

## Research Article

# Neighborhood Hypergraph Based Classification Algorithm for Incomplete Information System

**Feng Hu and Jin Shi**

*Chongqing Key Laboratory of Computational Intelligence, Chongqing University of Posts and Telecommunications, Chongqing 400065, China*

Correspondence should be addressed to Feng Hu; [hufeng@cqupt.edu.cn](mailto:hufeng@cqupt.edu.cn)

Received 5 March 2015; Revised 18 May 2015; Accepted 21 May 2015

Academic Editor: Evangelos J. Sapountzakis

Copyright © 2015 F. Hu and J. Shi. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The problem of classification in incomplete information system is a hot issue in intelligent information processing. Hypergraph is a new intelligent method for machine learning. However, it is hard to process the incomplete information system by the traditional hypergraph, which is due to two reasons: (1) the hyperedges are generated randomly in traditional hypergraph model; (2) the existing methods are unsuitable to deal with incomplete information system, for the sake of missing values in incomplete information system. In this paper, we propose a novel classification algorithm for incomplete information system based on hypergraph model and rough set theory. Firstly, we initialize the hypergraph. Second, we classify the training set by neighborhood hypergraph. Third, under the guidance of rough set, we replace the poor hyperedges. After that, we can obtain a good classifier. The proposed approach is tested on 15 data sets from UCI machine learning repository. Furthermore, it is compared with some existing methods, such as C4.5, SVM, NavieBayes, and KNN. The experimental results show that the proposed algorithm has better performance via Precision, Recall, AUC, and  $F$ -measure.

## 1. Introduction

A great deal of information system in reality life is incomplete information system [1]. When the precise value of some attributes in an information system is not known, that is, missing or known partially, such a system is called an incomplete information system (IIS). The problem of classification in incomplete information systems is a hot issue in intelligent information processing field. There are several approaches to deal with incomplete information systems. One of them is to remove samples with missing values. Another approach is to replace the missing value with the most common value [2]. These approaches are simple but they might destroy the original distribution of the data [3]. Other more complex approaches were presented in some literatures. Among these different data analysis theories and methods, rough sets [4] are the most frequently used. There are some extension models [5, 6] in rough set to deal with incomplete information system, such as tolerance relation, limited tolerance relation, and nonsymmetric similarity relation.

Hypernetwork was first proposed by Sheffi [7]. It has been presented as a probabilistic model of learning higher-order correlations using hypergraph structure consisting of a large number of hyperedges. Hypernetwork can be represented as hypergraph. Previous studies have shown that hypernetwork can be evolved to solve various machine learning problems. Segovia-Juarez et al. and Wang et al. [8, 9] use the hypernetwork model to realize DNA molecules; Kim and Zhang [10] use hypernetwork for pattern classification.

Previous researches have shown that the original hypergraph model has a good performance in classification. However, it still has some shortcomings: (1) the conventional hypergraph can only deal with the discrete data, and it still needs to discretize the continuous data. (2) The traditional hypergraph model has randomness in the process of creating new hyperedge. For incomplete information system, it is essential to supplement the missing value in the new hyperedge. The hypergraph takes measures like attribute value random filled, hyperedge random replacement strategies during

the process; it is more likely to impact the decision and classification ability of the training set.

To improve the problems mentioned above, we introduce the neighborhood rough set. Rough set theory, proposed by Pawlak in 1982 [11–13], can be seen as a new mathematical approach to vagueness. It has been successfully applied to various fields such as pattern recognition, machine learning, signal analysis, intelligent systems, decision analysis, knowledge discovery, and expert systems. The core concepts of rough set theory are approximations. Using the concepts of lower and upper approximations, knowledge hidden in information systems may be discovered and expressed in the form of decision rules. In other words, certain rules can be induced directly from the lower approximation, and possible rules can be derived from the upper approximation. So the study of approximation space has been developed widely. Once we apply rough set theory into hypergraph, we can supervise the hyperedge replacement process and improve the generalization ability of hyperedges as well. Lin [14] pointed out that neighborhood spaces are more general topological spaces than equivalence spaces and introduced neighborhood relation into rough set methodology. Hu et al. [15] discussed the properties of neighborhood approximation spaces and proposed the neighborhood-based rough set model. Then they used the model to build a uniform theoretic framework for neighborhood based classifiers. The neighborhood-based rough set solves the problem that classic rough set theory can not deal with the continuous data.

In this paper, we employ hypergraph model and rough set theory to build a neighborhood hypergraph model. After that, we propose a classification algorithm for incomplete information systems based on neighborhood hypergraph. This algorithm is composed of the following three steps. (1) Initialize the hyperedge set: generate hyperedges for every sample in the training set and process distinctively with samples which have missing values. (2) Classify training set: classify the training set with hyperedge set and determine whether to replace the hyperedges according to the accuracy of the classification. (3) Replace hyperedges: under the guidance of rough set, replace the unsuitable hyperedges. Compared to the algorithms implemented on WEKA platform with the existing methods, the experimental results show that the proposed classification algorithm is better than other algorithms.

The remainder of the paper is organized as follows. The basic concepts on hypergraph and neighborhood hypergraph models are shown in Section 2. The neighborhood hypergraph classifier algorithm for incomplete information system is developed in Section 3. Section 4 presents the experimental analysis. Finally, the paper is concluded in Section 5.

## 2. Hypergraph and Neighborhood Hypergraph Model

**2.1. The Definition of Hypergraph.** In 1970, Berge and Miniéka [16] used hypergraph to define hypernetwork. It was the first time to establish undirected hypergraph theory systematically and it was applied on the operations research by matroid.

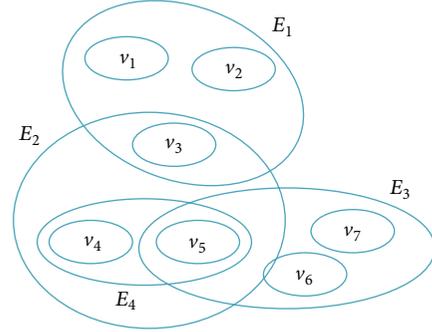


FIGURE 1: An example of hypergraph.

**Definition 1** (hypergraph [16]). Given  $V = \{v_1, v_2, \dots, v_n\}$  is finite set, if

- (1)  $E_i \neq \emptyset$  ( $i = 1, 2, \dots, m$ );
- (2)  $\bigcup_{i=1}^m E_i = V$ ,

then the binary relation  $H = \{E, V\}$  is defined as a hypergraph. The elements  $v_1, v_2, \dots, v_n$  of  $V$  are defined as vertices of the hypergraph and  $E = \{e_1, e_2, \dots, e_m\}$  is defined as the edge set of hypergraph.  $E_i = \{e_{i_1}, e_{i_2}, \dots, e_{i_j}\}$  ( $i = 1, 2, \dots, m$ ;  $j = 1, 2, \dots, n$ ) is defined as hyperedge (see Figure 1).

### 2.2. Neighborhood Hypergraph Based on IIS

**Definition 2** (the relevant degree between samples [17]). Let  $x_1$  and  $x_2$  be two samples in an incomplete system; each sample has  $N$  attributes;  $x_1$  and  $x_2$  have the same attribute value in  $S$  attributes, different value in  $P$  attributes, and uncertain value in  $F$  attributes (at least one of them has missing value at the attribute). We define  $S/N$  as the homogeneous degree of  $x_1$  and  $x_2$ ,  $P/N$  as the antagonisms degree of  $x_1$  and  $x_2$ , and  $F/N$  as the discrepant degree of  $x_1$  and  $x_2$ . We employ  $u(x_1, x_2) = S/N + (F/N)i + (P/N)j$  to represent the relevant degree of  $x_1$  and  $x_2$ ; namely,

$$u(x_1, x_2) = a + bi + cj, \quad (1)$$

where  $a + b + c = 1$ ,  $i \in [-1, 1]$ ,  $j \in [-1, 1]$ ,  $u$  denotes the relevant degree of  $x_1$  and  $x_2$ .

However, the relevant degree mentioned in paper [17] can only be used to compute discrete features and the continuous features are invalid. Here, we define that two continuous attribute values are considered equal if they fluctuate within an certain range in comparison.

For instance,  $U = \{x_1, x_2, x_3, x_4\}$ ,  $c$  is an attribute of  $U$ , and  $f(x, c)$  denotes the value on attribute  $c$  of sample  $x$ ;  $f(x_1, c) = 0.3$ ,  $f(x_2, c) = 0.27$ ,  $f(x_3, c) = 0.35$ ,  $f(x_4, c) = 0.1$ , and  $h = 0.05$ . So for the attribute  $c$ , if only  $|x_i - x_1| \leq h$ ,  $x_i$  is equal to  $x_1$  on attribute  $c$ . Obviously, we can find that  $x_1, x_2$ , and  $x_3$  are equivalent on attribute  $c$ , while they are not equal to  $x_4$  on attribute  $c$ .

**Definition 3** (the neighborhood of a sample in IIS). Lin [14] pointed out the neighborhood model in 1988. Hu et al. [15] discussed the properties of neighborhood approximation

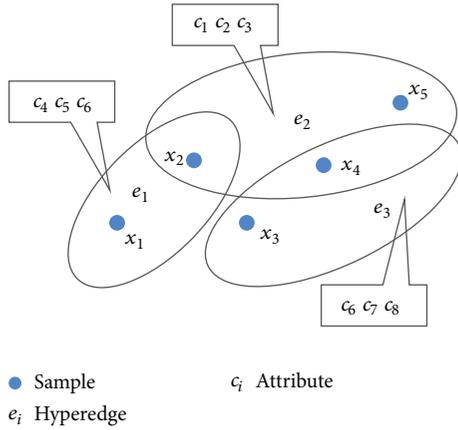


FIGURE 2: An example of neighborhood hypergraph.

spaces and proposed the neighborhood-based rough set model. Hu et al. [18] employed the neighborhood rough set to classify data by using the Minkowski distance to calculate the samples neighborhood threshold which involves all the attribute values of the one calculated. However, for incomplete information system, it is difficult to compute the distance for the sake of the missing value. Thus, we present an extension neighborhood of sample in incomplete information system by combining with relevant degree.

Given arbitrary  $x_i \in U$  and  $B \subseteq C$ , the neighborhood  $\delta_B(x_i)$  of  $x_i$  on attribute set  $B$  is defined as

$$\delta_B(x_i) = \{x \mid x \in U, u_B(x, x_i) \geq \delta\}, \quad (2)$$

where  $u_B(x, x_i)$  is the relevant degree between  $x$  and  $x_i$ .  $\delta_B(x_i)$  denotes the sample set within the neighborhood of  $x_i$ .

According to the definition, we can find out easily that

- (1)  $\forall x_i \in U, \delta_B(x_i) \neq \emptyset$ ;
- (2)  $\forall x_i \in U, \forall x_j \in U (x_j \in \delta_B(x_i) \leftrightarrow x_i \in \delta_B(x_j))$ ;
- (3)  $\bigcup_{x_i \in U} \delta_B(x_i) = U$ .

Combining with the neighborhood rough set theory, we define the neighborhood hypergraph as follows.

**Definition 4** (the neighborhood hypergraph of IIS). Let  $G = \langle X, E \rangle$  be a neighborhood hypergraph of incomplete information system, referred to as neighborhood hypergraph. Then  $X = \{x_1, x_2, \dots, x_n\}$  is the vertex set of  $G$ , indicating that it has  $n$  vertices,  $E = \{e_1, e_2, \dots, e_n\}$  is hyperedge set, and each  $e_i$  in  $E$  is a hyperedge which connects  $k$  vertices  $(x_{i1}, x_{i2}, \dots, x_{ik})$ .  $C = \{c_1, c_2, \dots, c_m\}$  is the attribute set, and  $D$  is the decision set, where  $x_i$  denotes sample.

Vertices of hypergraph represent the attribution of samples in some literatures like [19] and so on. However, in this paper, vertices of hypergraph are denoted as samples and different samples on one hyperedge have the same attributes set (see Figure 2).

**Definition 5** (the sample in neighborhood hypergraph). Given  $G = \langle X, E \rangle$ ,  $C = \{c_1, c_2, \dots, c_m\}$  denotes the attribute set

of hyperedge,  $x = \{c_1(x), c_2(x), c_3(x), \dots, c_p(x), D(x), \delta_B\}$  is a sample, where  $c_i(x)$  denotes the values of  $x$  on the attribute  $c_i$  ( $c_i \in C$ ),  $D(x)$  denotes the decisions of  $x$ , and  $\delta_B$  is the threshold of a neighborhood.

**Definition 6** (the neighborhood hyperedge set of a sample). Given  $G = \langle X, E \rangle$  and the attribute set  $B$  ( $B \subseteq C$ ), the hyperedge set which is included by sample  $x$  is defined as

$$\delta_B(x) = \{e \mid (e \in E) \wedge u(e, x) \geq \delta\}. \quad (3)$$

**Definition 7** (the sample set related to a hyperedge). Given  $G = \langle X, E \rangle$ ,  $\forall e \in E$ , and attributes set  $B$  ( $B \subseteq C$ ), for arbitrary  $e \in E$ , the sample set related to  $e$  is defined as  $R_B(e) = \{x \mid e \in \delta_B(x), x \in X\}$ . Given arbitrary  $Y \subseteq E$  and attributes set  $B$  ( $B \subseteq C$ ), the sample set related to  $Y$  is defined as

$$R_B(Y) = \{R_B(e) \mid e \in Y\}. \quad (4)$$

**Definition 8** (the confidence degree of a hyperedge). Given  $G = \langle X, E \rangle$ , for arbitrary  $e \in E$ , assume that  $D(e) = \{d \mid d \in D\}$ , where  $D$  denotes decision set.  $R_B(e)$  is the sample set related to hyperedge  $e$  on attributes set  $B$  ( $B \subseteq C$ ). According to the decisions  $D$ ,  $R_B(e)$  is divided into  $p$  equivalence classes:  $X_1, X_2, \dots, X_p$ ; when  $R_B(e) \neq \emptyset$ , the confidence degree of  $e$  is defined as follows:

$$\text{Conf}_B(e) = \frac{|\{x \mid x \in R_B(e), D(x) = D(e)\}|}{|\{x \mid x \in R_B(e)\}|}. \quad (5)$$

**Definition 9** (the upper approximation, lower approximation, boundary region, and negative domains of hyperedge set for the sample decision set). Given  $G = \langle X, E \rangle$ ,  $C$  is the attitudes set of samples and  $D$  is the decision set of samples. For arbitrary hyperedge set  $E'$  ( $E' \subseteq E$ ), according to the decisions  $D$ , the hyperedge set  $E'$  is divided into  $p$  equivalence classes:  $E_1, E_2, \dots, E_p$ . For arbitrary  $B \subseteq C$ , the upper approximation, lower approximation, boundary region, and negative domains of decision  $D$  related to set of attributes  $B$  are, respectively, defined as

$$\begin{aligned} \overline{N}_B(D) &= \bigcup_{i=1}^p \{e \mid R_B(e) \cap R_B(E_i) \neq \emptyset \vee (e \in E_i), e \in E'\}, \\ \underline{N}_B(D) &= \bigcup_{i=1}^p \{e \mid \text{Conf}_B(e) > \lambda, e \in E_i\}, \end{aligned} \quad (6)$$

$(0 \leq \lambda \leq 1),$

$$BN_B(D) = \overline{N}_B(D) - \underline{N}_B(D)$$

$$\text{Neg}_B(D) = E - \overline{N}_B(D).$$

The lower approximation of decisions  $D$  related to attribute set  $C$  is also called positive domain. The size of positive domain reflects the separable degree of classification problem in a given attribute space and the bigger the positive region is, the smaller the border is.

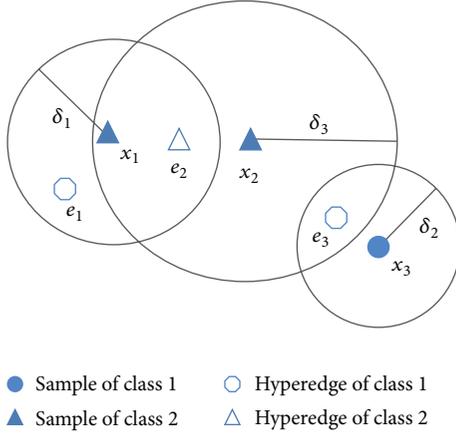


FIGURE 3: An example of upper and lower approximation and boundary region.

To explain how to divide the upper approximation, lower approximation, and boundary region, here we give an example (see Example 10).

*Example 10.* Given  $G = \langle X, E \rangle$ , where  $X = \{x_1, x_2, x_3\}$ ,  $E = \{e_1, e_2, e_3\}$ ,  $\lambda = 0.5$  (see Figure 3).

First, we calculate the sample set  $R_B(e_i)$  of each hyperedge:  $R_B(e_1) = \{x_1\}$ ,  $R_B(e_2) = \{x_1, x_2\}$ , and  $R_B(e_3) = \{x_2, x_3\}$ , according to formula (4) and Definition 7. Second, we calculate the confidence degree of each hyperedge in terms of formula (5) and Definition 8:  $\text{Conf}_B(e_1) = 0$ ,  $\text{Conf}_B(e_2) = 1$ , and  $\text{Conf}_B(e_3) = 0.5$ . Third, according to formula (6) and Definition 9, we can obtain the upper approximation, the lower approximation, and the boundary region of  $G = \langle X, E \rangle$ , respectively:  $\overline{N}_B(D) = \{e_1, e_2, e_3\}$ ,  $\underline{N}_B(D) = \{e_2\}$ , and  $BN_B(D) = \{e_1, e_3\}$ .

### 3. A Classification Algorithm Based on Neighborhood Hypergraph for IIS

The process of the hypergraph model classification is generally divided into 3 steps. (1) Step 1: initialize hypergraph. (2) Step 2: classify the training set. (3) Step 3: replace hyperedges: to improve the accuracy of classification by replacing the unsuitable hyperedges. Steps 2 and 3 are iterative process, which will stop as far as the accuracy reaches a specific threshold. Generating hyperedge is a random process in the traditional hypergraph classification method, which makes it difficult to search and replace the bad performance hyperedges at the hyperedge replacement phase. Therefore, we introduce the rough set theory to separate the hyperedge set into three parts, including the upper approximation, lower approximation, and boundary region. In addition, the hyperedges in the boundary region will be tackled specially, which will result in the improvement of classification accuracy. The flow chart of this algorithm can be seen in Figure 4.

**3.1. Initialize Hypergraph.** Initialize hypergraph, namely, to generate hyperedges based on samples. For each new hyperedge, we need to consider the initialization of both condition attributes and decision attributes.

In incomplete information system, there are two kinds of samples. One does not have missing values, while the other has missing values. We may call them type 1 and type 2, respectively. For type 1, we create three hyperedges for each sample while we create five new hyperedges for type 2. Furthermore, we need to fill the missing value in the new hyperedge. The specific process of creating new hyperedge is as follows.

(1) *Inherit the Condition Attributes Randomly.* The attribute number of hyperedge is the same as the sample. The attribute value of hyperedge is inherited from the sample. Given a sample, we select 70% of attributes in hyperedge randomly and inherit their values from the sample. For the values of the rest of attributes in hyperedge, we can process as follows. If the attribute is continuous, this attribute in the hyperedge will inherit the average of the attribute values from all samples. If the attribute is discrete, we generate the value randomly in the domain of the attribute value.

For the samples having missing values, we need to fill the values in the new hyperedge if the value is inherited from the sample which is missing. If the attribute value is continuous, we can fill the attribute value with the average value of all samples whose decision value is equal to the hyperedge. Otherwise, we can fill the attribute value with the most frequent attribute value.

(2) *Inherit Decision Attribute Directly.* The decision attribute of the new hyperedge is inherited directly from the sample which creates the hyperedge.

**3.2. Classify the Training Set.** Given a simple  $x$ , the neighborhood threshold of  $x$  is defined as follows:

$$\delta = \max \{u(x, x_i) - r, \quad (7)$$

where  $(x_i \in X) \cap (x_i \neq x)$ ,  $r$  is a fixed value.

For each sample in the training set, its classification can be determined by the voting result among its neighborhood hyperedge set. In order to get better hyperedge, we can compute the accuracy by analyzing the classification result of the training set. If the accuracy is higher than the threshold, we can output the hyperedge set. Otherwise, the hyperedge should be replaced.

*Definition 11* (the decision subset of  $\delta_B(x)$ ). Given  $G = \langle X, E \rangle$ ,  $\delta_B(x)$  is the neighborhood hyperedge set of sample  $x$ ; the set whose decision is  $j$  in  $\delta_B(x)$  is defined as follows:

$$\delta_B(x)^j = \{e \mid e \in \delta_B(x) \wedge D(e) = j\}. \quad (8)$$

Given a sample  $x$ , its classification process is as follows:

- (1) Compute the neighborhood hyperedge set of sample  $x$ :  $\delta_B(x)$ .
- (2) Classify the hyperedges in the set  $\delta_B(x)$ . For instance, add the hyperedge of classification  $j$  into  $\delta_B(x)^j$ .

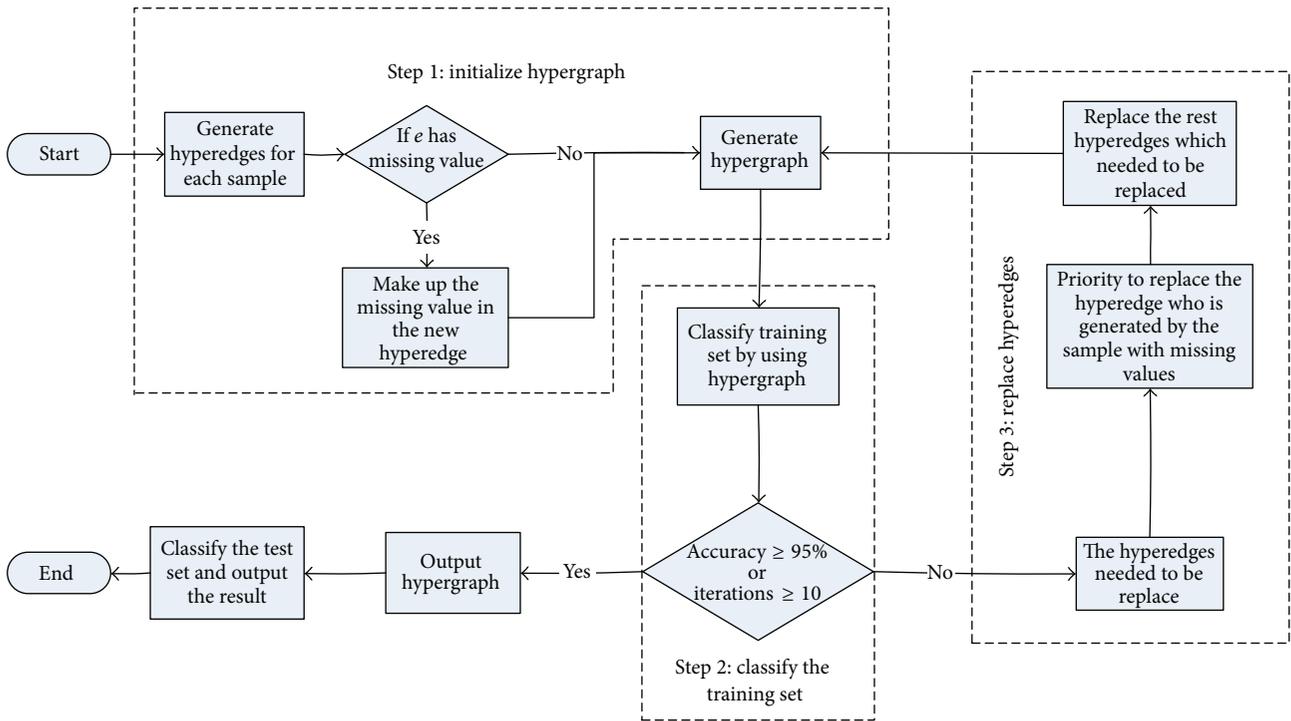


FIGURE 4: The flow chart of this algorithm.

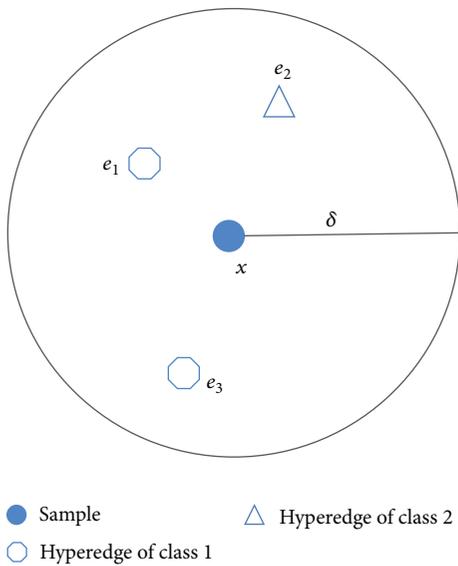


FIGURE 5: An instance of classifying a sample.

(3) Determine the classification of  $x$ ,  $D(x) = \operatorname{argmax}(|\delta_B(x)^j|)$ , where  $j = \{1, 2, 3, \dots\}$  is the decision attribute value.

*Example 12.* As it is shown in Figure 5, given a sample  $x$  and hyperedges  $e_1, e_2, e_3$ , we have the following.

According to Definition 6, we can obtain  $\delta_B(x) = \{e_1, e_2, e_3\}$ .

According to Definition 11, we can obtain  $\delta_B(x)^1 = \{e_1, e_3\}$ ,  $\delta_B(x)^2 = \{e_2\}$ .

So,  $D(x) = \operatorname{argmax}(|\delta_B(x)^1|, |\delta_B(x)^2|)$  and the classification of sample  $x$  is 1.

We use the classification rules above to classify the training set. Once the accuracy is no less than 0.95 or the iterations are not less than 10 times, we output the hypergraph. On the contrary, we should replace the unsuitable hyperedges.

**3.3. Replace Hyperedges.** In the process of hyperedge initialization, we fill the missing value and replace some attribute value in hyperedge. As a result, some of hyperedges are not suitable for sample classification. In order to acquire better performance, we should replace the poor hyperedges by generating new hyperedges, namely, replacing hyperedge.

The rough set theory divides hyperedge set into upper approximation, lower approximation, boundary region, and negative region. The confidence degree of hyperedges in lower approximation is 1. The confidence degree of hyperedges in boundary region is between 0 and 1. Hyperedges whose confidence degree is 0 belong to negative region. Hyperedges in lower approximation are all retained because they are very helpful for classification. On the contrary, since hyperedges in negative region are counteractive for classification, they will be replaced. For the hyperedges in boundary region, they will be dealt with by a threshold. When the confidence degree of a hyperedge is less than the threshold, it will be replaced. Through the above, we can enhance the pertinence and validity of hyperedge replacement. While we replace the hyperedge, it is prior to replace the hyperedge which is generated by the sample with missing values.

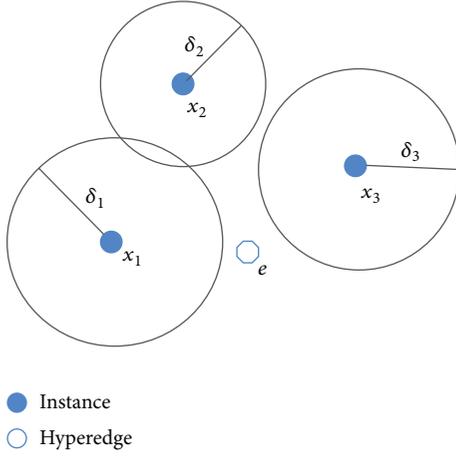


FIGURE 6: An example of Case 1.

3.3.1. *Search the Unsuitable Hyperedges.* Consider the following

- (1) Set the confidence degree threshold for hyperedge:  $\lambda = 0.5$ .
- (2) Find out the hyperedges whose confidence degree is under the threshold from the hyperedge set.

Given a hyperedge  $e$ , according to Definitions 7 and 8, we can calculate the confidence degree of hyperedge  $e$  by the following three cases.

*Case 1.* If  $R_B(e) = \emptyset$ , then no sample is related to the hyperedge (see Figure 6).

In this situation, we find out five samples whose relevant degree with  $e$  is the maximum (not all the five samples are shown in Figure 6) and we assume that  $e$  is related to this five samples. Then calculate the confidence degree according to formula (5). If  $\text{Conf}_B(e) > \lambda$ , keep  $e$ ; otherwise replace  $e$ .

*Case 2.* If  $\text{Conf}_B(e) > \lambda$ , we note that we can keep hyperedge  $e$ .

*Case 3.* If  $0 \leq \text{Conf}_B(e) \leq \lambda$ , we note that hyperedge  $e$  needs to be replaced.

Now, we present an example to illustrate the process (see Example 13).

*Example 13.* Given  $G = \langle X, E \rangle$ , where attribute set  $B = C$ ,  $X = \{x_1, x_2, x_3\}$ ,  $E = \{e_1, e_2\}$  (see Figure 7).

According to the formulas (3) and (4), we know  $R_B(e_1) = \{x_1, x_2, x_3\}$  and  $R_B(e_2) = \{x_3\}$ . After that, we calculate the confidence degree in terms of formula (5);  $\text{Conf}_B(e_1) = 1/3$  and  $\text{Conf}_B(e_2) = 0$ .

This kind of hyperedges has the same classification with minor samples surrounding them which results in the poor effect on classification. Thus, they should be replaced.

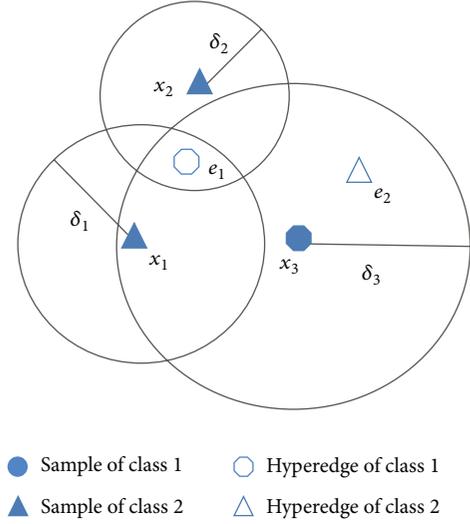


FIGURE 7: An example of Case 3.

3.3.2. *Replace the Unsuitable Hyperedges.* Take one hyperedge  $e_i$  out from the hyperedge replacement set and get the sample  $x$  which generates the hyperedge. Generate a new hyperedge  $e_j$  using this sample  $x$  and replace  $e_i$  with  $e_j$ .

It is worth mentioning that we need to preferentially replace those hyperedges created by sample with missing values when replacing.

3.4. *Algorithm Description.* See Algorithm 1.

## 4. Experimental Analyses

4.1. *Data Sets.* Experimental analysis is conducted on 15 UCI [20] data sets. There are four incomplete data sets, namely, Mammographic, Credit Approval, Hungarian, Postoperative, and eleven complete data sets. Complete data sets are modified to obtain incomplete data sets, by random missing some attribute values. The missing degree ranges from 0.6% to 18.97%. The data sets are outlined in Table 1 (sorted by the size of the data set).

4.2. *Experimental Evaluation.* In order to evaluate the performance of the developed approach, we use accuracy, Precision, Recall,  $F$ -measure [21], and area under ROC curve to evaluate the performance of classifier.

We assume that

$A$  is the number of the relevant samples that are predicted;

$B$  is the number of the irrelevant samples that are predicted;

$C$  is the number of the relevant samples that are not predicted;

$D$  is the number of the irrelevant samples that are not predicted.

```

Input: Training set  $X$ 
Output: Hyper-edge set  $E$ 
Step 1. (Initialize hypergraph)
  FOR each  $x$  in  $X$  DO
    Create one hyper-edge  $e$  of sample  $x$ : First, Inherit attributes from the sample randomly
    and replace the values randomly on the rest attribute. Second,  $e$  inherits the decision
    attribute of  $x$ . Third, if  $e$  has missing value, we fill the attribute value in terms with
    continuous attributes or discrete attribute.
     $E = E \cup \{e\}$ ;
  END FOR
Step 2. (Classify the training set)
  According to formula (7), calculating the neighborhood threshold  $\delta$  for each sample;
  FOR each  $x$  in  $X$  DO
    FOR each  $e$  in  $E$  DO
      According to formula (1), calculate the relevant degree of  $x$  and  $e$ ,  $u(x, e)$ .
      IF  $u(x, e) \geq \delta$  THEN  $\delta_B(x) = \delta_B(x) \cup \{e\}$ ; END IF
    END FOR
    FOR each  $e$  in  $\delta_B(x)$  DO
      IF  $D(e) == j$  THEN
         $\delta_B(x)^j = \delta_B(x)^j \cup \{e\}$ ; //  $j$  is the decision attribute value
      END IF
    END FOR
    Compute the classification of  $x$ ,  $D(x) = \operatorname{argmax}(|\delta_B(x)^j|)$ .
  END FOR
  Compute the correctly classified ratio of the training set: accuracy;
  IF accuracy  $\geq 0.95$  or iterations  $\geq 10$  THEN GOTO Step 4;
  ELSE GOTO Step 3;
  END IF
Step 3. (Replace hyper-edge)
  replace = 0; //the initialize of hyper-edge replacement set
  FOR each  $e_i$  in  $E$  DO
    According to Definition 7 and formula (5), calculate the confidence degree of  $e_i$ :  $\operatorname{Conf}_B(e_i)$ .
    IF  $\operatorname{Conf}_B(e_i) \leq 0.5$  THEN replace = replace + 1;  $E = E - \{e_i\}$ ; END IF
  END FOR
  While we replace the hyper-edge, it is prior to replace the hyper-edge which is generated by
  the sample with missing values.
  WHILE (replace  $\neq 0$ )
    Generate a new hyper-edge  $e_j$  through the process similar to Step 1.
     $E = E \cup \{e_j\}$ ; replace = replace - 1;
  END WHILE
  GOTO Step 2;
Step 4. (Return)
  RETURN  $E$ ;

```

ALGORITHM 1: A classification algorithm for incomplete information system based on neighborhood hypergraph.

Precision or confidence denotes the proportion of predicted samples that are relevant samples:

$$\text{Precision} = \frac{A}{A + B}. \quad (9)$$

Recall or sensitivity denotes the proportion of relevant samples that are correctly predicted:

$$\text{Recall} = \frac{A}{A + C}. \quad (10)$$

We expect the Precision and the Recall value are both high. In fact, they conflicted. The two values can not be high at the same time. So we use  $F$ -measure to consider them

comprehensively;  $F$ -measure is the harmonic mean of Precision and Recall:

$$F\text{-measure} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (11)$$

Another appropriate metric that could be used to measure the performance of classification is Receiver Operating Characteristic (ROC) [22] graphics. In these graphics, the tradeoff between the benefits and costs can be visualized. The area under the ROC curve (AUC) [23] corresponds to the probability of correctly identifying which of the two stimuli is noise and which is signal plus noise. AUC provides a single number summary of the performance of learning algorithms.

TABLE 1: Experimental data sets (C: continuous, N: nominal).

Number	Data set	Size	Attribute	Missing degree	Class label
1	Mammographic	961	1C 4N	4.25%	0:1
2	Blood	748	4C	5%	0:1
3	Credit Approval	690	6C 9N	0.6%	+:-
4	Balance	625	4N	5%	0:1:2
5	ILPD	583	9C 1N	5%	1:2
6	Haberman	306	3C	5%	1:2
7	Hungarian	294	5C 8N	18.97%	0:1
8	Statlog Heart	270	8C 5N	5%	1:2
9	Spect Heart	267	22N	5%	0:1
10	Bankruptcy	250	6N	5%	B:NB
11	Sonar	208	60C	5%	R:M
12	Iris	150	4C	5%	Iris-setosa:Iris-versicolor:Iris-virginica
13	Fertility	100	3C 6N	5%	N:O
14	Postoperative	90	1C 7N	3%	I:S:A
15	Lenses	24	4N	5%	1:2:3

TABLE 2: Accuracy.

No.	Data set	C4.5	SVM	NaiveBayes	KNN	Rough (incomplete)	Rough (complete)	HyperGraph
1	Mammographic	0.823	0.817	0.834	0.798	0.680	0.649	0.834
2	Blood	0.775	0.762	0.751	0.721	0.600	0.480	0.765
3	Credit Approval	0.861	0.849	0.777	0.846	0.609	0.217	0.861
4	Balance	0.653	0.878	0.890	0.819	0.635	0.635	0.861
5	ILPD	0.703	0.714	0.559	0.638	0.310	0.069	0.705
6	Haberman	0.722	0.735	0.748	0.680	0.548	0.258	0.752
7	Hungarian	0.810	0.827	0.837	0.769	0.400	0.233	0.845
8	Statlog Heart	0.762	0.810	0.829	0.795	0.259	0.259	0.843
9	Spect Heart	0.809	0.824	0.794	0.764	0.556	0.704	0.841
10	Bankruptcy	0.972	0.988	0.992	0.996	0.998	0.840	0.996
11	Sonar	0.755	0.764	0.678	0.760	0.429	0.715	0.792
12	Iris	0.913	0.927	0.953	0.880	0.933	0.667	0.960
13	Fertility	0.880	0.880	0.880	0.820	0.880	0.500	0.880
14	Postoperative	0.700	0.689	0.678	0.656	0.333	0.556	0.711
15	Lenses	0.750	0.583	0.667	0.667	0.667	0.333	0.733
	Average	0.793	0.803	0.791	0.774	0.584	0.474	0.825

Most of time, the value of AUC is from 0.5 to 1.0: the higher, the better. When the AUC value is under 0.5, it means the classifier has no positive effect on classification and it should be abandoned.

*4.3. Experimental Method and Results.* In order to evaluate the performance of HyperGraph we point out, we compared it with some other algorithms in related literatures: C4.5, SVM, NaiveBayes, and KNN [24]. C4.5, SVM, and NBC are all classic classification algorithms. They have great performance for the majority of data sets in both theory and practice. KNN is a simple and classic algorithm; we choose it because the main idea of KNN is majority voting with near samples, which are very similar to the proposed methods. Their source codes are afforded by Weka software [25]. What is more is that we also compared to two rough set methods whose source codes

are implemented in RIDAS [26]. One of the rough set algorithms directly handles the data sets by using rough set theory (labeled as “Rough (incomplete)” in Tables 2, 3, 4, 5, and 6) [27, 28]. The other algorithm based on rough set theory is to fill the missing values of the data sets before classification (labeled as “Rough (complete)” in Tables 2, 3, 4, 5, and 6) [29–31].

The proposed HyperGraph algorithm is implemented by JAVA. All the results are obtained from 10-fold cross validation [32].

Contrastive experiment results on accuracy, Precision, Recall,  $F$ -measure, and AUC within each algorithm are shown in Tables 2 to 7 and Figure 8.

we also compare the accuracy values of cases of whether missing values exist or not by using 22 data sets. There are 11 complete data sets and 11 incomplete data sets that derive

TABLE 3: Precision.

Number	Data set	C4.5	SVM	NaiveBayes	KNN	Rough (incomplete)	Rough (complete)	HyperGraph
1	Mammographic	0.823	0.825	0.836	0.798	0.723	0.777	0.834
2	Blood	0.751	0.581	0.706	0.714	0.722	0.775	0.732
3	Credit Approval	0.861	0.861	0.793	0.846	0.637	0.827	0.862
4	Balance	0.602	0.810	0.820	0.765	0.757	0.634	0.790
5	ILPD	0.658	0.509	0.793	0.607	0.539	0.476	0.655
6	Haberman	0.621	0.541	0.715	0.629	0.719	0.490	0.664
7	Hungarian	0.807	0.825	0.836	0.769	0.580	0.590	0.848
8	Statlog Heart	0.762	0.810	0.829	0.795	0.424	0.421	0.847
9	Spect Heart	0.792	0.818	0.841	0.817	0.633	0.737	0.854
10	Bankruptcy	0.973	0.988	0.992	0.996	0.998	0.998	0.997
11	Sonar	0.755	0.765	0.704	0.790	0.639	0.381	0.798
12	Iris	0.918	0.932	0.954	0.905	0.944	0.571	0.962
13	Fertility	0.774	0.774	0.774	0.832	0.830	0.733	0.774
14	Postoperative	0.503	0.501	0.499	0.494	0.427	0.444	0.506
15	Lenses	0.758	0.542	0.649	0.639	0.250	0.312	0.485
	Average	0.757	0.739	0.783	0.760	0.655	0.611	0.774

TABLE 4: Recall.

Number	Data set	C4.5	SVM	NaiveBayes	KNN	Rough (incomplete)	Rough (complete)	HyperGraph
1	Mammographic	0.823	0.817	0.834	0.798	0.680	0.649	0.834
2	Blood	0.775	0.762	0.751	0.721	0.600	0.480	0.765
3	Credit Approval	0.861	0.849	0.777	0.846	0.609	0.217	0.861
4	Balance	0.653	0.878	0.890	0.819	0.635	0.635	0.861
5	ILPD	0.703	0.714	0.559	0.638	0.310	0.069	0.705
6	Haberman	0.722	0.735	0.748	0.680	0.548	0.258	0.752
7	Hungarian	0.810	0.827	0.837	0.769	0.400	0.233	0.845
8	Statlog Heart	0.762	0.810	0.829	0.795	0.259	0.259	0.843
9	Spect Heart	0.809	0.824	0.794	0.764	0.556	0.704	0.841
10	Bankruptcy	0.972	0.988	0.992	0.996	0.998	0.840	0.996
11	Sonar	0.755	0.764	0.678	0.760	0.429	0.715	0.792
12	Iris	0.913	0.927	0.953	0.880	0.933	0.667	0.960
13	Fertility	0.880	0.880	0.880	0.820	0.880	0.500	0.880
14	Postoperative	0.700	0.689	0.678	0.656	0.333	0.556	0.711
15	Lenses	0.750	0.583	0.667	0.667	0.667	0.333	0.733
	Average	0.793	0.803	0.791	0.774	0.584	0.474	0.825

from each complete data set by a random missing process (the missing degree is 5%). The final results are shown in Table 7.

In order to view the performance on 5 algorithms, the average value of different indicator of 5 algorithms is shown in Figure 8.

From Tables 2 and 4 and Figure 8, we can figure out that HyperGraph has higher average accuracy, Recall, and  $F$ -measure. Furthermore, the average AUC value is superior to majority algorithms as shown in Tables 3 and 6 and Figure 8. And it is also indicated in Table 7 that the proposed method is suitable for incomplete information system. It still has good performance when the data set has missing values.

Tables 3 and 6 and Figure 8 show that NaiveBayes has higher average Precision and AUC value than HyperGraph, because NaiveBayes classifier classifies data by using the NBC

model. In theory, when the properties of sample are independent of each other, NBC model has the minimum misclassification error rate compared with other classification methods [33]. Majority of the data sets we use in this paper have independent properties, which makes the NaiveBayes have average lower misclassification error rate. Moreover, Precision and AUC are generally inversely proportional to the misclassification error rate, according to their definition. Thus, NaiveBayes has higher average Precision and average AUC than HyperGraph.

However, as indicated in Tables 3, 5, and 6, the proposed classifier on Precision,  $F$ -measure, and AUC value for lenses data set is poorer than most of the other algorithms. By analyzing the distribution of the lenses data set, we find out most attributes value are extremely approximate between the two

TABLE 5: *F*-measure.

No.	Data set	C4.5	SVM	NaiveBayes	KNN	Rough (incomplete)	Rough (complete)	HyperGraph
1	Mammographic	0.823	0.814	0.834	0.798	0.698	0.708	0.832
2	Blood	0.755	0.659	0.713	0.717	0.658	0.569	0.731
3	Credit Approval	0.861	0.850	0.769	0.846	0.622	0.303	0.859
4	Balance	0.626	0.878	0.853	0.786	0.695	0.646	0.823
5	ILPD	0.659	0.594	0.562	0.619	0.394	0.114	0.667
6	Haberman	0.633	0.623	0.700	0.646	0.612	0.380	0.670
7	Hungarian	0.806	0.823	0.836	0.769	0.461	0.320	0.839
8	Statlog Heart	0.760	0.810	0.828	0.794	0.328	0.347	0.833
9	Spect Heart	0.797	0.821	0.808	0.781	0.568	0.544	0.835
10	Bankruptcy	0.972	0.988	0.992	0.996	0.998	0.913	0.996
11	Sonar	0.754	0.764	0.673	0.749	0.483	0.444	0.786
12	Iris	0.913	0.926	0.953	0.881	0.933	0.611	0.958
13	Fertility	0.824	0.824	0.824	0.826	0.830	0.587	0.821
14	Postoperative	0.586	0.580	0.575	0.563	0.388	0.444	0.586
15	Lenses	0.752	0.554	0.656	0.641	0.300	0.313	0.531
	Average	0.768	0.767	0.772	0.761	0.598	0.483	0.784

TABLE 6: AUC.

Number	Data set	C4.5	SVM	NaiveBayes	KNN	Rough (incomplete)	Rough (complete)	HyperGraph
1	Mammographic	0.861	0.809	0.901	0.858	0.731	0.729	0.832
2	Blood	0.707	0.500	0.705	0.645	0.648	0.569	0.586
3	Credit Approval	0.887	0.856	0.896	0.873	0.629	0.566	0.882
4	Balance	0.695	0.877	0.951	0.879	0.762	0.713	0.872
5	ILPD	0.679	0.500	0.713	0.519	0.471	0.478	0.572
6	Haberman	0.533	0.500	0.644	0.568	0.567	0.525	0.547
7	Hungarian	0.752	0.794	0.906	0.733	0.536	0.512	0.814
8	Statlog Heart	0.751	0.806	0.897	0.851	0.470	0.561	0.827
9	Spect Heart	0.744	0.714	0.846	0.748	0.574	0.728	0.750
10	Bankruptcy	0.991	0.988	0.992	0.996	0.998	0.920	0.996
11	Sonar	0.788	0.761	0.798	0.762	0.589	0.500	0.789
12	Iris	0.971	0.960	0.995	0.923	0.953	0.800	0.972
13	Fertility	0.424	0.500	0.481	0.614	0.689	0.304	0.676
14	Postoperative	0.441	0.497	0.375	0.277	0.256	0.578	0.500
15	Lenses	0.800	0.565	0.851	0.889	0.500	0.344	0.668
	Average	0.735	0.707	0.797	0.742	0.625	0.588	0.752

classes in lenses. Thus, for one sample, the relevant degree of this sample and all the hyperedges are always high and higher than the sample's neighborhood threshold  $\delta$  that makes many hyperedges which have negative effect on classification in  $\delta_B(x)$ . And the classification for the sample is the voting result of the hyperedges in  $\delta_B(x)$ ; as a result, the error rate increases.

## 5. Conclusions

The processing of incomplete information system is rather complex. It is hard to process the incomplete information system by the traditional hypergraph, due to two reasons: (1) the hyperedges are generated randomly; (2) there are many missing values in incomplete information system. In this paper, we propose a new algorithm based on hypergraph and

rough set theory to solve the problem of classifying incomplete data set. The rough set theory can be used to supervise the classifying and replacing of hyperedges. The experimental results on 15 UCI data sets show that the classification result of the proposed algorithm improves in comparison to another six algorithms. However, the algorithm HyperGraph will cost much time, due to calculating the relevant degree between hyperedge and sample. Thus, how to reduce the running time of the algorithm is our future work.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

TABLE 7: The accuracy of data sets whether missing values exist or not (in: incomplete).

Number	Data set	C4.5	SVM	NaiveBayes	KNN	Rough	HyperGraph
1	Blood	0.778	0.762	0.754	0.749	0.480	0.772
2	(in)Blood	0.775	0.762	0.751	0.721	0.600	0.765
3	Balance	0.645	0.902	0.914	0.838	0.698	0.860
4	(in)Balance	0.653	0.878	0.890	0.819	0.635	0.861
5	ILPD	0.684	0.714	0.557	0.645	0.356	0.710
6	(in)ILPD	0.703	0.714	0.559	0.638	0.310	0.705
7	Haberman	0.719	0.735	0.748	0.703	0.387	0.751
8	(in)Haberman	0.722	0.735	0.748	0.680	0.548	0.752
9	Statlog Heart	0.767	0.830	0.848	0.811	0.111	0.852
10	(in)Statlog Heart	0.762	0.810	0.829	0.795	0.259	0.843
11	Spect Heart	0.708	0.727	0.685	0.704	0.704	0.721
12	(in)Spect Heart	0.809	0.824	0.794	0.764	0.556	0.841
13	Bankruptcy	0.980	0.996	0.992	0.996	0.996	0.996
14	(in)Bankruptcy	0.972	0.988	0.992	0.996	0.998	0.996
15	Sonar	0.712	0.760	0.678	0.861	0.619	0.814
16	(in)Sonar	0.755	0.764	0.678	0.760	0.429	0.792
17	Iris	0.960	0.960	0.960	0.953	0.667	0.953
18	(in)Iris	0.913	0.927	0.953	0.880	0.933	0.960
19	Fertility	0.870	0.880	0.880	0.820	0.800	0.880
20	(in)Fertility	0.880	0.880	0.880	0.820	0.800	0.880
21	Lenses	0.833	0.708	0.708	0.792	0.667	0.750
22	(in)Lenses	0.750	0.583	0.667	0.667	0.667	0.733

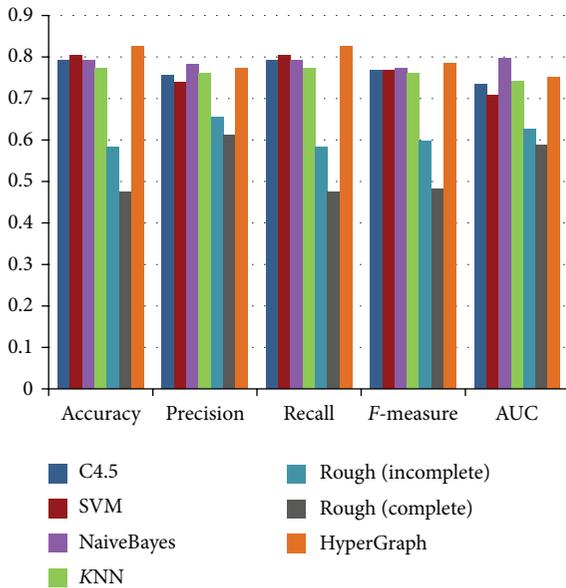


FIGURE 8: The average value of each indicator.

**Acknowledgments**

This project is supported by the National Natural Science Foundation of China (NSFC) under Grant nos. 61309014, 61379114, and 61472056 and Natural Science Foundation Project of CQ CSTC under Grant no. cstc2013jcyjA40063.

**References**

- [1] G. Boolos and G. Sambin, "An incomplete system of modal logic," *Journal of Philosophical Logic*, vol. 14, no. 4, pp. 351–358, 1985.
- [2] P. Clark and T. Niblett, "The CN2 induction algorithm," *Machine Learning*, vol. 3, no. 4, pp. 261–283, 1989.
- [3] G. Y. Wang, "The extension of rough set in incomplete information system," *Frontiers of Electrical and Electronic Engineering in China*, vol. 3, no. 4, pp. 399–405, 2008.
- [4] C. I. Faustino Agreira, M. M. Travassos Valdez, C. M. Machado Ferreira, and F. P. Maciel Barbosa, "Analysis of an incomplete information system using the rough set theory," in *Mathematical Methods in Engineering*, pp. 193–203, Springer, Amsterdam, The Netherlands, 2014.
- [5] Y. Hou, "Rough set extension model of incomplete information system," *TELKOMNIKA Indonesian Journal of Electrical Engineering*, vol. 12, no. 4, pp. 2843–2849, 2014.
- [6] L. Guan and G. Y. Wang, "Generalized approximations defined by non-equivalence relations," *Information Sciences*, vol. 193, pp. 163–179, 2012.
- [7] Y. Sheffi, *Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods*, Prentice-Hall, 1985.
- [8] J. L. Segovia-Juarez, S. Colombano, and D. Kirschner, "Identifying DNA splice sites using hypernetworks with artificial molecular evolution," *BioSystems*, vol. 87, no. 2-3, pp. 117–124, 2007.
- [9] K.-Y. Wang, B.-S. Kim, C.-H. Lee, J. Park, and D.-J. Chung, "CVD level prediction processor using DNA computing," *IEICE Electronics Express*, vol. 6, no. 9, pp. 529–534, 2009.

- [10] J.-K. Kim and B.-T. Zhang, "Evolving hypernetworks for pattern classification," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '07)*, pp. 1856–1862, September 2007.
- [11] Z. Pawlak, "Rough sets," *International Journal of Computer & Information Sciences*, vol. 11, no. 5, pp. 341–356, 1982.
- [12] Z. Pawlak and A. Skowron, "Rudiments of rough sets," *Information Sciences*, vol. 177, no. 1, pp. 3–27, 2007.
- [13] Z. Pawlak, "Rough set approach to knowledge-based decision support," *European Journal of Operational Research*, vol. 99, no. 1, pp. 48–57, 1997.
- [14] T. Y. Lin, "Granular computing on binary relations I: data mining and neighborhood systems," *Rough Sets in Knowledge Discovery*, vol. 1, pp. 286–318, 1998.
- [15] Q. Hu, D. Yu, and Z. Xie, "Neighborhood classifiers," *Expert Systems with Applications*, vol. 34, no. 2, pp. 227–239, 2006.
- [16] C. Berge and E. Minieka, *Graphs and Hypergraphs*, American Elsevier, 1973.
- [17] B. Huang and X.-Z. Zhou, "Extension of rough set model based on connection degree under in complete information systems," *System Engineering Theory and Practice*, vol. 24, no. 1, pp. 88–92, 2004.
- [18] F. Hu, X. Liu, J. Dai, and H. Yu, "A novel algorithm for imbalance data classification based on neighborhood hypergraph," *The Scientific World Journal*, vol. 2014, Article ID 876875, 13 pages, 2014.
- [19] J. Wang, L. Jin, and K. Sun, "Chinese text categorization based on evolutionary hypernetwork," *Journal of Jiangsu University: Natural Science Edition*, vol. 34, no. 2, pp. 196–201, 2013 (Chinese).
- [20] Y. S. Jing, V. Pavlović, and J. M. Rehg, "Boosted Bayesian network classifiers," *Machine Learning*, vol. 73, no. 2, pp. 155–184, 2008.
- [21] F. Hu and H. Li, "A novel boundary oversampling algorithm based on neighborhood rough set model: NRSBoundary-SMOTE," *Mathematical Problems in Engineering*, vol. 2013, Article ID 694809, 10 pages, 2013.
- [22] S. Wang, D. Li, N. Petrick, B. Sahiner, M. G. Linguraru, and R. M. Summers, "Optimizing area under the ROC curve using semi-supervised learning," *Pattern Recognition*, vol. 48, no. 1, pp. 276–287, 2015.
- [23] J. Huang and C. X. Ling, "Using AUC and accuracy in evaluating learning algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 3, pp. 299–310, 2005.
- [24] X. Wu, V. P. Kumar, J. R. Quinlan et al., "Top 10 algorithms in data mining," *Knowledge and Information Systems*, vol. 14, no. 1, pp. 1–37, 2008.
- [25] G. Holmes, A. Donkin, and I. H. Witten, "WEKA: a machine learning workbench," in *Proceedings of the 2nd Australian and New Zealand Conference on Intelligent Information Systems*, pp. 357–361, IEEE, Brisbane, Australia, December 1994.
- [26] G.-Y. Wang, Z. Zheng, and Y. Zhang, "RIDAS—a rough set based intelligent data analysis system," in *Proceedings of the International Conference on Machine Learning and Cybernetics*, vol. 2, pp. 646–649, IEEE, 2002.
- [27] J. G. Gao and Y. Q. Cui, "Continuous attribute discretization method based on comentropy theory," *Microelectronics & Computer*, vol. 28, no. 7, pp. 187–194, 2011 (Chinese).
- [28] F. C. Liu, "A set pair rough set model based on the limited tolerance relation," *Computer Science*, vol. 32, no. 6, pp. 124–128, 2005 (Chinese).
- [29] G. Y. Wang, *Rough Set Theory and Knowledge Acquisition*, Xi'an Jiaotong University, Xi'an, China, 2001, (Chinese).
- [30] L. Y. Chang, G. Y. Wang, and Y. Wu, "An approach for attribute reduction and rule generation based on rough set theory," *Journal of Software*, vol. 10, no. 11, pp. 1206–1211, 1999 (Chinese).
- [31] B. Han and B. R. Yang, "A simplified algorithm attribute and attribute value reduction," *Journal of Chinese Computer Systems*, vol. 2, pp. 245–247, 2004 (Chinese).
- [32] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI '95)*, vol. 2, pp. 1137–1143, 1995.
- [33] G. I. Webb and M. J. Pazzani, "Adjusted probability naive Bayesian induction," in *Proceedings of the Australian Joint Conference on Artificial Intelligence (AI '98)*, vol. 1502, pp. 285–295, Brisbane, Australia, July 1998.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

