A Dynamic Method Based-matrix for Batch Updating Decision Regions of Hybrid Information System

Jiajun Chen*, Shuhao Yu , Ping Jin, Yan Ma
College of Electronics and Information Engineering, West Anhui University, Lu’an , Anhui, 237012,China
*Corresponding. chenjj@wxc.edu.cn

Abstract. Dynamics and complexity are the important characteristics of current information systems, and intelligent decision-making problem of complex information systems in dynamic environment is an important research content in the fields of intelligent decision-making. Decision-theoretic rough set model is an important model based on rough computing, and matrix is an effective mathematical tool, which was widely used in various fields of intelligent computing because of its easy representation and convenient calculation. In this paper, neighbourhood decision-theoretic rough sets(NDTRS) and the dynamic updating methods of decision regions of hybrid decision information systems were studied ,and a dynamic method based-matrix for batch updating decision regions of hybrid decision information systems was proposed by introducing neighbourhood relation matrix and probability matrix to calculate the decision regions.At the same time, In the process of algorithm design, the incremental matrix is defined to updated the probability matrix, so as to improve the algorithm from the perspective of time complexity.Furthermore, the algorithms were described and analyzed in detail.Finally, the experimental results fully proved the effectiveness and feasibility of the algorithm.

1. Introduction
With the development of data science, complexity, dynamics and uncertainty are the important characteristics of data development in various industries. Intelligent decision of complex information system in dynamic environment is an important research content. Decision-theoretic rough set model is a rough computing model, which mainly solves the actual decision problems with uncertain factors. Decision-theoretic rough sets and its extended models have been widely concerned by scholars at home and abroad [2-9], and among which the dynamic knowledge updating method for complex data is a hot topic. Literature [10-12] studied the dynamic approximate updating problems of in different types of complex information systems. Literature [13-16] studied the dynamic knowledge acquisition methods under rough sets models. However, there are few researches on dynamic knowledge updating technology of decision-theoretic rough sets in complex information system. So, In this paper, we studied the hybrid decision information system and neighborhood decision-theoretic rough set model by considering the wide application of matrix [7-8,15-16], and a dynamic method based-matrix for batch updating decision regions of hybrid information system and its improved algorithm were proposed, the experimental results proved the effectiveness and advantages of the improved algorithm.

2. Preliminary knowledge
In this section, we will explain the theory of hybrid decision information system and neighborhood decision-theoretic rough set model.
2.1. Hybrid decision information system and neighborhood relationship

In the classical rough set theory, if a four tuple\( \text{DIS} = (U, AT = A^c \cup A^n, V, F) \) is a decision information system, where \( U \) is the universe of decision information system, \( AT \) represents the finite set of attributes, including \( A^c \) and \( A^n \), where \( A^c \) denotes symbolic attribute set and \( A^n \) denotes numerical attribute set, then the decision information system is called hybrid decision information system.

For a given a hybrid decision information system \( \text{DIS} = (U, AT = A^c \cup A^n, V, F) \), where \( U = \{x_i | i = 1,2,\cdots, |U|\} \), assumed attribute set \( B \subseteq AT \ (B = B^c \cup B^n) \), satisfied \( B^c \subseteq A^c \) and \( B^n \subseteq A^n \), then the neighborhood relation induced by attribute set \( B \) can be expressed as:

\[
N^\delta_B = \{(x_i, x_j) \in U \times U | (\forall b \in B^c, b(x_i) = b(x_j)) \land \Delta_B^\delta(x_i, x_j) \leq \delta, i, j = 1,2,\cdots, |U|\}.
\]

Where \( b(x_i) \), \( b(x_j) \) represents the attribute values of objects \( x_i \) and \( x_j \) under symbolic attribute \( B \) respectively, \( \Delta_B^\delta(x_i, x_j) \) denotes the distance measure of objects \( x_i \) and \( x_j \) under numerical attribute \( B^n \), \( \delta (\delta \geq 0) \) is the neighborhood radius. Where \( \Delta \) is a distance function, here only take Euclidean distance as a metric function and is defined as:

\[
\Delta_B^\delta(x_i, x_j) = \sqrt{\sum_{k=1}^{|B^n|} (b_k(x_i) - b_k(x_j))^2}.
\]

According to the above definition, for \( \forall x_i \in U \), the neighborhood granularity of \( x_i \) with respect to attribute set \( B \) is defined as: \( N^\delta_B(x_i) = \{x_j | (x_i, x_j) \in N^\delta_B, x_j \in U \} \).

2.2. Neighborhood decision-theoretic rough set

For a given a hybrid decision information system \( \text{DIS} = (U, AT = A^c \cup A^n, V, F) \), where attribute set \( B \subseteq AT \ (B = B^c \cup B^n) \), satisfied \( B^c \subseteq A^c \) and \( B^n \subseteq A^n \), and \( B^c \subseteq A^c \) represents symbolic attribute sets, \( B^n \subseteq A^n \) represents numeric attribute sets, for \( \forall x_i \in U \), order \( N^\delta_B(x_i) \) is the neighborhood granularity of \( x_i \) with respect to attribute set \( B \), for \( \forall X \subseteq U \) and the threshold pair \( (\alpha, \beta) \), where \( 1 \geq \alpha \geq \beta \geq 0 \), then the decision rules of neighborhood decision-theoretic rough set about \( X \) can be defined as:

\[
\begin{align*}
\text{POS}_N(X) &\text{ if } P(X | N^\delta_B(x_i)) \geq \alpha, \text{then } x_i \in \text{POS}_N(X) \\
\text{BND}_N(X) &\text{ if } \beta < P(X | N^\delta_B(x_i)) \leq \alpha, \text{then } x_i \in \text{BND}_N(X) \\
\text{NEG}_N(X) &\text{ if } P(X | N^\delta_B(x_i)) \leq \beta, \text{then } x_i \in \text{NEG}_N(X)
\end{align*}
\]

POS\(_N\)(\(X\)), BND\(_N\)(\(X\)) and NEG\(_N\)(\(X\)) represents the positive decision regions, boundary regions and negative decision regions with respect to \( X \) based on NDTRS model can be expressed as follows:

\[
\begin{align*}
\text{POS}_N(X) &\text{ if } \{ x_i | P(X | N^\delta_B(x_i)) \geq \alpha, i = 1,2,\cdots, |U|, k = 1,2,\cdots, m \} \\
\text{BND}_N(X) &\text{ if } \{ x_i | \beta < P(X | N^\delta_B(x_i)) \leq \alpha, i = 1,2,\cdots, |U|, k = 1,2,\cdots, m \} \\
\text{NEG}_N(X) &\text{ if } \{ x_i | P(X | N^\delta_B(x_i)) \leq \beta, i = 1,2,\cdots, |U|, k = 1,2,\cdots, m \}
\end{align*}
\]

3. Decision regions dynamic batch updating method based-matrix

In this section, calculation method of decision regions based on matrix in hybrid decision information system is introduced in detail.

3.1. Matrix representations of decision regions

Definition 2. Neighborhood relation matrix. For a given a hybrid decision information system \( \text{DIS} = (U, AT = A^c \cup A^n, V, F) \), assumed attribute set \( B \subseteq AT \ (B = B^c \cup B^n) \), satisfied \( B^c \subseteq A^c \) and \( B^n \subseteq A^n \), the neighborhood relation matrix \( N^\delta_B \) can be expressed as follows:

\[
N^\delta_B(x_i, x_j) = \{(x_i, x_j) \in U \times U | (\forall b \in B^c, b(x_i) = b(x_j)) \land \Delta_B^\delta(x_i, x_j) \leq \delta, i, j = 1,2,\cdots, |U|\}.
\]

According to the above definition, for \( \forall x_i \in U \), the neighborhood granularity of \( x_i \) with respect to attribute set \( B \) is defined as: \( N^\delta_B(x_i) = \{x_j | (x_i, x_j) \in N^\delta_B, x_j \in U \} \).
A^n, δ (δ ≥ 0) is the neighborhood radius, Δ is a distance function, then neighborhood relation matrix NM = [m_{ij}] with respect to B based on NDTRS model can be expressed as follows:

\[ m_{ij} = \begin{cases} 
1 & \text{if } (\forall b \in B^c, b(x_i) = b(x_j)) \land \Delta_{B^n}(x_j, x_i) \leq \delta; \\
0 & \text{other}.
\end{cases} \]

**Definition 3.** Characteristic function vector. For given a hybrid decision information system DIS = \( (U, AT = A^c \cup A^n, V, F) \), order characteristic function vector based on U is expressed as \( F(U) = [1]_{|U| \times 1} \), then the characteristic function vector of object sets X about the universe U can be defined as: \( F(X) = [f_i]_{|U| \times 1} \), \( f_i = \begin{cases} 
1 & \text{if } x_i \in X; \\
0 & \text{other }
\end{cases} \text{ for } \forall x_i \in U.

**Definition 4.** Probability matrix. For given \( DIS = (U, AT = A^c \cup A^n, V, F) \). Assumed attribute set \( B \subseteq AT \) \( (B = B^c \cup B^n) \), satisfied \( B^c \subseteq A^c \) and \( B^n \subseteq A^n \). Order \( NM = [m_{ij}] \) denotes neighborhood relation matrix with respect to B, \( F(U) = [1, 1, \cdots, 1] \) denotes characteristic function vector based on U, for \( \forall X \subseteq U \), \( F(X) = [f_i]_{|U| \times 1} \) denotes the characteristic function vector of object set X about the universe U, then the intermediate matrix \( PM1(X) = [v^i]_{|U| \times 1} \), \( PM2(X) = [\mu^i]_{|U| \times 1} \) and probability matrix \( PM(X) = [\theta^i]_{|U| \times 1} \) based on U are expressed as respectively:

\[
\begin{align*}
PM1(X) &= NM \times F(U) \\
PM2(X) &= NM \times F(U) \\
PM(X) &= PM1(X) / PM2(X)
\end{align*}
\]

where \( v^i = \sum_{j=1}^{|U|} m_{ij} \times f_j \), \( \mu^i = \sum_{j=1}^{|U|} m_{ij} \), \( \theta^i = v^i / \mu^i \).

According to decision rules \( (P_\alpha), (B_\beta), (N_\beta) \) and **definition 4**, the decision region matrix of hybrid information system can be easily obtained.

**Definition 5.** Given a hybrid decision information system \( DIS = (U, AT = A^c \cup A^n, V, F) \), for \( \forall X \subseteq U \), \( PM(X) = [\theta^i|_{|U| \times 1} \) denotes probability matrix about X and given threshold pair \( (\alpha, \beta) \), then the matrices of the positive regions, boundary regions, and negative regions of the object set \( X \):

\[
\begin{align*}
PM^\text{POS}(X) &= [\omega^\text{pos}(X)]_{|U| \times 1} \\
PM^\text{BD}(X) &= [\omega^\text{bd}(X)]_{|U| \times 1} \\
PM^\text{NEG}(X) &= [\omega^\text{neg}(X)]_{|U| \times 1}
\end{align*}
\]

can be expressed as:

\[
\begin{align*}
\omega^\text{pos} &= \{ 1, \text{if } \theta^i \geq \alpha; \} \\
\omega^\text{bd} &= \{ 1, \text{if } \beta \leq \theta^i < \alpha; \} \\
\omega^\text{neg} &= \{ 1, \text{if } \theta^i \leq \beta; \}
\end{align*}
\]

By **definition 1** and **definition 5**, suppose the partitions induced by decision attribute D is expressed as \( \pi_D = \{ D_1, D_2, \cdots, D_m \} \), then the matrices of the positive regions, boundary regions, and negative regions with respect to \( \pi_D \) in NDTRS model can be expressed as:

\[
\begin{align*}
PM^\text{POS}(\pi_D) &= \max_{1 \leq s \leq m} PM^\text{POS}(D_s) \\
PM^\text{BD}(\pi_D) &= \max_{1 \leq s \leq m} PM^\text{BD}(D_s) \\
PM^\text{NEG}(\pi_D) &= \max_{1 \leq s \leq m} PM^\text{NEG}(D_s)
\end{align*}
\]

**Definition 6.** decision recognition rate. Given \( DIS = (U, AT = A^c \cup A^n, V, F) \) be neighborhood decision information system, where \( \pi_D = \{ D_1, D_2, \cdots, D_m \} \) are the partitions about decision attribute D. Order \( PM^\text{POS}(\pi_D) \), \( PM^\text{NEG}(\pi_D) \) is the matrix of the positive regions and negative regions with respect to \( \pi_D \), then decision recognition rate in NDTRS model can be defined by:

\[
\text{DrecRate} = \frac{\sum_{\pi_D} PM^\text{POS}(\pi_D) + \sum_{\pi_D} PM^\text{NEG}(\pi_D)}{|U|}
\]

Where the definition of DrecRate reflects the probability of correctly identifying positive decision and negative decision for the decision table neighborhood information system.

### 3.2 Dynamic batch updating algorithm and improved algorithm description

we take the dynamic batch adding datasets as an example to describe the basic algorithm for solving the decision regions after adding datasets. See **algorithm 1** for detailed description.

**Algorithm 1:** Decision regions updating algorithm based on batch data addition
Input: Neighborhood decision information system \( DIS = (U1, AT) \), dynamic datasets \( U2 \), neighborhood radius \( \delta \) (\( \delta \geq 0 \)) and threshold pair \(( \alpha, \beta \))

Output: Updated decision region matrix \( DPOS, DBND, DNEG \)

1: Read the dynamic datasets \( U2 \), add the updated datasets \( U1 \) into \( \pi_U = \{D_1, D_2, \ldots, D_m\} \) according to the decision attributes and stored in the list \( DiList \), and the value of the decision category is stored in the list \( class \).

2: Order \( HAKPOS = [], HAKBND = [], HAKNEG = [] \) // store the positive decision region matrix list, boundary region matrix list and negative decision region matrix list respectively.

3: For \( \forall D_i \in \pi_U \)
   According to definition 2 and algorithm 3, establish the neighborhood relation matrix \( NM \) of \( U1 \).
   Calculate the intermediate matrix \( PM1(D_i), PM2(D_i) \) and probability matrix \( PM(D_i) \) according to definition 4 and the neighborhood relation matrix \( NM \).
   Calculate decision region matrix and stored in the lists \( hakipos, hakibnd \) and \( hakineg \) respectively according to definition 5, probability matrix \( PM(D_i) \) and threshold pair \(( \alpha, \beta \)) .
   \( HAKPOS.append(hakipos); HAKBND.append(hakibnd); HAKNEG.append(hakineg) \).

4: According to definition 5, calculate the decision region matrix and store it in \( DPOS, DBND, DNEG \) by matrix \( HAKPOS, HAKBND \) and \( HAKNEG \).

5: Output \( DPOS, DBND, DNEG \).

According to the execution steps of algorithm 1, assume that the original datasets is \( U1 \), the added datasets is \( U2 \), the decision class is \( m \), and the number of attributes is \( c \). Then the time complexity of steps 1 and 3 is \( m|U1 + U2| \) respectively, the time complexity of calculating neighborhood matrix \( NM \) in step 2 is \( c|U1 + U2|^2 \). For each decision class, neighborhood matrix \( NM \) needs to be solved, so the maximum time complexity in step 2 is \( mc|U1 + U2|^2 \). If the datasets is dynamically increased \( n \) times, the whole time complexity is \( mnc|U1 + U2|^2 \). The original datasets need to participate in the calculation every time when the batch data subset is added dynamically in algorithm1. In order to reduce the time complexity, we can first calculate the decision regions of the original datasets, and then dynamically modify the decision regions every time when the data subsets are updated. The specific algorithm is shown in algorithm 2.

**Algorithm 2**: Improved Decision regions updating algorithm based on batch data addition

**Input**: Neighborhood information system \( DIS = (U1, AT) \), dynamic datasets \( U2 \), neighborhood radius \( \delta \) (\( \delta \geq 0 \)) and threshold pair \(( \alpha, \beta \)).

**Output**: Updated decision region matrix \( DPOS, DBND, DNEG \)

1: Read the datasets \( U1 \) and divide \( U1 \) into \( \pi_U = \{D_1, D_2, \ldots, D_m\} \) according to the decision attributes and stored in the list \( DiList \), and the value of the decision category is stored in the list \( class \).

2: According to definition 2, establish the neighborhood relation matrix \( NM \) of \( U1 \).

3: Order \( DPM1=[], DPM2=[], DPM=[] \) // store the positive decision region matrix list, boundary region matrix list and negative decision region matrix list respectively.

4: For \( \forall D_i \in \pi_U \)
   Calculate the intermediate matrix \( PM1(D_i), PM2(D_i) \) and probability matrix \( PM(D_i) \) according to definition 4 and the neighborhood relation matrix \( NM \), and stored in the lists \( hakx1, hakx2 \) and \( hakx \) respectively.
   \( DPM1.append(hakx1), DPM2.append(hakx2), DPM.append(hakx) \)

5: By algorithm 3, read the dynamic datasets \( U2 \), updated \( \pi_U' = \{D_1', D_2', \ldots, D_k\} \), \( DiList \) and class.

6: Calculate the incremental matrix \( RMAK, RRMAK \) by algorithm 4.

7: For any \( D_i \in \pi_U' \)
   Update the intermediate matrix \( PM1, PM2 \) and probability matrix \( PM \) for each \( diList \) in \( DiList \) according to the incremental matrix \( RMAK, RRMAK \).

8: Order \( HAKPOS = [], HAKBND = [], HAKNEG = [] \) // store the positive decision region matrix list, boundary region matrix list and negative decision region matrix list respectively.
9: For any $D_i \in \pi_D^1$

   Calculate decision region matrix and stored in the lists hakipos, hakibnd and hakineg respectively according to definition 5, probability matrix $PM(D_i)$ and threshold pair $(\alpha, \beta)$. 

   HAKPOS.append(hakipos); HAKBND.append(hakibnd); HAKNEG.append(hakineg)

10: According to definition 5, calculate the decision region matrix and store them in DPOS, DBND, DNEG according to matrix HAKPOS, HAKBND and HAKNEG.

11: Output DPOS, DBND, DNEG.

**Algorithm 3.** The algorithm for dynamically modifying dataset informations

**Input:** Original dataset index list Uindex, partition list DiList, decision class value class, added datasets U2.

**Output:** Updated dataset index list Uindex1, partition list DiList1, decision class value list Class1.

1: Read the added datasets U2, get the value of decision attribute form U2 and store them in the list DV.

2: Order Uindex1 = [0, 1, ... |Uindex| + |Ddatai|] //The index list of the updated datasets.

3: Order class1 = class

   For any $vi \in DV$

   If $vi \in$ class

   For every $x_i$ satisfying the condition, find out the partition number k corresponding to $vi$ in the list DiList1, DiList[k].append(xi).

   Else   Save every $x_i$ that meets the condition in the list xlist.

   class1.append(vi)

   DiList.append(xlist)

4: Output Uindex1, DiList1, class1.

**Algorithm 4:** Solving incremental matrix algorithm.

**Input:** datasets U1, attribute set $B = B^c \cup B^n$, dynamic datasets U2 and neighborhood radius $\delta$.

**Output:** Incremental matrix RMAK, RRMAK.

1: Order RMAK=[] // Store the neighborhood relation matrix

2: RRMAK=[] //Store the neighborhood relation matrix generated by the incremental datasets itself

3: For any $x_i \in U1$

   mlow=[]

   for any $x_j \in U2$

   if $\forall b \in B^c, b(x_i) = b(x_j)$ and $\Delta_B^n(x_i, x_j) \leq \delta$ then mlow.append(1)

   else mlow.append(0)

   RMAK.append(mlow)

4: Calculate the neighborhood relation matrix of incremental datasets U2 and store it in RRMAK.

5: Output RMAK, RRMAK.

| Datasets | Class. | Samp. | Feat. |
|----------|--------|-------|-------|
| CarEval. | 4      | 1728  | 6     |
| Wine     | 3      | 178   | 13    |
| Iris     | 3      | 150   | 4     |
| Wdbc     | 2      | 569   | 30    |

**4. experimental analysis**

In order to verified the effectiveness of the proposed algorithm, the experiments were carried out on the standard datasets provided by UCI, the detailed information was shown in the Table 1. During the experiments, 70% of the datasets was selected as the original datasets, and the remaining 30% of the datasets was selected as the incremental datasets, and the incremental datasets was divided into 10
parts on average, one of which was added to the original datasets in batches in each experiment, then the experiment was carried out through the algorithm 1 and algorithm 2 proposed in this paper. The experiment was carried out in a computer with 16g memory in win10 system by using Python 3.8 compiler. A total of 10 experiments were carried out, and the decision recognition rate and algorithm running time were compared and analyzed. In the process of dynamic updating from 10 times experiments, algorithm 1 and algorithm 2 can correctly divided each object into the corresponding decision class, and the time required for 10 times experiments on different datasets is very different, as shown in Figure 1-4. The execution time of the algorithm 2 is much shorter than that of the algorithm 1. When the number of datasets is small, the difference between the two algorithms is not very obvious, but when the data volume is larger, the execution time of the two algorithms is very different, and the advantage of the algorithm 2 is significant. Therefore, the improved algorithm in this paper is more suitable for solving the decision updating problem of a large number of hybrid decision information systems.

5. Conclusion
NDTRS model can analyze decision-making problems of complex decision information systems with numerical or hybrid-valued, matrix has been widely used in various fields of rough set computing because of its easy representation and convenient calculation. In this paper, the matrix technology was introduced to study the dynamic updating method of NDTRS model, a matrix-based decision regions updating algorithm and improved algorithm of NDTRS was proposed, this algorithm can effectively solve the intelligent decision-making problem of dynamic hybrid information system in the current big data environment. Dynamic decision-making in distributed environment will be the focus of further research.

Acknowledgments
The authors gratefully acknowledge the support of the Universities Natural Science Key Project of Anhui Province (No.KJ2020A0637).

References
[1] MIAO Duoqian, LI Daoguo. Rough set theory, algorithm and application[M]. Beijing: Science Press,2007.
[2] Yiyu Yao. Decision-theoretic rough set models [J]. Rough Sets and Knowledge Technology, 2007, 4481:1-12.

[3] Yuhua Qian, Hu Zhang, Yanli Sang, Jiye Liang. Multigranulation decision-theoretic rough sets[J]. International Journal of Approximate Reasoning, 55 (2014): 225-237.

[4] Caihui Liu, Witold Pedrycz and Meizhi Wang. Covering-based multigranulation decision-theoretic rough sets[J]. Journal of Intelligent & Fuzzy Systems, 2017, 32(1): 749-765.

[5] Jiajun Chen, Yuanhuan Wang, Wenjie Wei et al. Granularity Reduction Method Based on Positive Decision Holding for Multi-granulation Decision-Theoretic Rough Set[J], The Journal of Engineering, 2018, 2018(10): 1389-1395.

[6] Li Weiwei, Huang Zhiqiu, Jia Xiuyi, et al. Neighborhood based decision-theoretic rough set models[J]. International Journal of Approximate Reasoning, 2016, 69(C): 1-17.

[7] J.B. Zhang, T.R. Li, D. Ruan, D. Liu. Rough sets based matrix approaches with dynamic attribute variation in set-valued information systems[J], Approximate Reasoning, 2012, 53 (4): 620-635.

[8] X. Hu, S.X. Liu, G.X. Liu, Matrix-based approaches for dynamic updating approximations in multigranulation rough sets[J]. Knowledge-Based Systems, 2017(122): 51-63.

[9] Chen, H., Li, T., Luo, C., Horng, S.-J., & Wang, G. (2015). A Decision-Theoretic Rough Set Approach for Dynamic Data Mining. IEEE Transactions on Fuzzy Systems, 23(6), 1958-1970.

[10] Hu, C., Liu, S., & Liu, G. Matrix-based approaches for dynamic updating approximations in multigranulation rough sets[J]. Knowledge-Based Systems, 2017(122): 51-63.

[11] Chengxiang Hu, Li Zhang. A dynamic framework for updating neighborhood multigranulation approximations with the variation of objects[J]. Information Sciences, 2020(519): 382-406.

[12] Wang, S., Li, T., Luo, C., Hu, J., Fujita, H., & Huang, T. A novel approach for efficient updating approximations in dynamic ordered information systems[J]. Information Sciences, 2020(507), 197-219.

[13] Hu, C., & Zhang, L. Incremental updating probabilistic neighborhood three-way regions with time-evolving attributes[J]. International Journal of Approximate Reasoning, 2020(120), 1-23.

[14] YANG Zhen, QIU Baozhi. Dynamic variable precision rough set model of mixed information system[J], Control and Decision, 2020, 35(2): 297-308.

[15] Chuan Luo, Tianrui Li, Zhang Yi, Hamido Fujita. Matrix approach to decision-theoretic rough sets for evolving data[J], Knowledge-Based Systems, 2016(99): 123-134.

[16] Caihui Liu, Kecan Cai, Duqiong Miao, JinQian. Novel matrix-based approaches to computing minimal and maximal descriptions in covering-based rough sets[J]. Information Sciences, 2020(539): 312-326