence (indiscernibility) relation is still an equivalence (indiscernibility) relation. S is a combination of C. The partition induced by C (the family of all equivalence classes) is equivalent to the partition induced by S, that is $U/IND(S) = U/IND(C)$, therefore $POS_{(S)}(D) = POS_{(C)}(D)$. □

Corollary 1. *The positive region in the mid-table T^{M_i} is equivalent to the positive region in the original decision table T, that is $POS_{(M_i)}(D) = POS_{(C)}(D)$.*

Property 4. *The joint feature b_i in the master-table T^S is dispensable, that is $POS_{(S\setminus\{b_i\})}(D) = POS_{(S)}(D)(SIG_\gamma(b_i, S - \{b_i\}, D) = 0)$, iff the condition feature set B_i in the original decision table T corresponding to the joint feature b_i is dispensable, that is $POS_{(C\setminus B_i)}(D) = POS_{(C)}(D)$ $(SIG_\gamma(B_i, C - B_i, D) = 0)$.*

Proof. According to Definition 8, b_i is a combination of B_i. The partition induced by $S - \{b_i\}$ (the family of all equivalence classes) is equivalent to the partition induced by $C - B_i$, that is $U/IND(S - \{b_i\}) = U/IND(C - B_i)$, therefore $POS_{(S\setminus\{b_i\})}(D) = POS_{(C\setminus B_i)}(D)$. According to Property 3, $POS_{(S)}(D) = POS_{(C)}(D)$. Therefore if $POS_{(S\setminus\{b_i\})}(D) = POS_{(S)}(D)$, then $POS_{(C\setminus B_i)}(D) = POS_{(C)}(D)$. And vice versa. □

Corollary 2. *The condition feature a in the mid-table T^{M_i} is dispensable, that is $\exists a \in B_i$, $POS_{(M_i\setminus\{a\})}(D) = POS_{(M_i)}(D)(SIG_\gamma(a, M_i - \{a\}, D) = 0)$, iff the condition feature a in the original decision table T is dispensable, that is $POS_{(C\setminus\{a\})}(D) = POS_{(C)}(D)(SIG_\gamma(a, C - \{a\}, D) = 0)$.*

Corollary 3. *If the joint feature b_i in the master-table T^S is indispensable (a core feature), that is $POS_{(S\setminus\{b_i\})}(D) \neq POS_{(S)}(D)$ $(SIG_\gamma(b_i, S - \{b_i\}, D) > 0)$, then a subset A in the condition feature set B_i corresponding to the joint feature b_i is indispensable (core features) in the original decision table T, that is $A \subseteq B_i$, $POS_{(C\setminus A)}(D) \neq POS_{(C)}(D)(SIG_\gamma(A, C - A, D) > 0)$.*

Corollary 4. *If the condition feature a in the mid-table T^{M_i} is indispensable (a core feature), that is $POS_{(M_i\setminus\{a\})}(D) \neq POS_{(M_i)}(D)$ $(SIG_\gamma(a, M_i - \{a\}, D) > 0)$, then the condition feature a in the original decision table T is indispensable (a core feature), that is $POS_{(C\setminus\{a\})}(D) \neq POS_{(C)}(D)(SIG_\gamma(a, C - \{a\}, D) > 0)$.*

These properties will be applied in following methods.

3. The feature selection methods based on decomposition and composition

In this section we introduce the strategy of decomposition and composition at first. Then two feature selection methods and an approach of computing core based on decomposition and composition are proposed.

Finding all reducts is NP-hard problem. However, it is usually enough for most practical applications to find one of the reducts. The feature selection methods of this paper are to find a reduct.

3.1. The strategy of decomposition and composition

The strategy of decomposition and composition can affect the efficiency dramatically. If there is not an appropriate strategy of decomposition and composition, some decomposition and composition methods may result in the loss of information or distortion of original data and knowledge, and can even lead to the original data mining system un-minable.

In general, the number of the condition features is relatively large comparing to the number of the decision features, therefore we only decompose the condition features into sub-tables and do not process the decision features. If there are a lot of decision features, we can break them down into sub-tables similarly. In this paper, our strategy of decomposition and composition is to decompose the condition features into several subsets and connect these subsets with some new features called joint features. The joint features and the set of decision features form the master-table. Every subset and the joint feature construct the sub-table. In the algorithm, we combine each sub-table with the master-table to compose a mid-table. Actually, we only combine condition features and do not change data table itself. So our methods will not lead to loss of data or incorrect information after decomposition and composition. The approach of decomposition for conditional features is also different. Two strategies are described as follows.

- *Random strategy*: The random strategy is that the condition features of the original decision table are divided equally among sub-tables randomly.
- *Heuristic strategy*: Firstly, we compute the significance of condition features. If there are indispensable condition features, they are put into one sub-table and the remaining condition features are decomposed equally into other sub-tables. Otherwise they are divided equally among sub-tables randomly.

3.2. The feature selection method based on decomposition and composition-random strategy

We employ random strategy to design a reduction algorithm. Suppose that the number of sub-tables is k. First, we break the original decision table down into one master-table and k sub-tables. The condition features of the original decision table are divided equally among k sub-tables. The joint feature and a subset of condition features compose a sub-table. The master-table is made up of a set of decision features and k joint features that are the key words in sub-tables.

Then if the joint feature in master-table is dispensable, we can delete the joint feature in master-table and combine the same objects (Properties 3 and 4) to decide the next joint feature. Otherwise, we combine a sub-table with the master-table to compose a mid-table, if the condition feature in the mid-table is dispensable, we can delete the condition feature in the mid-table and combine the same objects (Corollaries 1 and 2), or else continue the next loop. Finally, a reduction can be found.

We show the feature selection method based on decomposition and composition-random strategy in Algorithm 1.

Algorithm 1. Feature selection method based on decomposition and composition-random strategy (FSDC-RS)

Input: A decision table $T = \langle U, C \cup D, V, f\rangle$. The number of sub-tables is k.
Output: Feature selection $RED_{(D)}(C)$.

1. Break T down into one master-table $T^S = \langle U, S \cup D, V^S, f^S\rangle$ and sub-tables $T^{B_i} = \langle U^{B_i}, B_i \cup \{b_i\}, V^{B_i}, f^{B_i}\rangle,\ i = 1, 2, \ldots, k$.
2. $RED_{(D)}(C) \leftarrow C,\ i \leftarrow 1$.
3. **While** $i <= k$ **do**
4. **Begin**
5. **If** $POS_{(S\setminus\{b_i\})}(D) = POS_{(S)}(D)$, **then**
6. **Begin**
7. $S \leftarrow S - \{b_i\}$;
8. $RED_{(D)}(C) \leftarrow RED_{(D)}(C) - B_i$;
9. Combine the same objects;
10. **End**
11. **Else**
12. **Begin**
13. Combine T^{M_i} with T^{B_i} and T^S, $j \leftarrow card(B_i)$;
14. **While** $j > 0$ **do**
15. **Begin**
16. **If** $\exists a \in B_i,\ POS_{(M_i\setminus\{a\})}(D) = POS_{(M_i)}(D)$, **then**
18. **Begin**
19. $B_i \leftarrow B_i - \{a\}$;
20. $RED_{(D)}(C) \leftarrow RED_{(D)}(C) - \{a\}$;
21. Combine the same objects;
22. **End**
23. $j \leftarrow j - 1$;
24. **End**
25. **End**
26. $i \leftarrow i + 1$
27. **End**
28. **Return** $RED_{(D)}(C)$.

Actually, the condition features of a decision table are divided into several parts. We process every part instead of every condition feature. Every part is substituted by a joint feature. In other words, $|C|$ condition features of a decision table are compressed to k joint features of the master-table. If the joint feature is dispensable, the condition feature set corresponding to the joint feature is dispensable and can be deleted once. Each feature in this condition feature set does not need to be checked again. Even though the joint feature is indispensable, we need to convert a master-table and a sub-table into a mid-table, the scale of the mid-table is reduced a lot. Compare with $|C|$, the number of features of the mid-table is $(|C|/k) + k$, which is very small. Hence our methods achieved significant saving on the computation time. In the best case, we can get a reduction only by checking k joint features of the master-table. The minimum time complexity of FSDC-RS is $O(|N|^2 * (k + |D|)^2)(k \ll |C|)(|N|$ is the number of objects, $|C|$ is the number of condition features and $|D|$ is the number of decision features). In the worst case, we have to process $(|C|k) + k$ features of the mid-table to achieve a reduction. The maximum time complexity of FSDC-RS is $O(|N|^2 * ((|C|k + k) + |D|)^2)$. The average time complexity of traditional methods is $O(|N|^2 * (|C| + |D|)^2)$. The maximum time complexity of FSDC-RS is far less than the average time complexity of traditional methods. Therefore, the performance of FSDC-RS is better than other traditional approaches.

3.3. The feature selection method based on decomposition and composition-heuristic strategy

However, the probability of deleting joint features once is low on account of the absence of heuristic information,

thus we present another feature selection method based on decomposition and composition using heuristic strategy. In this paper, the significance of features is taken into account heuristic information. We calculate the significance of condition features. The condition features whose significances are greater than zero are indispensable, and they are core features at the same time. We put the indispensable features in one sub-table. The remainder is equally divided into other sub-tables.

First of all, we compute the significance of condition features in original decision table to find indispensable condition features. According to Property 2, the indispensable condition features are core features. The traditional methods of core consume a lot of time. Hence we introduce a novel method of calculating core based on decomposition and combination.

We assume that the number of sub-tables is k. We decompose the original decision table into one master-table and k sub-tables. The condition features of the original decision table are divided equally among k sub-tables. If the joint feature in master-table is indispensable (Properties 1–3 and Corollary 3), we compose a mid-table with a sub-table and the master-table. If the condition feature in mid-table is indispensable (Corollaries 1 and 4), it is a core feature, or else continues the next loop. At last, we get all the core features.

The method of calculating core based on decomposition and composition is shown in Algorithm 2.

Algorithm 2. The method of computing core based on decomposition and composition (CCDC)

Input: A decision table $T = \langle U, C \cup D, V, f\rangle$. The number of sub-tables is k.
Output: Core denoted by $CORE_{(D)}(C)$.

1. Break T down into one master-table $T^S = \langle U, S \cup D, V^S, f^S\rangle$ and sub-tables $T^{B_i} = \langle U^{B_i}, B_i \cup \{b_i\}, V^{B_i}, f^{B_i}\rangle,\ i = 1, 2, \ldots, k$.
2. $CORE_D(C) = \emptyset,\ i \leftarrow 1$.
3. **While** $i <= k$ **do**
4. **Begin**
5. **If** $SIG_\gamma(b_i, S - \{b_i\}, D) > 0$, **then**
6. **Begin**
7. Compose T^{M_i} with T^{B_i} and $T^S,\ j \leftarrow card(B_i)$;
8. **While** $j > 0$ **do**
9. **Begin**
10. **If** $\exists a \in B_i,\ SIG_\gamma(a, M_i - \{a\}, D) > 0$, **then**
11. $CORE_D(C) \leftarrow CORE_D(C) \cup \{a\}$;
12. $j \leftarrow j - 1$;
13. **End**
14. **End**
15. $i \leftarrow i + 1$
16. **End**
17. **Return** $CORE_{(D)}(C)$.

According to the above computing core method, core is put into the first sub-table and others are decomposed equally into $k - 1$ sub-tables. Initial condition starts at the second joint feature. Repeat the same procedure as FSDC-RS algorithm, all selected features consist of a feature selection.

As analyzed above, we have the following algorithm for feature selection based on decomposition and composition using heuristic strategy.

Algorithm 3. Feature selection method based on decomposition and composition-heuristic strategy (FSDC-HS)

Input: A decision table $T = \langle U, C \cup D, V, f\rangle$. The number of sub-tables is k.
Output: Feature selection $RED_{(D)}(C)$.

1. According to CCDC algorithm, calculate core in T.
2. **If** $CORE_D(C) \neq \emptyset$, **then**
3. **Begin**
4. **If** $POS_{(B_1)}(D) \neq POS_{(C)}(D)$, **then**
5. **Begin**
6. $B_1 \leftarrow CORE_{(D)}(C)$
7. Break T down into one master-table $T^S = \langle U, S \cup D, V^S, f^S\rangle$ and sub-tables $T^{B_i} = \langle U^{B_i}, B_i \cup \{b_i\}, V^{B_i}, f^{B_i}\rangle,\ i = 1, 2, \ldots, k$.
8. $i \leftarrow 2$.
9. **End**
10. **Else**
11. **Begin**
12. $RED_{(D)}(C) \leftarrow CORE_{(D)}(C)$.
13. **Return** $RED_{(D)}(C)$.
14. **End**
15. **End**
16. **Else**
17. **Begin**
18. Break T down into one master-table $T^S = \langle U, S \cup D, V^S, f^S\rangle$ and sub-tables $T^{B_i} = \langle U^{B_i}, B_i \cup \{b_i\}, V^{B_i}, f^{B_i}\rangle,\ i = 1, 2, \ldots, k$.
19. $i \leftarrow 1$.
20. **End**
21. $RED_{(D)}(C) \leftarrow C$.
22. **While** $i <= k$ **do**
23. **Begin**
24. **If** $POS_{(S\setminus\{b_i\})}(D) = POS_{(S)}(D)$, **then**
25. **Begin**
26. $S \leftarrow S - \{b_i\}$;
27. $RED_{(D)}(C) \leftarrow RED_{(D)}(C) - B_i$;
28. Combine the same objects;
29. **End**
30. **Else**
31. **Begin**
32. Compose T^{M_i} with T^{B_i} and $T^S, j \leftarrow card(B_i)$;
33. **While** $j > 0$ **do**
34. **Begin**
35. **If** $\exists a \in B_i, POS_{(M_i\setminus\{a\})}(D) = POS_{(M_i)}(D)$, **then**
36. **Begin**
37. $B_i \leftarrow B_i - \{a\}$;
38. $RED_{(D)}(C) \leftarrow RED_{(D)}(C) - \{a\}$;
39. Combine the same objects;
40. **End**
41. $j \leftarrow j - 1$;
42. **End**
43. **End**
44. $i \leftarrow i + 1$
45. **End**
46. **Return** $RED_{(D)}(C)$.

Clearly, FSDC-HS is almost the same as FSDC-RS algorithm except the decomposition strategy. Although the chance of deleting joint features of FSDC-HS once is higher than FSDC-RS algorithm, the procedure of computing core will increase the computation time. The time complexity of FSDC-HS involves two parts which are the time complexity of CCDC and time complexity of rest pro-

cedures. In the best case, there is no core feature in CCDC algorithm and a reduction can be obtained only by checking k joint features of the master-table in the rest procedures. The minimum time complexity of FSDC-HS is $O(|N|^2 * (k+|D|)^2 + |N|^2 * (k+|D|)^2) = O(2 * (|N|^2 * (k+|D|)^2)) \approx O(|N|^2 * (k+|D|)^2)(k \ll |C|)$ ($|N|$ is the number of objects, $|C|$ is the number of condition features and $|D|$ is the number of decision features). In the worst case, there are core features in each sub-table of CCDC and we have to check $(|C|/k)+k$ features of the mid-table to achieve a reduction in remaining steps. The maximum time complexity of FSDC-HS is $O(|N|^2 * (((|C|/k+k)+|D|)^2 + |N|^2 * (((|C|/k+k)+|D|)^2) = O(2 * (|N|^2 * (((|C|/k+k)+|D|)^2)) \approx O(|N|^2 * (((|C|/k+k)+|D|)^2)$. Similarly, the average time complexity $O(|N|^2 * (|C|+|D|)^2)$ of traditional methods is higher than the maximum time complexity of FSDC-HS. Therefore, FSDC-HS can get better performance than other classical methods.

If distributed and parallel technique is adopted, the running time of our methods can be shortened.

4. Experiments

In this section, we show that our feature selection methods based on decomposition and composition can reduce the computation complexity significantly. Firstly, we evaluate the proposed methods by comparing with four traditional methods on seven various datasets from UCI database (these datasets can be downloaded at http://www.ics.uci.edu), for which we show our methods can ease the computation complexity on different datasets. The second experiment uses Insurance-Company-Benchmark dataset with different features. We perform the experiment to discover the tendency of running time for different methods as features are increased gradually. In the third experiment, we select different objects on Connect-4 dataset. We repeat the experiment to find the trend of execution time for several approaches as objects are added one by one. The last subsection contains the analysis on the suit number of sub-tables.

4.1. A comparative experiment on seven datasets

In order to test the validity of the algorithm, we compare the proposed methods with four classical algorithms for feature selection. They are described as follows: General feature selection algorithm (General); Feature selection algorithm based on positive region (Positive) (computing core firstly and appending the most important feature according to significance of features until achieving reduction); Feature selection algorithm based on information entropy (Entropy); Feature selection algorithm based on discernibility matrix (Matrix). According to FSDC-RS and FSDC-HS algorithms of this paper, we suppose the number of sub-tables is four. We perform the experiments on publicly available datasets from UCI database (these datasets can be downloaded at http://www.ics.uci.edu). Datasets Breast-Cancer-Wisconsin (Diagnostic), SPECT-Heart, Ayduikigt-standard, Madelon, Connect-4, Optdigits and Insurance-Company-Benchmark are used to test. The experiment results are shown in Table 7. The leftmost column consists of dataset names. A brief description is below the datasets (C: condition feature; D: decision feature; O: object). The rest columns present running time of different algorithms (its unit is second and the abbreviation is S) which is average of repeating 10 times experiments. ∞ means that the running time is more than 43,200 s (12 h).

When there are missing values in datasets, these values are filled with mean values for continuous features and majority values for nominal features (Grzymala-Busse & Grzymala-Busse, 2005). If the datasets are numerical, all continuous features are discretized using Equal Frequency per Interval (Grzymala-Busse, 2002).

As listed in Table 7, general feature selection algorithm (General) outperforms other three classic feature selection methods. The performance of feature selection algorithm based on positive region (Positive) is worse than that of general feature selection algorithm (General). The performance of feature selection algorithm based on information entropy (Entropy) is the worst. Feature selection algorithm based on discernibility matrix (Matrix) is less time consuming for small dataset while this algorithm gain worse performance even get no result for large-scale dataset. FSDC-RS and FSDC-HS have been shown to be superior to other methods. At the same time, by Table 7, we can also observe that FSDC-RS algorithm has a higher computational performance as compared to FSDC-HS algorithm, which is analyzed in Section 3. The procedure of computing core of FSDC-HS algorithm will increase the computation time.

4.2. An experiment on Insurance-Company-Benchmark dataset with different features

The second experiment is performed on Insurance-Company-Benchmark dataset which has 86 features and 9822 objects. We select bottom 20, 30, 40, 50, 60, 70 and 86 features from this dataset respectively. We use four classical approaches and the feature selection algorithms given in this paper to test these data. According to our two methods we break the datasets down into one master-table and four sub-tables.

From Fig. 2, we can see the comparison of efficiencies of various methods as features increasing gradually. As depicted in Fig. 2, FSDC-RS and FSDC-HS outperform other methods. The running

Table 7
Comparison of efficiencies of different feature selection algorithms.

Dataset	General	Positive	Entropy	Matrix	FSDC-RS	FSDC-HS
Audiology (Standardized) (69C,1D,226O)	12S	44S	286S	14S	9S	10S
Breast-Cancer-Wisconsin (Diagnostic) (31C,1D,569O)	3S	6S	45S	200S	2S	3S
Connect-4 (42C,1D,67557O)	287S	1651S	∞	∞	111S	131S
Insurance-Company-Benchmark (COIL 2000) (85C,1D,9822O)	190S	1706S	∞	∞	44S	64S
Madelon (499C,1D,4400O)	∞	∞	∞	∞	41S	55S
Optical Recognition of Handwritten Digits (64C,1D,1796O)	15S	44S	∞	∞	4S	10S
SPECT-Heart (44C,1D,267O)	6S	15S	109S	6S	1S	3S

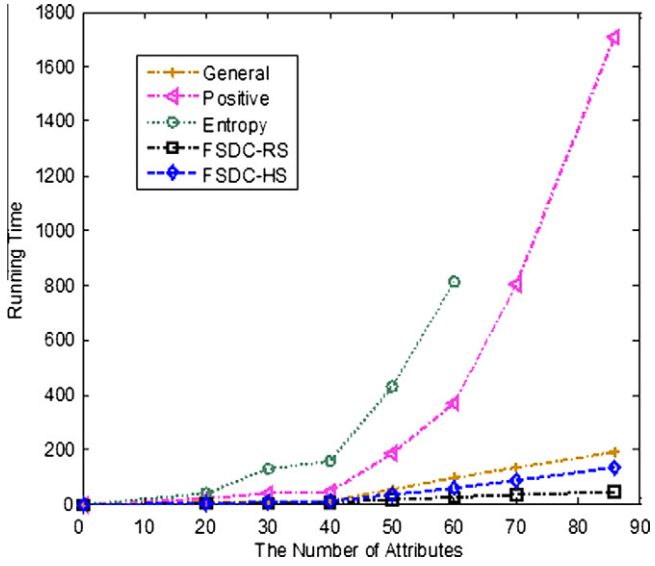

Fig. 2. The comparison of performances of different features on Insurance-Company-Benchmark dataset.

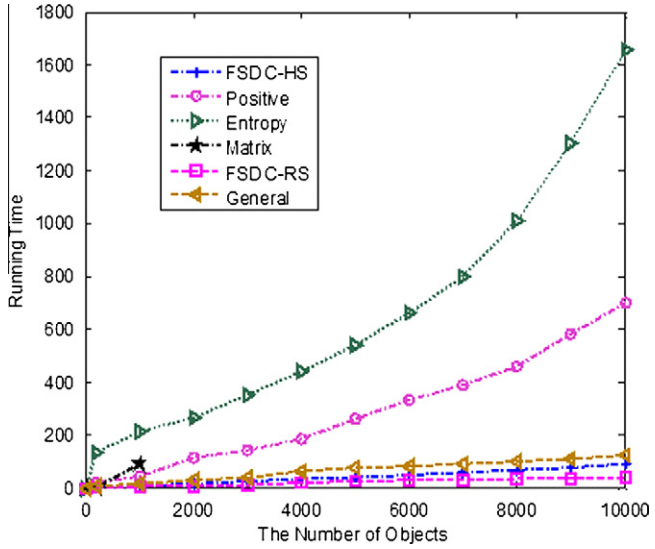

Fig. 3. The comparison of performances of different objects on Connect-4 dataset.

time of our methods increases slightly as features increasing gradually. However, other methods consume much more time. There is no execution time of feature selection algorithm based on discernibility matrix (Matrix) because it is always equal to ∞. The time of feature selection algorithm based on information entropy (Entropy) is ∞ when the number of features is greater than 70.

4.3. An experiment on Connect-4 dataset with different objects

We do another experiment on Connect-4 dataset which has 43 features and 67,557 objects. We select top 200, 2000, 4000, 6000, 8000 and 10,000 objects from this dataset. Four classic approaches and our methods are used to test these data. The number of sub-tables is the same as the above experiments.

Fig. 3 shows the comparison of efficiencies of various algorithms based on different size of objects. As depicted in Fig. 3, FSDC-RS and FSDC-HS can achieve better performance than other

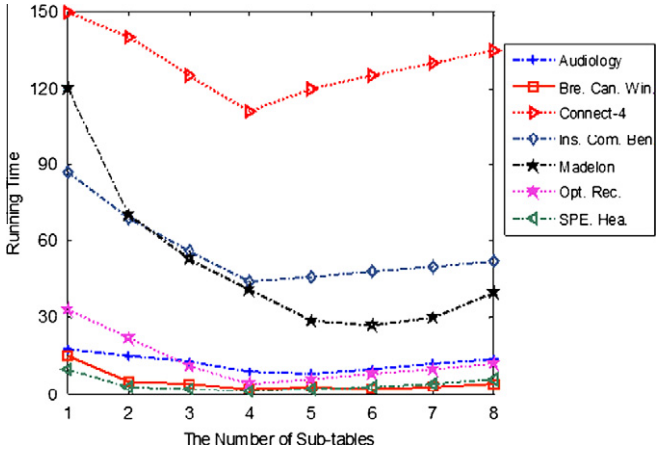

Fig. 4. The comparison of performances of different sub-tables on seven datasets with FSDC-RS.

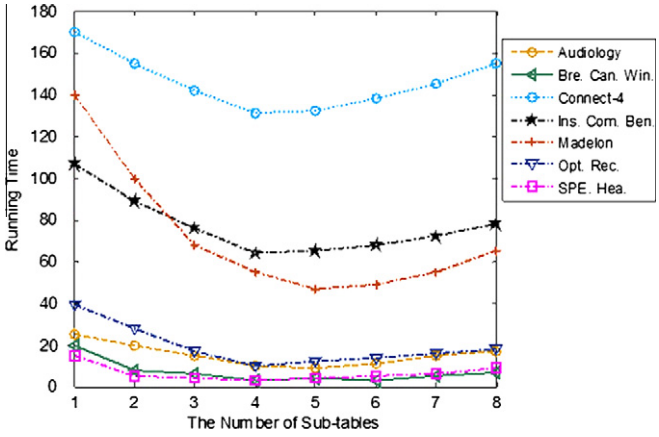

Fig. 5. The comparison of performances of different sub-tables on seven datasets with FSDC-HS.

methods. The execution time of feature selection algorithm based on discernibility matrix (Matrix) is ∞ when the number of objects is more than 2000.

4.4. Analysis of experiment results

In this subsection, we empirically discuss the suit number of sub-tables. The number of sub-tables will affect the efficiency. The seven datasets in Section 4.1 are used to test again. We repeatedly evaluate the methods of this paper with different number of sub-tables. The experiment results are shown in Figs. 4 and 5.

As shown in Figs. 4 and 5, we find the number of sub-tables cannot be neither too large nor too small. At the same time, the execution time is less when the number of sub-tables is between 3 and 7. Although we cannot decide the most suit number of sub-tables at the beginning, the running time of our methods is relatively small compared to other four classic feature selection approaches (Table 7).

5. Conclusions

Feature selection is an important task in rough set theory. Existing methods do not perform very well on large datasets. In this paper, we introduce some novel decomposition and composi-

tion methods for rough set feature selection and core calculation. The purpose of decomposition and composition is to break a complex table down into smaller, simpler and more manageable sub-tables that are solvable by using existing methods, then joining them together to solve the initial table. In order to test the validity of the algorithm, we have done numerous experiments. Firstly, we compare the methods of this paper with four classic methods on seven standard datasets from UCI database. Secondly, we use Insurance-Company-Benchmark dataset with different features. And different objects on Connect-4 dataset are selected to perform the third experiment. Experimental results demonstrate that our methods are efficient for various datasets. Finally, a discussion about the suit number of sub-tables is shown.

There are two directions for future work. The first one is to develop other efficient algorithms based on decomposition and composition for feature selection. The second one is to focus on how to exactly make sure the number of sub-tables.

Acknowledgements

This paper is supported by The National Natural Science Foundation of PR China (Nos. 60475019, 60775036) and The Research Fund for the Doctoral Program of Higher Education (No. 20060247039).

References

Bazan, J. G., Latkowski, R., & Szczuka, M. (2006). Missing template decomposition method and its implementation in rough set exploration system. In *Proceedings of the fifth international conference on rough sets and current trends in computing, Kobe, Japan* (pp. 254–263).

Cheng, C. B., & Wang, K. P. (2009). Solving a vehicle routing problem with time windows by a decomposition technique and a genetic algorithm. *Expert Systems with Applications, 36*, 7758–7763.

Fischer, B. (1995). *Decomposition of time series-comparing different methods in theory and practice*. Eurostat Working Paper.

Greco, S., Inuiguchi, M., & Slowinski, R. (2006). Fuzzy rough sets and multiple-premise gradual decision rules. *International Journal of Approximate Reasoning, 41*(2), 179–211.

Grzymala-Busse, J. W. (2002). Discretization of numerical attributes. In W. Klösgen & J. Zytkow (Eds.), *Handbook of data mining and knowledge discovery* (pp. 218–225). Oxford University Press.

Grzymala-Busse, J. W., & Grzymala-Busse, W. J. (2005). Handling missing attribute values. In O. Maimon & L. Rokach (Eds.), *Handbook of data mining and knowledge discovery* (pp. 37–57).

Guyon, E. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research, 3*, 1157–1182.

Han, J. W., & Kamber, M. (2006). *Data mining: Concepts and techniques* (2nd ed.). Morgan Kaufman.

He, D. W., Strege, B., Tolle, H., & Kusiak, A. (2000). Decomposition in automatic generation of petri nets for manufacturing system control and scheduling. *International Journal of Production Research, 38*(6), 1437–1457.

Hu, Q. H., Liu, J. F., & Yu, D. (2008). Mixed feature selection based on granulation and approximation. *Knowledge-Based Systems, 21*, 294–304.

Kusiak, A. (2000). Decomposition in data mining: An industrial case study. *IEEE Transactions on Electronics Packaging Manufacturing, 23*(4), 345–353.

Liu, H., & Motoda, H. (1998). *Feature Selection for Knowledge Discovery and Data Mining*. Boston: Kluwer.

Maimon, O., & Rokach, L. (2005). *Decomposition methodology for knowledge discovery and data mining: Theory and applications*. World Scientific.

Miao, D. Q., & Wang, J. (1997). Information-based algorithm for reduction of knowledge. In *IEEE international conference on intelligent processing systems* (pp. 1155–1158).

Nguyen, H. S., Nguyen, S. H., & Skowron, A. (1999). Decomposition of task specification problems. In *Proceedings of the 11th international symposium on foundations of intelligent systems, Warsaw, Poland* (pp. 310–318).

Oh, I. S., Lee, J. S., & Moon, B. R. (2004). Hybrid genetic algorithms for feature selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 26*(11), 1424–1437.

Parthalcin, N. M., & Shen, Q. (2009). Exploring the boundary region of tolerance rough sets for feature selection. *Pattern Recognition, 42*, 655–667.

Pawlak, Z. (1982). Rough sets. *International Journal of Computer and Information Sciences, 11*(5), 341–356.

Rokach, L. (2006). Decomposition methodology for classification tasks: A meta decomposer framework. *Pattern Analysis and Applications, 9*, 257–271.

Swiniarski, R. W., & Skowron, A. (2003). Rough set methods in feature selection and recognition. *Pattern Recognition Letters, 24*, 833–849.

Wang, G. Y. (2001). *Rough Set Theory and Knowledge Acquisition*. Xi'an: Xi'an Jiaotong University Press [in Chinese].

Yang, M., & Yang, P. (2008). A novel condensing tree structure for rough set feature selection. *Neurocomputing, 71*, 1092–1100.

Yao, Y. Y., & Zhao, Y. (2009). Discernibility matrix simplification for constructing attribute reducts. *Information Sciences, 179*, 867–882.

Yu, L., & Liu, H. (2004). Efficient feature selection via analysis of relevance and reducdancy. *Journal of Machine Learning Research, 5*, 1205–1224.

Zhang, Q. Z. (2007). An approach to rough set decomposition of incomplete information systems. In *IEEE conference on industrial electronics and applications, Harbin, China* (pp. 2455–2460).