Instance selection and attribute reduction based on the nearest tolerance relation

Mengsong Wang[a,*], Xingchen Wu[b]

College of Computer Science and Technology, Henan Polytechnic University, Jiaozuo City, China
[a]212209010012@home.hpu.edu.cn, [b]212309020054@home.hpu.edu.cn
[*]Corresponding author

Abstract: *To address the problem of data classification uncertainty caused by redundant information in information systems (IIS), a tolerance relation is used to expand rough sets, and an instance selection algorithm (ISM) and attribute reduction algorithm (ARM) based on the nearest tolerance relation are proposed. Firstly, the process of computing the approximate set is vectorized using a matrix approach. Based on the results of the lower approximations, an instance selection algorithm (ISM) is designed to determine the instance set. An attribute reduction algorithm (ARM) is designed using attribute dependency as heuristic information. Starting from a core set of attributes in a bottom-up manner, non-core attributes are added to the core set based on the importance of external attributes, resulting in a core reduction set. The experimental results on nine UCI datasets show that the matrix based on ISM algorithm and ARM algorithm effectively remove redundant samples and attributes, and improve various performance indicators. Compared with the original dataset, the ISM algorithm achieved an average instance reduction ratio of 44.2%, and the ARM algorithm achieved an average attribute reduction ratio of 33%. Compared with other attribute reduction algorithms, the ARM algorithm has an overall improvement of 1.15% in classification accuracy on KNN and SVM classifiers.*

Keywords: *tolerance relation; instance selection; attribute reduction; attribute dependency; matrix*

1. Introduction

Redundant information is inevitably generated during the data collection process. Therefore, it is necessary to delete unnecessary instances and attributes in the data, which can help to improve classification accuracy. Rough set theory[1] (RST) is a powerful math tool in data preprocessing, it is highly effective in handling imprecise and inconsistent data[1] , and has been widely applied in fields such as data mining and image processing[2-3].

The classic rough set model is based on equivalent relation and can only handle complete information systems. But, data in daily life is imprecise and incomplete. To solve this problem, scholars have conducted many effective studies. Kryszkiewicz et al[4] first proposed the tolerance relation, which provides a theoretical basis for studying the relation between objects in IIS. On this basis, the tolerance relation model was expanded[5]. Xu et al[6] proposed a data-driven value tolerance relation rough set model to obtain better classification performance. Deris et al[7] used the similar classes and tolerance relation between two objects to delete some incomplete data and achieve reduction effect. But the way data is deleted or modified will change the original semantics of the information system. Wu et al[8]designed neighborhood equivalent tolerance classes to solve the problem of relaxed conditions during sample partitioning. Rohmat et al[9] proposed a relative tolerance relation for rough sets to handle IIS, improving the flexibility and accuracy of data classification. It can be seen that tolerance relation are highly effective as a direct way to handle IIS.

Instance selection and attribute reduction are important steps in data preprocessing, aimed at reducing unimportant instances and attributes in the dataset, thereby reducing the horizontal and vertical dimensions of data and improving the speed of data preprocessing. In addition, due to the widespread existence of default values in information systems, they can have an impact on classification results. Therefore, using tolerance relation theory to improve calculations in IIS[10] and developing data reduction algorithms is worth studying. Zhang et al[11] proposed an incremental attribute reduction algorithm to complete the attribute reduction task of dynamic data, while L et al[12] used dependency specific attribute reduction algorithm and inconsistency based object specific reduction algorithm. Xu et al[13] designed an attribute reduction algorithm based on relative decision entropy in fuzzy neighborhood, which improved the classification performance of data. Xia et al[14] introduced the concept of knowledge granularity and proposed an attribute reduction algorithm using attribute importance as heuristic information. Zhao et al[15] introduced importance measurement and proposed an extension called sub relation tolerance class. They developed an incremental feature selection algorithm for handling incomplete data streams, which improved classification accuracy and shortened computation time. It can be seen that most studies aim to calculate reduction through significance measures, which mainly include dependency[16-18], information entropy[19], etc., and are widely popular because of simplicity and effectiveness.

2. Nearest tolerance relation set representation

Definition 1: An incomplete information system $IIS=(U,\mathrm{A},V,f)$ $U=\{x_1,x_2,\dots,x_n\},\forall x\in U$. If there is no missing value in condition attribute value, it is a reliable element, represented as x_r. If condition attribute value contains missing values, it is a controversial element, represented as x_c.

Definition 2: (Nearest Tolerance Relation) An incomplete information system $IIS=(U,A,V,f)$, $x,y\in U$, $B\subseteq A$, the nearest tolerance relation T_B is defined as:

$$T_B=\{x\mapsto y|a(x)=a(y)\vee(a(x)\neq *\wedge a(y)=*)\} \tag{1}$$

Definition 3: (Nearest Tolerance Class) An incomplete Information System $IIS=(U,A,V,f)$ $B\subseteq C$,$x_r\in U.NT_B=\{x_r\mapsto x_c^1\mapsto x_c^2\mapsto\cdots\mapsto x_c^k\}$. A sequence that represents the starting point and satisfies definition 2 between adjacent objects is called the nearest tolerance class NT_B.

Definition 4 (approximation space) An incomplete information system $IIS=(U,A,V,f)$,$B\subseteq C,U/D=\{d_1,d_2,\dots,d_m\}$, where the upper and lower approximations of are defined as:

$$\overline{\psi}_B(D)=\cup_{i=1}^m\{NT_B|NT_B\in U,NT_B\cap d_i\neq\emptyset\} \tag{2}$$

$$\underline{\psi}_B(D)=\cup_{i=1}^m\{NT_B|NT_B\in U,NT_B\subseteq d_i\} \tag{3}$$

2.1 Boolean operations on matrices

In matrix operations, it is necessary to convert the intersection, inclusion, parallelism, and deduplication operations of set operations into matrix form to achieve the equivalent transformation from set to matrix.

Definition 5: Let $\odot$ be a Boolean operation on two matrices, $\boldsymbol{R}_{k\times n}=(r_{ij})_{k\times n}$ and $\boldsymbol{D}_{n\times m}=(d_{ij})_{n\times m}$ is two matrix. $\boldsymbol{C}_{k\times m}=\boldsymbol{R}\odot\boldsymbol{D}=(c_{ij})_{k\times m}$ is defined as

$$c_{ij}=\vee_{k=1}^n\left(r_{ik}\wedge d_{kj}\right) \tag{4}$$

Definition 6: Let $\otimes$ be a Boolean operation on two matrices, $\boldsymbol{R}_{k\times n}=(r_{ik})_{k\times n},\boldsymbol{D}_{n\times m}=(d_{kj})_{n\times m},\boldsymbol{C}_{k\times m}=\boldsymbol{R}\otimes\boldsymbol{D}=(c_{ij})_{k\times m}$ is defined as

$$c_{ij}=\wedge_{k=1}^n\left[(1-r_{ik})\vee d_{kj}\right] \tag{5}$$

Definition 7: Let $\circledast$ be a Boolean operation on two matrices, $\boldsymbol{A}=(a_{ij})_{k\times n}$,$\boldsymbol{B}=(b_{ij})_{k\times m}=[\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2,\dots,\boldsymbol{\alpha}_n]^T$,where $\boldsymbol{\alpha}_i=[r_{i1},r_{i2},\cdots,r_{im}]$, $\boldsymbol{C}_{k\times n}=\boldsymbol{A}\circledast\boldsymbol{B}=(c_{ij})_{k\times n}$ is defined as

$$c_{ij}=a_{ij}\times(\vee\boldsymbol{\alpha}_i) \tag{6}$$

Definition 8: Let $\oplus$be the unary operator of a matrix, $\boldsymbol{A}_{k\times n}=(a_{ij})_{k\times n}=[\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2,\dots,\boldsymbol{\alpha}_n]$, $\boldsymbol{\alpha}_i=[b_{1i},b_{2i},\dots,b_{ki}]^T$, $\boldsymbol{A}_{1\times n}=\oplus\boldsymbol{A}_{k\times n}=[a_1,a_i,\dots,a_n]$ is defined as

$$a_j=\vee\alpha_i=b_{1i}\vee b_{2i}\vee\dots\vee b_{ki} \tag{7}$$

2.2 Approximations matrix representation based on nearest tolerance relation

Definition 9: Let $U=\{x_1,x_2,\cdots,x_n\}$, $X\subseteq U$ .The Boolean representation of $\boldsymbol{G}(X)=[g_1,g_2,\dots,g_n]$ is defined as:

$$g_i=\begin{cases}1,x_i\in X\\0,x_i\notin X\end{cases} \tag{8}$$

Definition 10: An incomplete information system $IIS = (U, A, V, f)$. $U = \{x_1, x_2, \dots, x_n\}$, $U/D = \{d_1, d_2, \dots, d_m\}$is partitioned by D on U. $\forall d_j \in U/D$, $\boldsymbol{d_j} = \left(d_{1j}\ d_{2j} \cdots d_{nj}\right)^T$ is a n-row Boolean vector of $\boldsymbol{d_j}$. The decision matrix $\boldsymbol{M_D}$ is defined as:

$$\boldsymbol{D} = (d_{ij})_{n\times m} = (\boldsymbol{d}_1, \boldsymbol{d}_2, \cdots, \boldsymbol{d}_m) \tag{9}$$

Definition 11: An incomplete information system $IIS = (U, A, V, f)$. $U = \{x_1, x_2, \dots, x_n\}$, $B \subseteq A$. The nearest tolerance relation matrix $\boldsymbol{R}_L^B = \left(r_{ij}\right)_{k\times n}$ is defined as

$$\boldsymbol{R}_L^B = \left(\boldsymbol{G}(NT_1), \boldsymbol{G}(NT_2), \cdots, \boldsymbol{G}(NT_k)\right)^T \tag{10}$$

Where $\boldsymbol{G}(NT_1)$ is the n- row Boolean vector of the nearest tolerance class sequence.

Definition 12: An incomplete information system $IIS = (U, A, V, f)$, $U/D = \{d_1, d_2, \dots, d_m\}$,$B \subseteq A$,$\boldsymbol{R}_B = (r_{ij})_{k\times n}$ is a nearest tolerance relation matrix, $\boldsymbol{\lambda}_D = (a_i)_{n\times 1}$ is a Boolean vector about d_i, the eigenvectors corresponding to the upper and lower approximations of are represented as

$$\boldsymbol{\lambda}_{\overline{T}(D)} = \oplus \left(\boldsymbol{R}_B \circledast (\boldsymbol{R}_B \odot \boldsymbol{D})\right) \tag{11}$$

$$\boldsymbol{\lambda}_{\underline{T}(D)} = \oplus \left(\boldsymbol{R}_B \circledast (\boldsymbol{R}_B \otimes \boldsymbol{D})\right) \tag{12}$$

At this point, the vectorization work has been completed

3. Instance selection and attribute reduction based on nearest tolerance relation

Definition 13: An incomplete information system $IIS = (U, A, V, f)$. $U/D = \{d_1, d_2, \dots, d_m\}$,$A = \{a_1, a_2, \dots, a_k\}$,$B \subseteq A$, the attribute dependency of D relative to B is defined as:

$$\gamma_B(D) = \frac{\left|\lambda_{\underline{T}(D)}^B\right|}{|U|} \tag{13}$$

Definition 14: An incomplete information system $IIS = (U, A, V, f)$, $\forall B \subseteq A$.

(1) If $a \in B$, it is called internal attribute importance, defined as

$$Sig_B^{inner}(a, D) = \gamma_B(D) - \gamma_{B-a}(D) \tag{14}$$

(2) If $a \in A - B$, it is called external attribute importance, defined as

$$Sig_B^{outer}(a, D) = \gamma_{B\cup a}(D) - \gamma_B(D) \tag{15}$$

Definition 15: An incomplete information system $IIS = (U, A, V, f)$, $Sig_B^{inner}(a, D)$ is an internal attribute importance function, ϵ represents a threshold, the attribute kernel $Core$ is defined as

$$Core = Sig_B^{outer}(a, D) > \epsilon \tag{16}$$

Definition 16: An incomplete information system $IIS = (U, A, V, f)$, $\forall B \subseteq A$, if the attribute set B satisfies the following condition, it is called B relative reduction of A

$$\gamma_B(D) = \gamma_A(D) \tag{17}$$

$$\gamma_{B-a}(D) < \gamma_B(D), \forall a \in B \tag{18}$$

The following presents an instance selection algorithm (ISM) and attribute reduction algorithm (ARM) based on the above definition. The ARM algorithm is based on the results of the lower approximations, and the attribute reduction algorithm uses attribute dependency as heuristic information, starting from an attribute core and adopting a bottom-up approach. In each iteration, the attribute with the highest importance is added, and the attribute with zero importance is deleted.

```
Algorithm1: Matrix based instance selection algorithm (ISM)
Input: $IIS = (U, A, V, f)$
Output: Instances $data$, dependency $\gamma_A(D)$
1 Calculate the nearest tolerance relation matrix $\boldsymbol{R}_A = (r_{ij})_{k \times n}$.
2 Calculate the decision matrix $\boldsymbol{D} = (d_{ij})_{n \times m}$.
3 Calculate the lower approximate vector $\boldsymbol{\lambda}_{\underline{T}(D)} = \oplus (\boldsymbol{R}_B \circledast (\boldsymbol{R}_B \otimes \boldsymbol{D}))$
4 Return the approximate set instance object $data \leftarrow \lambda_{\underline{T}(D)}$
5 return $\gamma_A(D) = \frac{|\lambda^B_{\underline{T}(\boldsymbol{D})}|}{|U|}, data$
```

Line 1, calculate the nearest tolerance relation matrix, the time complexity is $O(nk)$. Line 2, calculate the decision matrix, the time complexity is $O(n)$. Line 3, calculate the lower approximate Boolean vector of each d with respect to U. According to definition 6 and definition 7, the time complexity of $\otimes$ is $O(mnk)$, the time complexity of $\circledast$ is $O(m(n+k))$ Thus, the total time complexity of the algorithm is $O(nk + m(n+k)) = O(n(m+k))$.

```
Algorithm2: Matrix based attribute reduction algorithm (ARM)
Input: $IIS = (U, A, V, f)$, threshold $\epsilon$
Output: reduction set $Core$
1 Calculate all attribute dependency $\gamma_A(D)$
2 For each $a \in A$ do
3     $Sig_A^{inner}(a, D) = \gamma_A(D) - \gamma_{A-a}(D)$ // Calculate the degree of internal attributes
4 End for
5 $Core = Sig_A^{inner}(a, D) >= \epsilon$ // Core reduction set
6 $N = A - Core$ // Non-core reduction set
7 While $N \neq \emptyset$ do
8     $list_{outer} = \emptyset$
9     For each $a \in N$ do
10        If $Sig_N^{outer}(a, D) <= 0$ Then
11            $N - a$ // Delete attributes with external attribute degree less than zero
12        Else
13            $list_{outer} \cup a$ // Retain attributes with external attribute degree greater than zero
14    End for
15    If $N \neq \emptyset$ do
16        $Core \cup \boldsymbol{Max}(list_{outer})$ //Add attribute with the highest external attribute degree value
17        $N - a$
18    Else
19        Break
20 End while
21 Return $Core$
```

Lines 1-4, calculate attribute dependency, the time complexity is $O(|A| \times mnk)$. Lines 9-14, calculate the external attribute dependency, the time complexity is $O(|N| \times mnk)$. Lines 15-19, determine whether the non-reduction set is empty and add the attribute with the highest external attribute importance to the reduction set, the time complexity is $O(|N|)$. Thus, the total time complexity of ARM is $O(|A| \times mnk + |N| \times mnk + |N|) = O(mnk(|A| + |N|))$.

4. Experiment

4.1 Datasets and experimental environment

To further demonstrate the effectiveness of the algorithm, nine datasets were selected from the UCI standard machine learning library for experimental analysis. Table 1 provides a basic description of all the datasets used in the experiment. The susy and hepmass datasets were randomly selected from the source data with 10000 pieces of data. The experimental code was written using Python 3.9.6, and the experimental environment was Windows 7 operating system. The Intel (R) Core (TM) i3-9100F CPU had a 3.60GHz and 32GB of memory. The index values obtained in the experiment are the average of ten folds cross validation.

Table 1. The description of datasets

Source	Dataset	Abbreviation	Samples	Attributes	Classes
UCI	hepatitis	HE	155	19	2
UCI	wine	WI	178	13	3
UCI	glass	GL	214	10	7
UCI	ionosphere	LO	351	33	2
UCI	biodeg	BI	1055	41	2
UCI	mushroom	MU	8124	22	2
UCI	hepmass	HP	10000	27	2
UCI	susy	SU	10000	18	2
UCI	magic	MA	19020	10	2

4.2. Instance selection based on the nearest tolerance relation

This section selects instances for the dataset based on the lower approximation results of the ISM algorithm. Firstly, the classification accuracy of the dataset processed by the ISM algorithm was compared with the original dataset without any method processing. The results are shown in Table 2, and the classification accuracy of the dataset on the KNN classifier and SVM classifier, as well as the reduction ratio of instance selection, were provided. ODP represents the raw dataset that has not undergone any algorithmic processing. A higher ratio means removing more instances.

Table 2 The classification accuracy and reduction ratio of ISM algorithm compared to the original datasets

Dataset	ODP		Samples	ISM		Samples	Ratio
	KNN	SVM		KNN	SVM		
HE	0.7283	0.7925	155	**0.8138**	**0.7929**	146	5.81%
WI	**0.9052**	0.916	178	0.8875	**0.9313**	160	10.11%
GL	0.6255	0.6344	214	**0.7956**	**0.7022**	97	54.67%
LO	**0.866**	**0.8748**	351	0.7886	0.721	146	58.4%
BI	0.8102	0.8321	1055	**0.8347**	**0.8689**	817	22.56%
MU	0.9655	0.9186	8124	**0.9793**	**0.9854**	2468	69.62%
HP	0.8686	0.8731	10000	**0.882**	**0.8842**	7279	27.21%
SU	0.6944	0.7325	10000	**0.8444**	**0.8439**	4383	56.17%
MA	0.7018	0.7359	19020	**0.9362**	**0.9386**	1255	93.4%
AVG	0.7962	0.8122	5455	**0.8624**	**0.852**	**1863**	44.2%

To clearly represent the classification data, the optimal classification accuracy is indicated by bold symbols. The experimental results show that ISM algorithm can effectively perform instance selection under the nearest tolerance relation. In terms of instance selection, the instance reduction ranges from 5.81% to 93.4% on the nine datasets, with an average reduction rate of 44.2%. KNN classifier and SVM classifier have higher classification accuracy for ISM processed data. The ISM algorithm has improved the average classification accuracy by 8.3% (KNN) and 4.9% (SVM) compared to the original data. On the KNN classification algorithm, seven out of nine datasets have higher classification accuracy than the original dataset. On the SVM classification algorithm, eight datasets have higher classification accuracy than the original dataset. The experimental results demonstrate the effectiveness of the ISM algorithm under the nearest tolerance relation.

In addition, the performance evaluation indicators include not only classification accuracy, but also precision Pre (Precision), recall (Recall), and F1 (F1 score) indicators. The specific experimental results are shown in Table 3. Compared with the original dataset, the ISM algorithm has significantly improved in various performance indicators. On the SVM classifier, among the nine datasets, only the IO dataset had metrics lower than ODP, while the metrics of the other datasets were all higher than ODP. On the KNN classifier, among the nine datasets, only the WI and MU datasets had indicators lower than ODP, the F1 indicator of the HE dataset was lower than ODP, and the indicators of the other six datasets were all higher than ODP. On the SVM classifier, only the IO dataset had indicators lower than ODP, while the indicators of the other 8 experimental datasets were all higher than ODP. This indicates that the ISM algorithm can improve classification performance with less high-quality resolution.

Table 3 Performance metrics of ODP and ISM algorithm processed datasets on KNN and SVM classifiers

Dataset	Method	KNN			SVM		
		Pre	Rec	F1	Pre	Rec	F1
HE	ODP	0.7383	0.7218	0.7082	0.6395	0.6462	0.6234
	ISM	**0.7509**	**0.722**	0.6953	**0.6641**	**0.6958**	**0.6559**
WI	ODP	0.9201	0.9104	0.9052	0.929	0.9211	0.9169
	ISM	0.905	0.8883	0.8837	**0.9486**	**0.9311**	**0.9305**
GL	ODP	0.5364	0.5494	0.5134	0.5476	0.5847	0.5384
	ISM	**0.7154**	**0.7694**	**0.7099**	**0.5847**	**0.6055**	**0.5723**
IO	ODP	0.7356	0.7121	0.6996	0.7288	0.6776	0.6866
	ISM	**0.7466**	**0.7352**	**0.7222**	0.6445	0.6416	0.6292
BI	ODP	0.7943	0.7975	0.7908	0.8191	0.8049	0.8082
	ISM	**0.821**	**0.8263**	**0.8192**	**0.8618**	**0.8467**	**0.8505**
MU	ODP	0.9751	0.965	0.9635	0.9425	0.9183	0.9121
	ISM	0.953	0.9504	0.9504	**0.9926**	**0.955**	**0.955**
HP	ODP	0.8691	0.8688	0.8687	0.8733	0.873	0.8731
	ISM	**0.8832**	**0.8825**	**0.882**	**0.8843**	**0.8842**	**0.8841**
SU	ODP	0.6947	0.6885	0.689	0.7329	0.7282	0.7289
	ISM	**0.8468**	**0.841**	**0.8423**	**0.8454**	**0.8456**	**0.8441**
MA	ODP	0.7895	0.5808	0.5503	0.7136	0.7227	0.7169
	ISM	**0.9205**	**0.944**	**0.9297**	**0.9239**	**0.9456**	**0.9323**

4.3 Attribute Reduction Based on Nearest Tolerance Relation

4.3.1 The influence of a value on the reduction result

ϵ is a hyperparameter, and different ϵ values will determine different kernel elements, resulting in different reduction results. When selecting the ϵ value, it is important to consider both the dimensionality of the simplified dataset and the classification accuracy of the simplified dataset. In Figure 1, the impact of different ϵ values on the classification accuracy of KNN classifier (K=3) and SVM classifier, as well as the number of reduction sets for ARM algorithm, were studied for nine datasets in Table 1. The results are shown in Figure 1. The selection rule for thresholds is based on the minimum and maximum values of internal attribute importance, divided into ten equal steps, and removing attributes with zero importance.

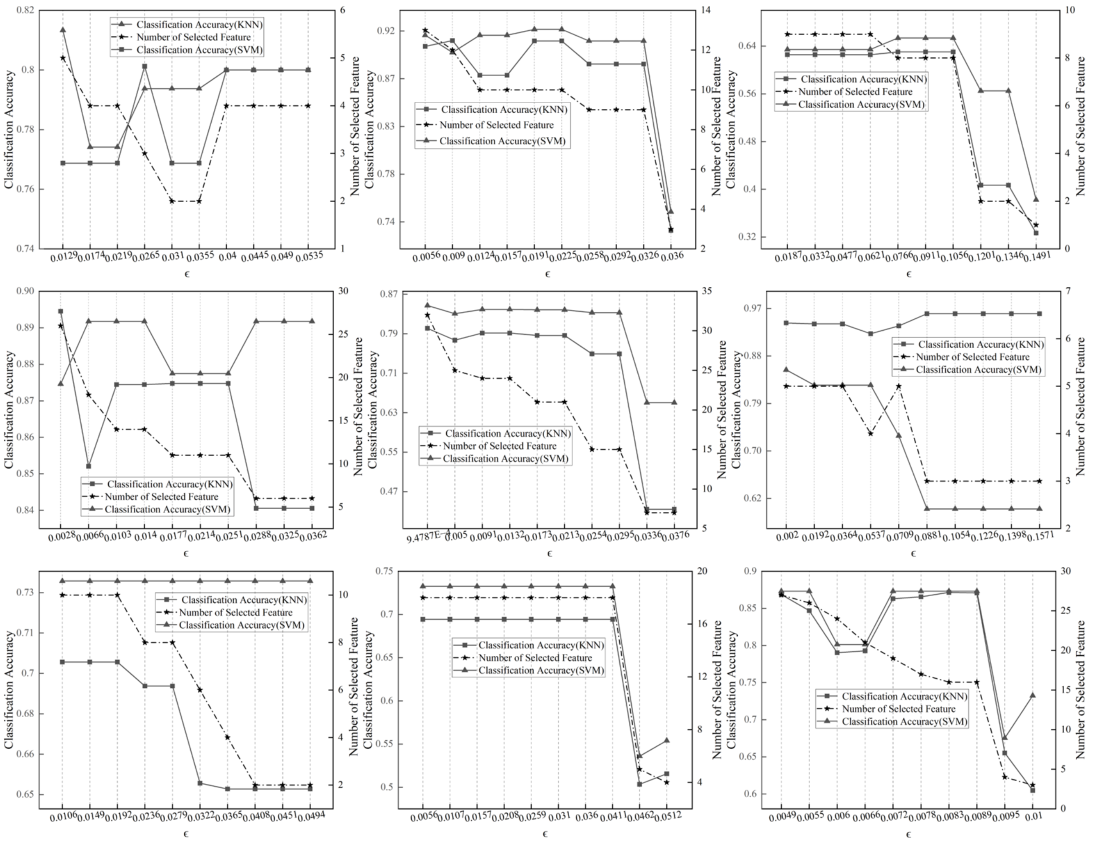

Figure 1. Classification accuracy and number of feature under different ϵ values

The experimental results show that there is no strict unified rule between the size of the threshold ϵ and the classification accuracy. Overall, as the threshold ϵ increases, the KNN classification accuracy and SVM classification accuracy show a downward trend. This is because when the threshold ϵ is large, the selection of kernel attributes is more stringent, and some important attributes are deleted. The information learned by KNN and SVM is limited, which leads to a decrease in classification accuracy. When the threshold is small, the selection of kernel attributes is relatively loose, and there are more attributes in the kernel attribute set, resulting in information systems having higher dimensional features. Therefore, KNN classifiers and SVM classifiers learn more information and have higher classification accuracy.

On the other hand, on datasets HE, WI, GL and HP, as the threshold ϵ increases, the KNN classification accuracy and SVM classification accuracy increase. This is because when adding attributes to the kernel attribute set in a bottom-up manner, attributes with high importance values of external attributes are added to the kernel attributes, resulting in an improvement in classification accuracy, even exceeding the classification accuracy of KNN and SVM classifiers on the original dataset. At the same time, the dimensionality of the dataset decreases, which proves the effectiveness and feasibility of the ARM algorithm.

4.3.2 Comparison experiment of reduction algorithms

In order to verify the effectiveness of the ARM algorithm, this section compares it with the Univariate Feature Selection (UFS) and Principal Component Analysis (PCA) algorithms. The classification accuracy of the reduced dataset and the original dataset is compared using K-Nearest Neighbor (KNN) classifier and Support Vector Machine (SVM) classifier. The results are shown in Table 4. The value of K in the KNN algorithm is 3. Reduction represents the number of attributes obtained through ARM algorithm.

The UFS algorithm can test each feature to be classified, measure the relation between the feature and the classification, select important features based on statistical relation, and obtain the top n optimal features for classification. PCA is a method of reducing the dimensionality of data by establishing a mapping from high-dimensional space to low dimensional space through covariance analysis. The dimension settings of UFS algorithm and PCA algorithm are consistent with those of ARM algorithm. The threshold selection in ARM algorithm is the ϵ value corresponding to the overall highest KNN classification accuracy and SVM classification accuracy in Section 4.3 experiment.

Table 4 The reduction results of different algorithms

Dataset	Attributes	ODP		Reduction	UFS		PCA		ARM	
		KNN	SVM		KNN	SVM	KNN	SVM	KNN	SVM
HE	19	0.7283	0.7925	5	**0.7283**	**0.7938**	**0.7283**	**0.7938**	**0.8046**	**0.8046**
WI	13	0.9052	0.916	10	0.8771	0.9049	**0.9105**	0.8993	**0.9105**	**0.9216**
GL	10	0.6255	0.6344	8	**0.6344**	**0.6535**	**0.6348**	0.6253	**0.6303**	**0.6535**
IO	33	0.866	0.8748	26	0.8632	0.8689	**0.8689**	0.8632	**0.8945**	0.8746
BI	41	0.8102	0.8321	32	0.8017	0.8302	**0.815**	0.8293	0.8008	**0.8463**
MU	22	**0.9655**	**0.9186**	5	0.9172	0.87	0.9393	0.8676	0.9413	0.8543
HP	27	0.8686	0.8731	16	**0.8721**	**0.8731**	0.8477	**0.8757**	**0.8715**	**0.8731**
SU	18	0.6944	0.7325	18	**0.6944**	**0.7325**	0.6949	0.7294	**0.6944**	**0.7325**
MA	10	0.7018	0.7359	10	**0.7018**	**0.7359**	**0.7277**	**0.7359**	**0.7018**	**0.7359**
AVG	19.3	0.7962	0.8122	12.9	0.7878	0.807	0.7963	0.8022	**0.8055**	0.8107

From the overall experimental mean, the reduced classification accuracy of ARM algorithm is better than UFS algorithm and PCA algorithm, with a 2.2% (KNN) and 0.5% (SVM) higher than UFS algorithm, and 1.1% (KNN) and 1% (SVM) higher than PCA algorithm. Compared with the original dataset, the classification accuracy is 1.2% higher (KNN), and the average reduction effect reaches 33%. Although the average SVM classification accuracy has decreased by 0.2%, overall, the classification accuracy on KNN and SVM classifiers is better. Among the nine datasets, seven datasets were equal to or higher than the classification accuracy of the original data.

In terms of attribute reduction, the ARM algorithm has the better reduction effect on the HE and MU datasets, indicating a high level of redundant information in the datasets. The attribute reduction effect of the HE dataset reaches 70%, and the classification accuracy is improved. Although the classification accuracy of the MU dataset has decreased, it is still within an acceptable range, and the attribute reduction effect has reached 77%. The lack of reduction effect in the SU and MA datasets indicates that there are no redundant attributes in the dataset.

In terms of classification accuracy, among the 9 sets of experimental data, the ARM algorithm had seven datasets with KNN and SVM classifiers and classification accuracy higher than or equal to the original dataset, while the UFS algorithm had only five datasets with KNN and SVM classifiers and classification accuracy higher than or equal to the original dataset. The PCA algorithm had six datasets with KNN and three datasets with SVM classifiers and classification accuracy higher than the original dataset, indicating that the ARM algorithm has better classification performance compared to the UFS algorithm and PCA algorithm.

In addition, the accuracy Pre (Precision), recall (Recall), and F1 (F1 score) indicators were compared, as shown in Table 5. In nine comparative experiments, on the KNN classifier, the ARM algorithm had six datasets with metrics higher than or equal to the original dataset; On the SVM classifier, there are seven datasets with metrics higher than or equal to the original dataset. This indicates that the ARM algorithm has shown good performance in attribute reduction.

Table 5 Performance metrics of ODP and ISM algorithm processed datasets on KNN and SVM classifiers

Dataset	Method	KNN			SVM		
		Pre	Rec	F1	Pre	Rec	F1
HE	ODP	0.7383	0.7218	0.7082	0.6395	0.6462	0.6234
	ARM	0.7117	0.6881	0.6781	**0.7312**	**0.6881**	**0.6912**
WI	ODP	0.9201	0.9104	0.9052	0.929	0.9211	0.9169
	ARM	**0.9333**	**0.9125**	**0.908**	**0.9344**	**0.9277**	**0.9229**
GL	ODP	0.5364	0.5494	0.5134	0.5476	0.5847	0.5384
	ARM	**0.5868**	**0.5751**	**0.5482**	**0.5566**	**0.5942**	**0.548**
IO	ODP	0.7356	0.7121	0.6996	0.7288	0.6776	0.6866
	ARM	**0.7717**	**0.7171**	**0.7241**	0.4959	0.5012	0.4862
BI	ODP	0.7943	0.7975	0.7908	0.8191	0.8049	0.8082
	ARM	0.7833	0.7931	0.7826	**0.8348**	**0.822**	**0.8249**
MU	ODP	0.9751	0.965	0.9635	0.9425	0.9183	0.9121
	ARM	0.9633	0.9417	0.9356	0.8854	0.8532	0.8447
HP	ODP	0.8691	0.8688	0.8687	0.8733	0.873	0.8731
	ARM	**0.8721**	**0.8716**	**0.8715**	**0.8733**	**0.873**	**0.8731**
SU	ODP	0.6947	0.6885	0.689	0.7329	0.7282	0.7289
	ARM	**0.6947**	**0.6885**	**0.689**	**0.7329**	**0.7282**	**0.7289**
MA	ODP	0.7895	0.5808	0.5503	0.7136	0.7227	0.7169
	ARM	**0.7895**	**0.5808**	**0.5503**	**0.7136**	**0.7227**	**0.7169**

5. Conclusion

To reduce redundant samples and attributes in incomplete information systems, this paper first expands the tolerance relation, proposes the nearest tolerance relation, and vectorize the calculation process of the approximations. To achieve instance selection and attribute reduction in information systems, an instance selection algorithm was designed based on the lower approximation results. In addition, the concept of attribute importance was introduced, and the attribute reduction algorithm was designed in a bottom-up manner. Finally, the experiment proved that the instance selection algorithm and attribute reduction algorithm have improved various performance indicators compared to the original dataset. This model provides a new approach for handling incomplete data, and in future work, more and better rough set models will be further studied to handle incomplete data and improve its approximation quality.

References

[1] PAWLAK Z. Rough sets[J]. International Journal of Computer & Information Sciences, 1982, 11(5): 341-356.

[2] Guoqiang Wang, Tianrui Li, Pengfei Zhang, Qianqian Huang and Hongmei Chen. Double-local rough sets for efficient data mining[J]. Information Sciences, 2021, 571: 475-498.

[3] Shuyin Li, Yang Liu. Classification Rule Mining Algorithm for Weighted Fuzzy Rough Sets[J]. Computer Engineering, 2019, 45(9): 211-215.

[4] Kryszkiewicz M, Rough Set Approach to Incomplete Information System[J]. Information Science, 1998, 11(2): 39-49.

[5] STEFANOWSKI J, TSOUKIAS A. Incomplete Information Tables and Rough Classification[J]. Computational Intelligence, 2011, 17(3): 545-566.

[6] Yi Xu and Shanzhong Hu. Extended Rough Set Model Based on Modified Data-driven Valued Tolerance Relation[J]. Journal of Intelligent & Fuzzy Systems, 2019, 36(2): 1615-1625.

[7] DERIS M M, HAMID M A, NORAINI I, et al. Data Reduction Using Similarity Class and Enhanced Tolerance Relation for Complete and Incomplete Information Systems[C]//Proceedings of the 2019 10th International Conference on Information and Communication Systems, Irbid, June 11-13, 2019: 134-139.

[8] Shangzhi Wu, Litai Wang, Shuyue Ge, Zheng Xiong and Jie Liu. Feature Selection Algorithm Using Neighborhood Equivalence Tolerance Relation for Incomplete Decision Systems[J]. Applied Soft Computing, 2024, 157: 111463.

[9] ROHMAT S R, HAIRULNIZAM M, SHAHREEN K, et al. A Relative Tolerance Relation of Rough Set for Incomplete Information System[C]//Proceedings of the 3rd International Conference on Soft Computing and Data Mining, Johor, February 6-8, 2018, 700:72-81.

[10] Wenhao Shu and Hong Shen. Incremental Feature Selection Based on Rough set in Dynamic Incomplete Data[J]. Pattern Recognition, 2014, 47(12): 3890-3906.

[11] Hailiang Zhang and Runliang Jia. Dynamic Attribute Reduction Algorithm Based on Neighborhood Dominance Rough Set[J]. Computer Engineering and Design, 2024, 45(08): 2320-2328.

[12] Lianhui Luo, Jilin Yang, Xianyong Zhang and Junfang Luo. Tri-level Attribute Reduction Based on Neighborhood Rough Sets[J]. Applied Intelligence, 2024, 54(5): 3786-3807.

[13] Jiucheng Xu, Shan Zhang and Qing Bai. Attribute Reduction Algorithm Based on Fuzzy Neighborhood Relative Decision Entropy [J/OL]. Computer Science, 1-13

[14] Bingying Xia and Chen Wu. Research on Attribute Reduction Algorithms Based on Knowledge Dependence by Tolerance Relation [J]. Journal of Jiangsu University of Science and Technology (Natural Science Edition), 2020, 34(02): 72-79.

[15] Jie Zhao, Yun Ling, Faliang Huang, Jiahai Wang and See-To Eric W.K. Incremental Feature Selection for Dynamic Incomplete Data Using Sub-tolerance Relations[J]. Pattern Recognition, 2024, 148: 110125.

[16] Wenhao Shu and Hong Shen. Incremental Feature Selection Based on Rough Set in Dynamic Incomplete Data[J]. Pattern Recognition, 2014, 47(12): 3890-3906.

[17] Ngoc N T, Sartra W. A Novel Feature Selection Method for High-Dimensional Mixed Decision Tables[J]. IEEE Transactions on Neural Networks and Learning Systems, 2022, 33(7): 3024-3037.

[18] Xiaojun Xie and Xiaolin Qin. A Novel Incremental Attribute Reduction Approach for Dynamic Incomplete Decision Systems[J]. International Journal of Approximate Reasoning, 2018, 93: 443-462.

[19] Chuan Luo, Tianrui Li, Hongmei Chen, Jianchen Lv and Yi Zhang. Fusing Entropy Measures for Dynamic Feature Selection in Incomplete Approximation Spaces[J]. Knowledge-Based Systems, 2022, 252: 109329.