# A Parallel Implementation of Computing Composite Rough Set Approximations on GPUs

Junbo Zhang[1,2], Yun Zhu[2], Yi Pan[2], and Tianrui Li[1,⋆]

[1] School of Information Science and Technology, Southwest Jiaotong University, Chengdu 610031, China
[2] Department of Computer Science, Georgia State University, Atlanta, GA 30303, USA
JunboZhang86@163.com, yzhu7@student.gsu.edu, pan@cs.gsu.edu, trli@swjtu.edu.cn

**Abstract.** In information systems, there may exist multiple different types of attributes like categorical attributes, numerical attributes, set-valued attributes, interval-valued attributes, missing attributes, *etc.* Such information systems are called as composite information systems. To process such attributes with rough set theory, composite rough set model and corresponding matrix methods were introduced in our previous research. Rough set approximations of a concept are the basis for rule acquisition and attribute reduction in rough set based methods. To accelerate the computation process of rough set approximations, this paper first presents the boolean matrix representation of the lower and upper approximations in the composite information system, then designs a parallel method based on matrix, and implements it on GPUs. The experiments on data sets from UCI and user-defined data sets show that the proposed method can accelerate the computation process efficiently.

**Keywords:** Composite Rough Sets, Boolean Matrix, GPU, CUDA.

## 1 Introduction

The rough set (RS) theory is a powerful mathematical tool to describe the dependencies among attributes, evaluate the significance of attributes, and derive decision rules [15]. It plays an important role in the fields of data mining and machine learning [7,18,21,22]. Different attributes can be processed by different rough set models. For example, Hu et al. generalized classical rough set model with neighborhood relations to deal with numerical attributes [7]. Guan et al. defined the tolerance relation and used it to deal with set-valued attributes [5]. Grzymała-Busse integrated the tolerance relation [8] and the similarity relation [19], and proposed the characteristic relation [4] for missing attributes in incomplete information systems. In real-applications, there are multiple different types of attributes in information systems like categorical attributes, numerical

⋆ Corresponding author.

P. Lingras et al. (Eds.): RSKT 2013, LNAI 8171, pp. 240–250, 2013.

attributes, set-valued attributes, and missing attributes. Such information systems are called as composite information systems. Most of rough set models fail to deal with more than two types of attributes. To solve this problem, we gave the composite rough set model, defined a composite relation and used composite classes to drive approximations from composite information systems in our previous work [23]. Table 1 shows these rough set models for different types of attributes.

**Table 1.** Rough Set Models

| Model | Relation | Data Types | | | |
|---|---|---|---|---|---|
| | | C | N | S | M |
| Classical RS | Equivalence [15] | $\surd$ | $\times$ | $\times$ | $\times$ |
| Neighborhood RS | Neighborhood [7] | $\surd$ | $\surd$ | $\times$ | $\times$ |
| Set-valued RS | Tolerance [5] | $\surd$ | $\times$ | $\surd$ | $\times$ |
| Characteristic RS | Characteristic [3] | $\surd$ | $\times$ | $\times$ | $\surd$ |
| Composite RS | Composite [23] | $\surd$ | $\surd$ | $\surd$ | $\surd$ |

C: Categorical, N: Numerical, S: Set-valued, M: Missing

Rough set approximations of a concept are the basis for rule acquisition and attribute reduction in rough set based methods. The efficient calculation of rough set approximations can accelerate the process of knowledge discovery effectively. Parallelization of algorithms is a good way to speed up the computational process. In our previous work, we proposed a parallel algorithm for computing rough set approximation [21], however, it can only process categorical attributes. In this paper, to deal with composite attributes and compute rough set approximations from composite information systems, we give a boolean-based method for computing rough set approximations. It means that the calculation of rough set approximations can be processed as boolean matrix operations. We design the parallel matrix-based method and parallelize it on GPUs, which have recently been utilized in various domains, including high-performance computing [14]. NVIDIA GPUs [1] power millions of desktops, notebooks, workstations and supercomputers around the world, and accelerate computationally-intensive tasks for professionals, scientists, researchers, etc. NVIDIA CUDA [2] is a General Purpose Computation on GPUs (GPGPUs) framework, which uses a C-like programming language and does not require re-mapping algorithms to graphics concepts. These features help users develop correct and efficient GPU programs easily.

The remainder of the paper is organized as follows. Section 2 introduces some rough set models. Section 3 proposes the boolean matrix-based method. Section 4 designs the parallel method for computing composite rough set (CRS) approximations. Section 5 gives the experimental analysis. The paper ends with conclusions and future work in Section 6.

## 2 Rough Set Model

In this section, we first briefly review the concepts of rough set model as well as their extensions [5,7,10,15,16,23].

### 2.1 Classical Rough Set Model

Given a pair $K = (U, R)$, where $U$ is a finite and non-empty set called the universe, and $R \subseteq U \times U$ is an indiscernibility relation on $U$. The pair $K = (U, R)$ is called an approximation space. $K = (U, R)$ is characterized by an information system $IS = (U, A, V, f)$, where $U$ is a non-empty finite set of objects; $A$ is a non-empty finite set of attributes; $V = \bigcup_{a \in A} V_a$ and $V_a$ is a domain of attribute $a$; $f : U \times A \to V$ is an information function such that $f(x, a) \in V_a$ for every $x \in U$, $a \in A$. In the classical rough set model, $R$ is the equivalence relation. Let $B \subseteq A$ and $[x]_{R_B}$ denote an equivalence class of an element $x \in U$ under the indiscernibility relation $R_B$, where $[x]_{R_B} = \{y \in U | xR_By\}$.

Classical rough set model is based on the equivalence relation. The elements in an equivalence class satisfy reflexive, symmetric and transitive. It also cannot deal with the non-categorical attributes like numerical attributes, set-valued attributes, etc. However, non-categorical attributes appear frequently in real applications [3,6,8,17]. Therefore, it is necessary to investigate the situation of non-categorical attributes in information systems. In what follows, we just introduce two rough set models [5,7], which will be used in our examples. More rough set models for dealing with non-categorical attributes are available in the literatures [3,6,8,9,12,17,20].

### 2.2 Composite Rough Set (CRS) Model

In many practical issues, there are multiple different types of attributes in information systems, called composite information systems. A composite information system can be written as $CIS = (U, A, V, f)$, where
(i) $U$ is a non-empty finite set of objects;
(ii) $A = \bigcup A_k$ is a union of attribute sets, and $A_k$ is an attribute set with the same type of attributes;
(iii) $V = \bigcup_{A_k \subseteq A} V_{A_k}$, $V_{A_k} = \bigcup_{a \in A_k} V_a$, $V_a$ is a domain of attribute $a$;
(iv) $f : U \times A \to V$, namely, $U \times \bigcup A_k \to \bigcup V_{A_k}$, and $U \times A_k \to V_{A_k}$ is an information function, $f(x, a)$ denotes the value of object $x$ on attribute $a$.

**Definition 1.** [23] *Given $x, y \in U$ and $B = \bigcup B_k \subseteq A$, $B_k \subseteq A_k$, the composite relation $CR_B$ is defined as*

$$CR_B = \{(x, y) | (x, y) \in \bigcap_{B_k \subseteq B} R_{B_k}\} \tag{1}$$

*where $R_{B_k} \subseteq U \times U$ is an indiscernibility relation defined by an attribute subset $B_k$ on $U$* [16].

When $(x, y) \in CR_B$, we call $x$ and $y$ are indiscernible w.r.t. $B$. Let $CR_B(x) = \{y | y \in U, \forall B_k \in B, yR_{B_k}x\}$, we call $CR_B(x)$ the composite class for $x$ w.r.t. $CR_B$.

**Definition 2.** [23] *Given a composite information system $CIS = (U, A, V, f)$, $\forall X \subseteq U$, $B \subseteq A$, the lower and upper approximations of $X$ in terms of composite relation $CR_B$ are defined as*

$$\underline{CR_B}(X) = \{x \in U | CR_B(x) \subseteq X\} \tag{2}$$

$$\overline{CR_B}(X) = \{x \in U | CR_B(x) \cap X \neq \emptyset\} \tag{3}$$

## 3 Boolean Matrix-Based Method

In this section, we present the boolean matrix representation of the lower and upper approximations in the composite information system. Before this, we review the matrix-based approaches in rough set model. A set of axioms were constructed to characterize classical rough set upper approximation from the matrix point of view by Liu [11]. Zhang et al. defined a basic vector $H(X)$, which was induced from the relation matrix. And four cut matrices of $H(X)$, denoted by $H^{[\mu,\nu]}(X)$, $H^{(\mu,\nu]}(X)$, $H^{[\mu,\nu)}(X)$ and $H^{(\mu,\nu)}(X)$, were derived for the computation of approximations, positive, boundary and negative regions intuitively in set-valued information systems [24]. Furthermore, Zhang et al. gave the matrix-based methods for computing rough set approximations in composite information systems [23]. Here, we follow their work and give a novel boolean matrix-based method to process composite data.

### 3.1 Boolean Matrix-Based Method in the Composite Information System

**Definition 3.** [11] *Let $U = \{x_1, x_2, \ldots, x_n\}$, and $X$ be a subset of $U$. The characteristic function $G(X) = (g_1, g_2, \ldots, g_n)^T$ ($T$ denotes the transpose operation) is defined as*

$$g_i = \begin{cases} 1, & x_i \in X \\ 0, & x_i \notin X \end{cases} \tag{4}$$

*where $G(X)$ assigns 1 to an element that belongs to $X$ and 0 to an element that does not belong to $X$.*

**Definition 4.** *Given a composite information system $CIS = (U, A, V, f)$. Let $B \subseteq A$ and $CR_B$ be a composite relation on $U$, $M_{n \times n}^{CR_B} = (m_{ij})_{n \times n}$ be an $n \times n$ matrix representing $CR_B$, called the relation matrix w.r.t. $B$. Then*

$$m_{ij} = \begin{cases} 1, & (x_i, x_j) \in CR_B \\ 0, & (x_i, x_j) \notin CR_B \end{cases} \tag{5}$$

**Corollary 1.** *Let $M_{n\times n}^{CR_B}=(m_{ij})_{n\times n}$ and $CR_B$ be a composite relation on $U$. Then $m_{ii}=1$, $1\leq i\leq n$.*

Next, we discuss about boolean methods to derive lower and upper approximation in composite rough sets.

**Lemma 1.** *Given $X\subseteq U$ in a composite information system $CIS=(U,A,V,f)$, where $U=\{x_1,x_2,\ldots,x_n\}$. $B\subseteq A$ and $CR_B$ is a composite relation on $U$. Then the lower and upper approximations of $X$ in the composite information system can be computed as follows.*
*(1) The n-column boolean vector $G(\overline{CR_B}(X))$ of the upper approximation $\overline{CR_B}(X)$:*

$$G(\overline{CR_B}(X))=M_{n\times n}^{CR_B}\otimes G(X) \tag{6}$$

*where $\otimes$ is the Boolean product of matrices.*
*(2) The n-column boolean vector $G(\underline{CR_B}(X))$ of the lower approximation $\underline{CR_B}(X)$:*

$$G(\underline{CR_B}(X))=-(M_{n\times n}^{CR_B}\otimes G(-X)) \tag{7}$$

*where $-X$ denotes the complementary set to $X$.*

*Proof.* Suppose $M_{n\times n}^{CR_B}=(m_{ik})_{n\times n}$, $G(X)=(g_1,g_2,\cdots,g_n)^T$ and $G(\overline{CR_B}(X))=(u_1,u_2,\cdots,u_n)^T$. $\wedge,\vee$ denote minimum and maximum operators, respectively.

(1) "$\Rightarrow$": $\forall i\in\{1,2,\cdots,n\}$, if $u_i=1$, then $x_i\in\overline{CR_B}(X)$, $CR_B(x_i)\cap X\neq\emptyset$, and $\exists x_j\in CR_B(x_i),x_j\in X$, that is to say $(x_i,x_j)\in CR_B$. Thus, $m_{ij}=1$ and $g_j=1$, and $\bigvee\limits_{k=1}^{n}(m_{ik}\wedge g_k)=m_{ij}\wedge g_j=1$. Hence, $\forall i\in\{1,2,\cdots,n\}$, $u_i\leq\bigvee\limits_{k=1}^{n}(m_{ik}\wedge g_k)$.

"$\Leftarrow$": $\forall i\in\{1,2,\cdots,n\}$, if $\bigvee\limits_{k=1}^{n}(m_{ik}\wedge g_k)=1$, then $\exists j\in\{1,2,\cdots,n\}$, $m_{ij}=1$ and $g_j=1$. Thus, $x_j\in CR_B(x_i)$ and $x_j\in X$. Then, $CR_B(x_i)\cap X\neq\emptyset$, namely, $x_i\in\overline{CR_B}(X)$ and $u_i=1$. Therefore, $\forall i\in\{1,2,\cdots,n\}$, $u_i\geq\bigvee\limits_{k=1}^{n}(m_{ik}\wedge g_k)$.

Thus, $\forall i\in\{1,2,\cdots,n\}$, $u_i=\bigvee\limits_{k=1}^{n}(m_{ik}\wedge g_k)$, namely, $G(\overline{CR_B}(X))=M_{n\times n}^{CR_B}\otimes G(X)$.

(2) The proof is similar to that of (1).

**Corollary 2.** *Let $M_{n\times n}^{CR_B}=(m_{ik})_{n\times n}$ and $G(X)=(g_1,g_2,\cdots,g_n)^T$. Suppose $G(\overline{CR_B}(X))=(u_1,u_2,\cdots,u_n)^T$ and $G(\underline{CR_B}(X))=(l_1,l_2,\cdots,l_n)^T$, $\forall i\in\{1,2,\cdots,n\}$, we have*

$$\begin{cases} u_i=\bigvee\limits_{k=1}^{n}(m_{ik}\wedge g_k)\\ l_i=\bigwedge\limits_{k=1}^{n}(m_{ik}\odot g_k)\end{cases} \tag{8}$$

*where $\odot$ is the logical operation NXOR (not exclusive or).*

### 3.2 Boolean Matrix-Based Method in the Composite Decision Table

**Definition 5.** *Given a composite decision table* $CDT = (U, A \cup D, V, f)$, $B \subseteq A$, *let* $U/D = \{D_1, D_2, \cdots, D_r\}$ *be a partition over the decision* $D$. $\forall D_j \in U/D$, $G(D_j) = (d_{j1}, d_{j2}, \cdots, d_{jn})^T$ *is an* $n$*-column boolean vector of* $D_j$. *Let* $G(D) = (G(D_1), G(D_2), \cdots, G(D_r)) = (d_{kj})_{n\times r}$ *and* $G(-D) = (G(-D_1), G(-D_2), \cdots, G(-D_r)) = (-d_{kj})_{n\times r}$ *be* $n \times r$ *boolean matrices, called decision matrix and decision complementary matrix.*

**Lemma 2.** *Given a composite decision table* $CDT = (U, A \cup D, V, f)$, $B \subseteq A$. *Let* $U/D = \{D_1, D_2, \cdots, D_r\}$ *be a partition over the decision* $D$. $\forall j = 1, 2, \cdots, r$, *the upper and lower approximations of the decision* $D$ *in the composite information system can be computed as follows.*

(1) *The* $n \times r$ *boolean matrices* $G(\overline{CR_B}(D))$ *of the upper approximations of the decision* $D$:

$$G(\overline{CR_B}(D)) = M_{n\times n}^{CR_B} \otimes G(D) \tag{9}$$

(2) *The* $n \times r$ *boolean matrices* $G(\underline{CR_B}(D))$ *of the lower approximation of the decision* $D$:

$$G(\underline{CR_B}(D)) = -(M_{n\times n}^{CR_B} \otimes G(-D)) \tag{10}$$

*Proof.* The proof is similar to that of Lemma 1.

**Corollary 3.** *Let* $M_{n\times n}^{CR_B} = (m_{ik})_{n\times n}$ *and* $G(D) = (d_{kj})_{n\times r}$. *Suppose* $G(\overline{CR_B}(D)) = (u_{ij})_{n\times r}$ *and* $G(\underline{CR_B}(D)) = (l_{ij})_{n\times r}$, $\forall i \in \{1, 2, \cdots, n\}$, $j \in \{1, 2, \cdots, r\}$, *we have*

$$\begin{cases} u_{ij} = \bigvee_{k=1}^{n} (m_{ik} \wedge d_{kj}) \\ l_{ij} = \bigwedge_{k=1}^{n} (m_{ik} \odot d_{kj}) \end{cases} \tag{11}$$

## 4 Parallel Method for Computing CRS Approximations

### 4.1 Parallel Matrix-Based Method

According to the above matrix method, we first give the sequential algorithm for computing rough set approximations in the composite decision table, which is outlined in Algorithm 1. Step 2 is to construct the relation matrix and its time complexity is $O(n^2|B|)$; Step 3 is to construct the decision matrix and its time complexity is $O(n \log r + n)$; Step 4 is to compute the upper approximation matrix and its time complexity is $O(n^2 r)$; Step 5 is to compute the lower approximation matrix and its time complexity is $O(n^2 r)$; Hence, the total time complexity is $O(n^2|B| + n \log r + n + n^2 r + n^2 r) = O(n^2(|B| + r))$.

According to the above analysis, the most intensive computation to occur is the construction of the relation matrix and the computation of the upper and

**Algorithm 1.** The sequential algorithm for computing rough set approximations

**Input**: A composite decision table $CDT = (U, A \cup D, V, f)$, and attribute subset $B \subseteq A$.
**Output**: The rough set approximations of the decision $D$.
**begin**
  **Construct** the relation matrix: $M^{CR_B}_{n\times n} = (m_{ij})_{n\times n}$. // According to Definition 4
  **Construct** the decision matrix: $G(D) = (d_{jk})_{n\times r}$. // According to Definition 5
  **Compute** the upper approximation matrix: $G(\overline{CR_B}(D)) = M^{CR_B}_{n\times n} \otimes G(D)$. // According to Corollary 2
  **Compute** the lower approximation matrix: $G(\underline{CR_B}(D)) = -M^{CR_B}_{n\times n} \otimes G(-D)$. // According to Corollary 2
  **Output** the rough set approximations.
**end**

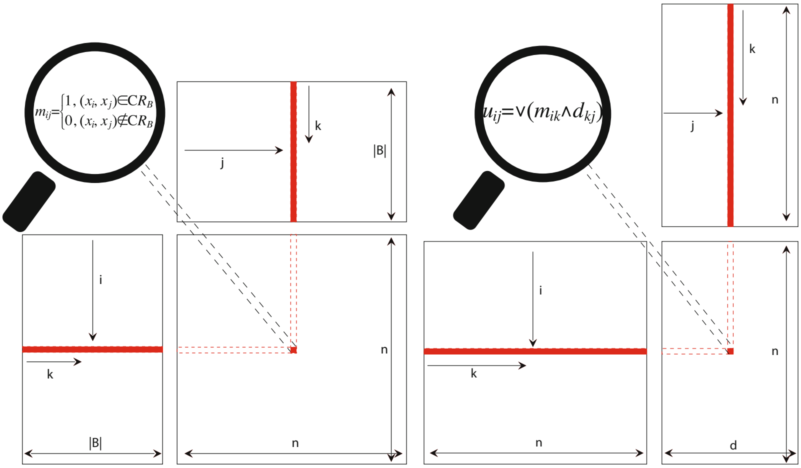
**Algorithm 2.** The parallel algorithm for computing rough set approximations

**Input**: A composite decision table $CDT = (U, A \cup D, V, f)$, and attribute subset $B \subseteq A$.
**Output**: The rough set approximations of the decision $D$.
**begin**
  [In Parallel ] **Construct** the relation matrix: $M^{CR_B}_{n\times n} = (m_{ij})_{n\times n}$.
  [In Sequential] **Construct** the decision matrix: $G(D) = (d_{jk})_{n\times r}$.
  [In Parallel ] **Compute** the upper approximation matrix: $G(\overline{CR_B}(D)) = M^{CR_B}_{n\times n} \otimes G(D)$.
  [In Parallel ] **Compute** the lower approximation matrix: $G(\underline{CR_B}(D)) = -M^{CR_B}_{n\times n} \otimes G(-D)$.
  [In Sequential] **Output** the rough set approximations.
**end**

lower approximation matrices. Obviously, we can accelerate the computational process through the parallelization of these steps, as shown in Algorithm 2.

There are $n$ objects and $|B|$ attributes in raw data. It can be seen as the matrix data with $n$ rows and $|B|$ columns denoted by $\mathcal{V} = (\nu)_{n\times|B|}$. In Figure 1(a), suppose $x_i = (c_{ik})_{1\times|B|}$ and $x_j^T = (c_{kj})_{|B|\times 1}$, then $m_{ij}$ is the result of composite operation of two vectors $x_i$ and $x_j^T$. If $(x_i, x_j) \in CR_B$, $m_{ij} = 1$; otherwise, $m_{ij} = 0$. Hence, each $m_{ij}$ can be computed independently and in parallel.

Similarly, after constructing relation matrix $M^{CR_B}_{n\times n} = (m_{ik})_{n\times n}$ and decision matrix $G(D) = (d_{kj})_{n\times r}$, we suppose the upper approximation matrix $G(\overline{CR_B}(D)) = (u_{ij})_{n\times r}$. According to Corollary 2, we have $G(\overline{CR_B}(D)) = M^{CR_B}_{n\times n} \otimes G(D)$. Hence, suppose $m_i = (m_{ik})_{1\times n}$ and $d_j = (d_{kj})_{n\times 1}$, then $u_{ij} = \bigvee_{k=1}^{n} (m_{ik} \wedge d_{kj})$ according to the Boolean operation, as shown in Figure 1(b). Hence, each $u_{ij}$ can be computed independently and in parallel. Similarly, we can also compute lower approximation.




(a) Relation matrix (b) Upper approximation matrix

**Fig. 1.** Computing relation matrix and upper approximation matrix in parallel

---

**Algorithm 3.** CUDA Computing CRS Approximations

---

**Input**: A composite decision table $CDT = (U, A \cup D, V, f)$, and attribute subset $B \subseteq A$.

**Output**: The rough set approximations of the decision $D$.

**begin**
- (a) **Construct** the relation matrix:
- **for** *each* $m_{ij} \in M_{n\times n}^{CR_B}$ **do**
  - $m_{ij} = \begin{cases} 1, & (x_i, x_j) \in CR_B \\ 0, & (x_i, x_j) \notin CR_B \end{cases}$
- **end**
- (b) **Construct** the decision matrix: $G(D) = (d_{jk})_{n\times r}$.
- (c) **Compute** the upper approximation matrix:
- **for** *each* $u_{ij} \in G(\overline{CR_B}(D))$ **do**
  - $u_{ij} = \bigvee_{k=1}^{n} (m_{ik} \wedge d_{kj})$
- **end**
- (d) **Compute** the lower approximation matrix:
- **for** *each* $l_{ij} \in G(\underline{CR_B}(D))$ **do**
  - $l_{ij} = \bigwedge_{k=1}^{n} (m_{ik} \odot d_{kj})$
- **end**
- **Output** the rough set approximations.

**end**

### 4.2 CUDA Implementation

Algorithm 3 shows the CUDA algorithm of computing CRS approximations. The Stages (a), (c), (d) can be computed in parallel with CUDA. The time complexities of Stages (a), (c), (d) are $O(n^2|B|/p)$, $O(n^2r/p)$, $O(n^2r/p)$, respectively, where $p$ is the total number of threads in kernel function with CUDA. The time complexity of Stage (b) is $O(n \log r + n)$. Hence, the total time complexity of the CUDA algorithm is $O(n^2|B|/p + n^2r/p + n^2r/p + n \log r + n) = O(n^2(|B| + r)/p)$.

## 5 Experimental Analysis

Our test system consists of a Inter(R) Core(TM)i7-2670QM @2.20GHz (4 cores, 8 threads in all) and an NVIDIA GeForce GT 555M. We have implemented a GPU version with CUDA C [2], and a CPU version with C/C++. Then, we give a performance comparison between these two versions.

To test the performance of the parallel algorithm, we download the data set *Connect* from the machine learning data repository, University of California at Irvine [13]. The data set *Connect* consists of tens of thousands of samples. Each sample consists of 42 condition attributes and 1 decision attribute. In our experiment, we extract the data from the data set *Connect* with different numbers of samples randomly, *i.e.*, 8000, 12000, 16000, 20000, 24000, 28000, and 32000. Besides, user-defined data sets are used in our experiments, which are generated randomly with different sizes of samples and features.

Table 2 shows the information of data sets with the computational time and speedup. From the result of the data set *Connect*, it is easy to know that the computational time of CPU and GPU implementations increases with the increase of the size of data, and the GPU implementation achieves 1.6-2.2x over the CPU implementation on this data set. From the result of used-defined data sets, we also find that the computational time of CPU and GPU implementations increases with the increase of the size of data. Moreover, the GPU implementation performs better, achieves 3.1-4.4x speedup over the CPU implementation on used-defined data sets.

## 6 Conclusions

In this paper, we presented the boolean matrix representation of the lower and upper approximations in the composite information system. According to characteristic of matrix operations, we proposed a parallel method based on matrix for computing approximations. By time complexity analysis, the key steps are to construct the relation matrix and to compute the boolean matrices of the lower and upper approximations. Therefore, we used GPUs to parallelize and accelerate these steps. The performance comparison between these GPU implementation and CPU implementation was given, which showed our implementation could accelerate the process of computing rough set approximations. We will optimize the GPU implementation and design the composite rough set based feature selection on GPUs in future.

**Table 2.** Comparison of GPU and CPU

| Data Set | Samples×Features(Classes) | Computational time (s) | | Speedup |
|---|---|---|---|---|
| | | CPU | GPU | |
| Connect | 8000 × 42(3) | 0.641315 | 0.377427 | 1.699176 |
| | 12000 × 42(3) | 1.424260 | 0.860306 | 1.655527 |
| | 16000 × 42(3) | 2.624810 | 1.462150 | 1.795171 |
| | 20000 × 42(3) | 4.029880 | 2.253230 | 1.788490 |
| | 24000 × 42(3) | 6.022810 | 3.301330 | 1.824358 |
| | 28000 × 42(3) | 8.470160 | 4.541230 | 1.865168 |
| | 32000 × 42(3) | 13.08390 | 5.910770 | 2.213569 |
| User-defined | 640 × 64(8) | 0.006069 | 0.001801 | 3.369794 |
| | 1280 × 128(8) | 0.026063 | 0.007054 | 3.694783 |
| | 2560 × 256(16) | 0.159052 | 0.038413 | 4.140577 |
| | 3840 × 384(16) | 0.366586 | 0.085946 | 4.265306 |
| | 5120 × 512(32) | 1.217970 | 0.277423 | 4.390299 |
| | 10240 × 1024(32) | 4.972580 | 1.114920 | 4.460033 |
| | 15360 × 1536(64) | 18.809700 | 5.913800 | 3.180645 |
| | 20480 × 2048(64) | 33.803200 | 10.527700 | 3.210881 |

**Acknowledgements.** This work is supported by the National Science Foundation of China (Nos. 60873108, 61175047, 61100117) and NSAF (No. U1230117), and the Science and Technology Planning Project of Sichuan Province (No. 2012RZ0009), China, and the Fostering Foundation for the Excellent Ph.D. Dissertation of Southwest Jiaotong University 2012.

## References

1. NVIDIA GPUs (2013), `https://developer.nvidia.com/cuda-gpus`
2. NVIDIA CUDA (2013), `http://www.nvidia.com/object/cuda_home_new.html`
3. Grzymała-Busse, J.W.: Characteristic relations for incomplete data: A generalization of the indiscernibility relation. In: Tsumoto, S., Słowiński, R., Komorowski, J., Grzymała-Busse, J.W. (eds.) RSCTC 2004. LNCS (LNAI), vol. 3066, pp. 244–253. Springer, Heidelberg (2004)
4. Grzymała-Busse, J.W.: Characteristic relations for incomplete data: A generalization of the indiscernibility relation. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets IV. LNCS, vol. 3700, pp. 58–68. Springer, Heidelberg (2005)
5. Guan, Y., Wang, H.: Set-valued information systems. Information Sciences 176(17), 2507–2525 (2006)
6. Hu, Q., Xie, Z., Yu, D.: Hybrid attribute reduction based on a novel fuzzy-rough model and information granulation. Pattern Recognition 40, 3509–3521 (2007)
7. Hu, Q., Yu, D., Liu, J., Wu, C.: Neighborhood rough set based heterogeneous feature subset selection. Information Sciences 178(18), 3577–3594 (2008)
8. Kryszkiewicz, M.: Rough set approach to incomplete information systems. Information Sciences 112(1-4), 39–49 (1998)
9. Leung, Y., Fischer, M.M., Wu, W.Z., Mi, J.S.: A rough set approach for the discovery of classification rules in interval-valued information systems. International Journal of Approximate Reasoning 47(2), 233–246 (2008)

10. Li, T., Ruan, D., Geert, W., Song, J., Xu, Y.: A rough sets based characteristic relation approach for dynamic attribute generalization in data mining. Knowledge-Based Systems 20(5), 485–494 (2007)
11. Liu, G.: The axiomatization of the rough set upper approximation operations. Fundamenta Informaticae 69(3), 331–342 (2006)
12. Mi, J.S., Zhang, W.X.: An axiomatic characterization of a fuzzy generalization of rough sets. Information Sciences 160(1–4), 235–249 (2004)
13. Newman, D., Hettich, S., Blake, C., Merz, C.: UCI Repository of Machine Learning Databases. University of California, Department of Information and Computer Science, Irvine, CA (1998), http://archive.ics.uci.edu/ml/
14. Owens, J.D., Luebke, D., Govindaraju, N., Harris, M., Krüger, J., Lefohn, A.E., Purcell, T.: A survey of general-purpose computation on graphics hardware (2007)
15. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data, System Theory, Knowledge Engineering and Problem Solving, vol. 9. Kluwer Academic Publishers, Dordrecht (1991)
16. Pawlak, Z., Skowron, A.: Rough sets: Some extensions. Information Sciences 177(1), 28–40 (2007)
17. Qian, Y., Dang, C., Liang, J., Tang, D.: Set-valued ordered information systems. Information Sciences 179(16), 2809–2832 (2009)
18. Qian, Y., Liang, J., Pedrycz, W., Dang, C.: Positive approximation: An accelerator for attribute reduction in rough set theory. Artificial Intelligence 174(9-10), 597–618 (2010)
19. Stefanowski, J., Tsoukiàs, A.: On the extension of rough sets under incomplete information. In: Zhong, N., Skowron, A., Ohsuga, S. (eds.) RSFDGrC 1999. LNCS (LNAI), vol. 1711, pp. 73–82. Springer, Heidelberg (1999)
20. Yao, Y.: Relational interpretations of neighborhood operators and rough set approximation operators. Information Sciences 111(1-4), 239–259 (1998)
21. Zhang, J., Li, T., Ruan, D., Gao, Z., Zhao, C.: A parallel method for computing rough set approximations. Information Sciences (2012)
22. Zhang, J., Li, T., Ruan, D., Liu, D.: Neighborhood rough sets for dynamic data mining. International Journal of Intelligent Systems 27(4), 317–342 (2012)
23. Zhang, J., Li, T., Chen, H.: Composite rough sets. In: Lei, J., Wang, F.L., Deng, H., Miao, D. (eds.) AICI 2012. LNCS, vol. 7530, pp. 150–159. Springer, Heidelberg (2012)
24. Zhang, J., Li, T., Ruan, D., Liu, D.: Rough sets based matrix approaches with dynamic attribute variation in set-valued information systems. International Journal of Approximate Reasoning 53(4), 620–635 (2012)