A comparison study of similarity measures for covering-based neighborhood classifiers

Fu-Lun Liu[a], Ben-Wen Zhang[b], Davide Ciucci[c], Wei-Zhi Wu[d,e], Fan Min[a]

[a] *School of Computer Science, Southwest Petroleum University, Chengdu, Sichuan 610500, PR China*
[b] *Department of Computer Science, Sichuan University for Nationalities, Kangding, Sichuan 626001, China*
[c] *DISCo, University of Milano-Bicocca, viale Sarca 336/14, 20126 Milano, Italy*
[d] *School of Mathematics, Physics and Information Science, Zhejiang Ocean University, Zhoushan, Zhejiang 316022, PR China*
[e] *Key Laboratory of Oceanographic Big Data Mining & Application of Zhejiang Province, Zhejiang Ocean University, Zhoushan, Zhejiang 316022, PR China*

Abstract

In data mining, neighborhood classifiers are valid not only for numeric data but also symbolic data. The key issue for a neighborhood classifier is how to measure the similarity between two instances. In this paper, we compare six similarity measures, *Overlap*, *Eskin*, occurrence frequency (*OF*), inverse OF (*IOF*), *Goodall3*, and *Goodall4*, for symbolic data under the framework of a covering-based neighborhood classifier. In the training stage, a covering of the universe is built based on the given similarity measure. Then a covering reduction algorithm is used to remove some of these covering blocks and determine the representatives. In the testing stage, the similarities between all unlabeled instances and representatives are computed. The closest representative or a few representatives determine the predicted class label of the unlabeled instance. We compared the six similarity measures in experiments on 15 University of California-Irvine (UCI) datasets. The results demonstrate that although no measure dominated the others in all scenarios, some measures had consistently high performance. The covering-based neighborhood classifier with appropriate similarity measures, such as *Overlap*, *IOF*, and *OF*, was better than ID3, C4.5, and the Naïve Bayes classifiers.

Keywords: Classifier, covering-based rough set, representative, similarity measure.

1. Introduction

Covering-based rough sets [52] and neighborhood rough sets [50] are closely related methodologies for knowledge representation [8, 25, 35, 55, 59], reduction

*Corresponding author. Tel.: +86 135 4068 5200. Email: minfanphd@163.com.

[20, 41, 46, 51], classification [19, 22], and clustering [29, 34]. For knowledge representation, Zhu [59] proposed topological-based covering construction, Qian et al. [35] defined multi-granulation rough sets, Zhang et al. [55] presented composite rough sets, and Chen et al. [8] proposed a rough set model for hybrid data. For knowledge reduction, Yao et al. [51] classified all approximation operators into three types, Wang et al. [46] defined two types of characteristic matrices of coverings, Wang et al. [41] proposed a matrix-based method, and Lang et al. [20] proposed a varying covering cardinalities method for dynamic decision information systems. For classification, Li et al. [22] compared multi-granulation rough sets and concept lattices through rule acquisition, Kumar et al. [19] combined them with a support vector machine (SVM) and neural network. For clustering, Prabhavathy et al. [34] proposed an approach based on coverings instead of partitions. These theories have been combined with other theories, such as fuzzy sets [26, 43], lattice theory [22, 40], and evidence theory [7] to provide helpful results.

Covering-based neighborhood classifiers [1, 16, 23, 33, 44, 45, 48] have been particularly successful in this field. Hu et al. [16] integrated an attribute reduction technique with classification learning under a uniform framework of a neighborhood rough set model. Wang et al. [45] proposed an approach based on the Kruskal–Wallis rank sum test and neighborhood rough set model for gene reduction. Wang et al. [44] presented a subclass-weighted neighborhood classifier for class imbalanced data. Yao et al. [33] constructed a hybrid SVM-based credit scoring model through a neighborhood rough set. Usiobaifo et al. [1] proposed a diabetes diagnosis model using a rough set and k-nearest neighbor (kNN) classifier algorithm. Li et al. [23] proposed a neighborhood-based decision-theoretic rough set model to process numerical data with noise. Xu et al. [48] proposed an effective collaborative representation-based classification algorithm.

Recently, Zhang et al. [53] built a representative-based classification (RC) algorithm for symbolic data by combining model-based and instance-based approaches. In the training stage, the neighborhood of each labeled instance is constructed. Unlike Hu et al.'s [15] approach, which only deals with numerical data, the similarity measure is also valid for symbolic data. These instances of which neighborhoods are retained after the blocks' reduction are considered as representatives. In the testing stage, the distances between unlabeled instances and representatives are computed. The closest representative or a few representatives determine the prediction of the class label of the unlabeled instance. Compared with instance-based approaches, the new algorithm can store the classification model (representatives) and reduce the computational complexity. Compared with model-based approaches, the new algorithm can classify any unlabeled instance that the existing model could not directly classify; that is, this algorithm balances classification accuracy and computational capacity. Clearly, the similarity measure is essential for the performance of the RC algorithm.

In this paper, we compare six similarity measures under the RC algorithm framework: *Overlap* [30, 39], *Eskin* [10], *Goodall3* [12], *Goodall4* [12], occurrence frequency (OF) [17], and inverse OF (IOF) [17]. As the simplest measure,

Overlap [39] computes the similarity between two instances by counting the number of attributes for which they match. Eskin et al. [10] further considered the number of values taken by an attribute and proposed the *Eskin* measure. If two instances mismatch on an attribute that takes many values, the *Eskin* measure provides a higher sub-similarity compared with an attribute that takes only a few values. The *Goodall3* measure provides a high sub-similarity to a match if the value is less frequent. As a variant, *Goddall4* [6] provides a high sub-similarity to a match if the value is more frequent. *IOF* and *OF* are a pair of complementary measures. The *IOF* measure assigns a low sub-similarity to mismatches on more frequent values, whereas the *OF* measure assigns the opposite sub-similarity. These six measures were chosen for the control experiment because they are representative of the entire set given in [6]. Some of the other similarities, such as *Goodall1*, *Goodall2*, *Smirnov*, and *Burnaby*, were not compared because they required another subtask for attribute selection [56].

We selected 15 datasets from the University of California-Irvine (UCI) Repository of Machine Learning Databases [2]: Zoo, Promoters, Iris, Wine, Sonar, Ionosphere, Dermatology, Voting, WDBC, Tic-Tac-Toe, Car, Kr-vs-kp, Waveform, Mushroom, and Penbased. The experimental results indicated that, while no one measure outperformed the others for all datasets, the RC algorithm with an appropriate similarity measure had consistently high performance. Among the six measures, *Overlap*, *IOF*, and *OF* were better than the others. If the training set was sufficiently large, the *Overlap* measure typically had a significant advantage over the *IOF* and *OF* measures. The *Eskin* measure made the RC algorithm obtain the highest accuracies on some datasets; however, its performance fell into the medium level. Exceptions were the *Goodall3* and *Goodall4* measures, which never achieved the best classification accuracies on these 15 datasets. By contrast, some datasets were not severely affected by the similarity measures, such as Zoo, Wine, WDBC, and Penbased. Additionally, the RC classifier with the *Overlap* measure even outperformed ID3 [36], C4.5 [37], and the Naïve Bayes [38] classifiers.

The remainder of this paper is organized as follows: In Section 2, we introduce some preliminary information and redefine the neighborhoods. In Section 3, we discuss a representative generation algorithm and RC algorithm. In Section 4, we conduct experiments to compare the classification precisions of the RC algorithm with different similarity measures. Finally, in Section 5, we state some conclusions from this research, and suggest further study ideas.

2. Preliminaries

In this section, we provide definitions of decision systems, indiscernibility relations, similarity relations, neighborhoods, and covering. Moreover, we recall the notion of the minimum threshold [53] that will be used to select representative elements in the RC algorithm.

Table 1: Decision system.

U	Sgpt	Gammagt	Alkphos	Hepatitis
x_1	low	low	low	yes
x_2	normal	high	middle	no
x_3	high	low	low	no
x_4	low	low	middle	yes

2.1. Decision system

The concept of a decision system is widely used in data mining [9, 47] and machine learning [2, 18, 31, 47]. Decision systems are fundamental for the classification of rough sets.

Definition 1. A decision system S is a five-tuple:

$$S = (U, C, d, V = \{V_a \mid a \in C \cup \{d\}\}, I = \{I_a \mid a \in C \cup \{d\}\}), \tag{1}$$

where

1. U is a nonempty finite set of instances called the universe;
2. C is a nonempty finite set of conditional attributes;
3. d is the decision attribute;
4. V_a is the set of values for each $a \in C \cup \{d\}$; and
5. I_a: $U \to V_a$ is an information function for each $a \in C \cup \{d\}$.

In this paper, we consider decision systems where all attributes are nominal. Table 1 lists a nominal decision system, where $U = \{x_1, x_2, x_3, x_4\}$, $C = \{$Sgpt, Gammagt, Alkphos$\}$, $d =$ Hepatitis, $V_{Sgpt} = \{$low, normal, high$\}$, $V_{Gammagt} = \{$low, normal, high$\}$, $V_{Alkphos} = \{$low, middle, high$\}$ and $V_{Hepatitis} = \{$yes, no$\}$.

2.2. Indiscernibility and similarity relations

Rough set theory is based on the notion of indiscernibility among objects which, based on conditional attributes, divides instances into classes with the same characteristics. Formally:

Definition 2. [32] Let S be a decision system and $A \subseteq C$. The indiscernibility relation induced by A is

$$IND(A) = \{(x, y) \in U \times U | \forall a \in A, I_a(x) = I_a(y)\}. \tag{2}$$

Clearly, $IND(A)$ is an equivalence relation that partitions universe U into equivalence classes: $[x]_A = \{y \in U | (x, y) \in IND(A)\}$. The lower and upper approximations of $X \subseteq U$ are defined as

$$l_A(X) = \{x \in U | [x]_A \subseteq X\}, \tag{3}$$

$$u_A(X) = \{x \in U | [x]_A \cap X \neq \emptyset\}. \tag{4}$$

The lower approximation $l_A(X)$ represents the elements that *definitely* belong to X up to the available knowledge given by an attributes subset A, whereas the upper approximation u_A, contains those objects that *possibly* belong to X, given knowledge A.

Now, let $U/\{d\} = \{X_1, X_2, \ldots, X_{|V_d|}\}$. That is, the universe is partitioned into $|V_d|$ decision classes. If two instances have the same conditional attribute values, that is, are indiscernible with respect to C, but different decisions, we call them *contradictory instances*. Decision systems with contradictory instances are *inconsistent*. For simplicity, in this work, we do not consider such a scenario. Thus, contradictory instances from U will be removed to form a new universe: the so-called *positive region*, formally defined as

$$POS_C(d) = \bigcup_{X_i \in U/\{d\}} l_C(X_i). \tag{5}$$

We observe that two instances in U fall into $IND(C)$ if and only if they have the same values on *all* attributes in C, according to Definition 2.

2.3. Similarity measures for nominal data

The indiscernibility relation is qualitative and does not comply with more relaxed requirements than the equality of attributes. Thus, a similarity relation is often used to describe the indiscernibility of objects and cluster them. Regarding nominal attributes, several similarity measures have been defined in the past several years [6, 13], with the *Overlap* measure being the most commonly used. Its popularity is perhaps related to its simplicity and realizability. In this section, we introduce several other similarity measures for nominal data.

In most cases, similarity measures assign a similarity value to a pair of instances x and y that belong to the decision system S as follows:

Definition 3. Let $S = (U, C, d, V = \{V_a \mid a \in C \cup \{d\}\}, I = \{I_a \mid a \in C \cup \{d\}\})$ be a nominal decision system. The similarity between $x, y \in U$ with respect to $\emptyset \subset A \subseteq C$ is

$$sim(x, y, A) = \frac{1}{|A|} \sum_{a \in A} subsim_a(I_a(x), I_a(y)), \tag{6}$$

where $subsim_a(I_a(x), I_a(y))$ is the per-attribute similarity between two values of nominal attribute a.

In the following context we also let $sim(x, y) = sim(x, y, C)$. For brevity and unambiguity, some notations will be redefined. Moreover, we need to identify the characteristics of a nominal dataset. We enumerate the notation and its characteristics for a nominal dataset as follows:

- The *size of data* N is used to represent the number of instances contained in the decision system, that is, $N = |U|$. As one of the most important pieces of information, N is used in all six measures.

- The *distribution* of $f(v)$ is defined as the number of times attribute a takes value $I_a(x) = v$ in decision system S. Note that if $v \notin V_a$, then $f(v) = 0$; that is, it refers to the distribution of the frequency of values. This function is significant for some measures that are sensitive to frequently occurring attribute values.

- The *distribution* of $p^2(v)$ is a function that represents another probability estimate of attribute a to take value $I_a(x) = v$ in S and is given by

$$p^2(v) = \frac{f(v)(f(v)-1)}{N(N-1)}. \tag{7}$$

We provide the formulas for the similarity measures used in this paper.

1. *Overlap*:

$$subsim_a(I_a(x), I_a(y)) = \begin{cases} 1, & \text{if } I_a(x) = I_a(y); \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

   The *Overlap* [30, 39] measure is one of most widely used measures because of its simplicity. It only counts the number of attributes that match in the two instances. The range of per-attribute similarities for the *Overlap* measure is $\{0, 1\}$, with a value of 0 occurring when there is no match and a value of 1 occurring when the attribute values match.
2. *Eskin*:

$$subsim_a(I_a(x), I_a(y)) = \begin{cases} 1, & \text{if } I_a(x) = I_a(y) \\ \frac{|V_a|^2}{|V_a|^2+2}, & \text{otherwise.} \end{cases} \tag{9}$$

   Eskin et al. [10] considered the difficulty of matching in an attribute that takes many values. The *Eskin* measure still provides similarity values when this mismatch occurs; that is, this measure provides more weight to mismatches that occur on attributes that take many values. The range of per-attribute similarity for mismatches in the *Eskin* measure is $[\frac{1}{3}, \frac{N^2}{N^2+2}]$, with the minimum value attained when attribute a takes only two values.
3. *IOF*:

$$subsim_a(I_a(x), I_a(y)) = \begin{cases} 1, & \text{if } I_a(x) = I_a(y); \\ \frac{1}{1+\log f(I_a(x)) \times \log f(I_a(y))}, & \text{otherwise.} \end{cases} \tag{10}$$

   The *IOF* [17] measure assigns a lower similarity to mismatches on more frequent values. The range of $subsim_a(I_a(x), I_a(y))$ for mismatches in the *IOF* measure is $[\frac{1}{1+(\log \frac{N}{2})^2}, 1]$. When a mismatch occurs, $V_a = 2$, $I_a(x)$ and $I_a(y)$ each occur $\frac{N}{2}$ times, and there is a minimum *IOF* similarity value; if $I_a(x)$ and $I_a(y)$ occur only once, then there is a maximum *IOF* similarity value.

4. *OF*:

$$subsim_a(I_a(x), I_a(y)) = \begin{cases} 1, & \text{if } I_a(x) = I_a(y); \\ \frac{1}{1+\log \frac{N}{f(I_a(x))} \times \log \frac{N}{f(I_a(y))}}, & \text{otherwise.} \end{cases} \tag{11}$$

The *OF* [17] measure provides the opposite weighting of the *IOF* measure for mismatches. The range of $subsim_a(I_a(x), I_a(y))$ for mismatches in the *OF* measure is $[\frac{1}{1+(\log N)^2}, \frac{1}{1+(\log 2)^2}]$, with the minimum value attained when $I_a(x)$ and $I_a(y)$ occur only once in the dataset, and the maximum value attained when $I_a(x)$ and $I_a(y)$ occur $\frac{N}{2}$ times.

5. *Goodall3*:

$$subsim_a(I_a(x), I_a(y)) = \begin{cases} 1 - p^2(I_a(x)), & \text{if } I_a(x) = I_a(y); \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

The *Goodall3* measure is the same as the *Goodall* measure [12] on a per-attribute basis. The main difference is that the *Goodall3* measure takes the average of the per-attribute similarities instead of combining the similarities by taking into account dependencies between attributes. The *Goodall3* measure assigns a higher similarity to a match if the value is infrequent than if the value is frequent. The range of $subsim_a(I_a(x), I_a(y))$ for matches in the *Goodall3* measure is $[0, 1 - \frac{2}{N(N-1)}]$, with the minimum value attained if $I_a(x)$ occurs only once and the maximum value attained if $I_a(x)$ is the only value of V_a.

6. *Goodall4*:

$$subsim_a(I_a(x), I_a(y)) = \begin{cases} p^2(I_a(x)), & \text{if } I_a(x) = I_a(y); \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

The *Goodall4* measure is a variant of the *Goodall* measure. The *Goodall3* and *Goodall4* measures are complementary; thus, the *Goodall4* measure assigns similarity 1 - *Goodall3* for matches. The range of $subsim_a(I_a(x), I_a(y))$ for matches in the *Goodall4* measure is $[\frac{2}{N(N-1)}, 1]$, with the minimum value attained if $I_a(x)$ occurs only once, and the maximum value attained if $I_a(x)$ is the only value of V_a.

These similarity measures focus on different characteristics of the nominal dataset. A pair of instances have obviously different per-attribute similarity values for different measures. For example, Table 1 lists a decision system, and the similarity values computed by different measures of each pair of instances are listed in Table 2. It is clear that the similarity measures influence the similarity values. The similarity relation is the first step toward (at least) two generalizations of rough sets: relation and covering-based.

2.4. Neighborhood and covering rough sets

We now briefly introduce some concepts for covering rough sets. First, we recall that a covering is a weaker notion of a partition that relinquishes the requirement that the classes are disjoint.

Table 2: Similarity matrix.

	Overlap	*Eskin*	*Goodall3*	*Goodall4*	*IOF*	*OF*
(x_1, x_1)	1.0000	1.0000	0.8888	0.1111	1.0000	1.0000
(x_1, x_2)	0.0000	0.8181	0.0000	0.0000	0.8918	0.5092
(x_1, x_3)	0.3333	0.8787	0.2777	0.0555	1.0000	0.6733
(x_1, x_4)	0.3333	0.8787	0.2777	0.0555	0.8918	0.7284
(x_2, x_2)	1.0000	1.0000	0.9444	0.0555	1.0000	1.0000
(x_2, x_3)	0.0000	0.8181	0.0000	0.0000	0.8918	0.5092
(x_2, x_4)	0.3333	0.8787	0.2777	0.1111	1.0000	0.6733
(x_3, x_3)	1.0000	1.0000	0.8888	0.0555	1.0000	1.0000
(x_3, x_4)	0.3333	0.8787	0.2777	0.0555	0.8918	0.7284
(x_4, x_4)	1.0000	1.0000	0.8333	0.1666	1.0000	1.0000

Definition 4. Given a universe U, a *covering* of U is a collection of sets $C_i \subseteq \mathcal{P}(U)$ such that $\cup C_i = U$, where $\mathcal{P}(U)$ is the power set of U.

A possible way to obtain a covering is through a similarity (instead of equivalence) relation that clusters objects in similarity (instead of equivalence) classes.

Definition 5. The *neighborhood* of $x \in U$ with respect to similarity measure $sim : U \times U \mapsto \mathcal{R}$ and threshold $\theta \in \mathcal{R}$ is

$$N(x, \theta) = \{y \in U | sim(x, y) \geq \theta\}. \tag{14}$$

Clearly, the size of $N(x, \theta)$ is inversely proportional to threshold θ. In the extreme case of θ equal to zero, all the instances are neighbors of x. Typically, threshold θ is fixed a priori by the user. We are interested in automatically assigning this value by selecting the minimum possible value of θ.

Definition 6. Let $S = (U, C, d, V, I)$ be a nominal decision system and $U/\{d\} = \{X_1, X_2, \ldots, X_{|V_d|}\}$. The minimum possible threshold value for $x \in X_i$ is

$$\theta_x^* = \min\{\theta | N(x, \theta) \subseteq X_i\}. \tag{15}$$

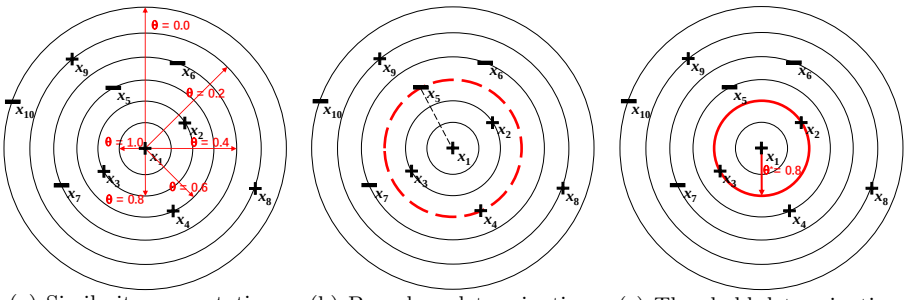
Note that θ_x^* is not specified by the user; instead, it is determined by decision system S and instance x. This determines, in turn, the maximum possible neighborhood of x.

Definition 7. The maximum neighborhood of $x \in U$ is

$$N^*(x) = N(x, \theta_x^*). \tag{16}$$

Fig. 1 illustrates the determination of $\theta_{x_1}^*$ with the *Overlap* measure similarity, where '+' and '−' represent two class labels. The training set is $\{x_1, x_2, \ldots, x_{10}\}$, there are five conditional attributes, and the set of candidate similarity values is {1.0, 0.8, 0.6, 0.4 ,0.2, 0.0}. Fig. 1(a) shows the similarities between x_1 and

all other labeled instances. Fig. 1(b) shows that x_5 is the most similar instance of x_1 with a different class label. We call x_5 a boundary instance of x_1. The boundary similarity of x_1 is $sim_b(x_1) = sim(x_1, x_5) = 0.6$. Fig. 1(c) shows the actual threshold $\theta^*_{x_1} = sim_b(x_1) + 1/|C| = 0.8$; that is, instances with a higher than or equal similarity to $\theta^*_{x_1}$ have the same class labels as x_1.



(a) Similarity computation (b) Boundary determination (c) Threshold determination

Figure 1: Illustration of the threshold determination for x_1.

Finally, it may occur that in a given covering, block C_i is the union of some other block or the subset of another block. In some sense, we thus have redundant information, which is a scenario that we may wish to avoid in applications. Indeed, several notions of covering *reduction* have been proposed in the literature [49]. We are interested in obtaining a so-called *genuine* covering [4, 5] by maximizing the size of the neighborhoods and thus eliminating small (redundant) classes.

Definition 8. Let $\mathcal{C} = \{C_i\}$ be a covering of set U. Set $C_i \in \mathcal{C}$ is *redundant* if it is a subset of another set $C_j \in \mathcal{C}$. A *reduct* of $\mathcal{C}$ is the covering obtained by eliminating redundant sets from $\mathcal{C}$.

3. Algorithm design

In this section, we follow the framework proposed in [53] to design our algorithms for different similarity measures. All algorithms have two subroutines: representative generation and RC. These subroutines are executed successively.

3.1. Representative generation

In this subsection, we describe the representative generation algorithm. There are two stages:

1. Neighborhood construction stage. Based on Definition 7, the maximum neighborhood of each instance is constructed.
2. Representative selection and redundancy removal stage. A greedy approach is designed to select a set of representatives whose neighborhoods cover the training set.

Algorithm 1 Representative generation RG

Input: Decision system $S = (U, C, d, I, V)$.
Output: Representative instances set Y and covering $CR = \{(x, \theta_x^*)|x \in Y\}$.
Constraint: $Y \subseteq U$ and $\bigcup CR = POS_C(d)$.
Optimization objective: Minimize $|Y|$.

1: $Y = \emptyset$, $CR = \emptyset$;
2: Compute V_a, where $a \in C$;
3: Compute $f(v)$, where $v \in V_a$;
4: Compute $sim(x, y)$, where $(x, y) \in U \times U$;
5: **for** (each $x \in U$) **do**
6: $\theta_x^* = 0$;
7: **for** (each $x \in U$) **do**
8: **for** (each $y \in U$) **do**
9: **if** $((d(x) \neq d(y)) \wedge sim(x, y) \geq \theta_x^*)$ **then**
10: $\theta_x^* = sim(x, y)$;
11: **end if**
12: **end for**
13: **end for**
14: **end for**
15: **for** (each $x \in U$) **do**
16: $N^*(x) = \emptyset$;
17: **for** $(y \in U)$ **do**
18: **if** $(sim(x, y) > \theta_x^*)$ **then**
19: $N^*(x) = \cup\{y\}$;
20: **end if**
21: **end for**
22: **end for**
23: Compute $U/\{d\} = \{X_1, X_2, \ldots, X_{|V_d|}\}$;
24: **for** $(i = 1$ to $|V_d|)$ **do**
25: $X = X_i$;
26: **while** $X \neq \emptyset$ **do**
27: Select $x \in U \cap X_i$ st. $|N^*(x) \cap X|$ is maximum;
28: $Y_i = Y_i \cup \{x\}$;
29: $X = X - N^*(x)$;
30: **end while**
31: **end for**
32: $CR = \{(x, \theta_x^*)|x \in Y\}$;
33: Return Y and CR.

Algorithm 1 lists the representative generation algorithm. There are four steps that correspond to data preprocessing, threshold computation, neighborhood computation, and representative selection.

Lines 1–4 show the data preprocessing step. Different similarity measures need relevant information about the dataset. We compute the number of values $|V_a|$ of each conditional attribute $a \in A$ for the *Eskin* measure. Distribution $f(v)$ of each value is computed for the *Goodall3*, *Goodall4*, *IOF*, and *OF* measures. Then, the similarity of each pair of instances can be computed.

Lines 5–14 show the process of threshold computation. For each instance x, we determine the most similar instance y with a different class label, and $sim(x, y)$ is the boundary similarity. The candidate similarity value nearest to and greater than $sim(x, y)$ is assigned to threshold θ_x^*. This threshold ensures that the similar instances that have greater similarity values to x have the same class labels. This process is illustrated in Fig. 1.

Lines 15–22 show the process of neighborhood computation. When the threshold of instance x is computed, the neighborhood of x is also determined. Any instance whose similarity to x is greater than θ_x^* is a neighbor of x, and all these neighbors constitute the neighborhood.

Lines 23–31 show the process of representative selection. With the neighborhood of each instance determined, the next issue is to select the representatives from these labeled instances. To minimize the number of neighborhoods, we adopt a greedy-based strategy to remove some useless instances. Line 23 indicates that U is divided into several positive areas $POS_C(d)$ using the decision attribute. As Fig. 2 shows, X is the current positive area, which is composed of several covering blocks (neighborhoods) A–Z, and expressed as a 5×5 grid. Among these covering blocks, block A (3×3), block B (3×2), and block C (2×2) are the three largest blocks. At the beginning, X is empty, and block A can cover most grids compared with other blocks. When block A covers X, block B is the largest block. However, block C is selected to cover the X second instead of block B because block C can cover four uncovered grids and block B can cover only three uncovered grids; that is, the number of newly covered grids determines which block is selected to cover the positive areas $POS_C(d)$ instead of the block size. Lines 26–30 correspond to this greedy covering process. In this manner, we can always select the most-covered block, and this operation does not stop until these positive areas are completely covered. Blocks that involve covering positive areas are retained and other blocks are removed. Thus, the core instances of these blocks are marked as representatives.

Proposition 1. *The computational complexity of Algorithm 1 is $O(n^2(p+m))$, where n is the number of instances, m is the number of conditional attributes, and p is the number of representatives.*

Proof. The time complexity is determined for the four components of Algorithm 1.

1. Algorithm 1 with different similarity measures has different time complexities. For the *Overlap* measure, the algorithm does not need extra statistical information. For the *Eskin* measure, the algorithm needs the number of values of each attribute, but this information is provided by the dataset. However, the other four measures need distribution $f(v)$, which

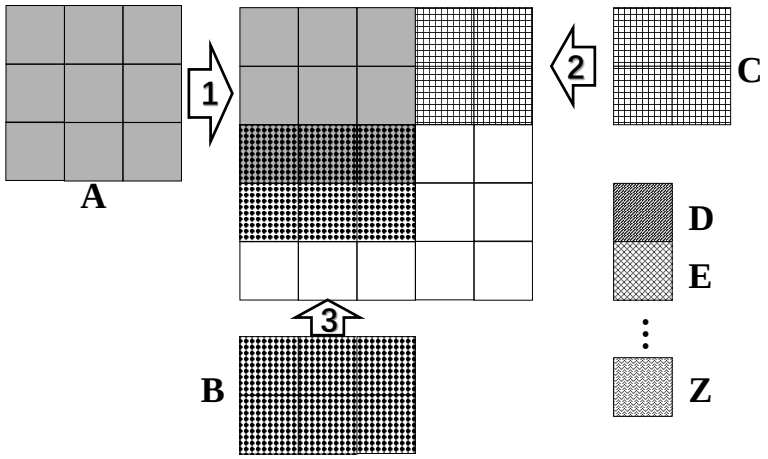


Figure 2: Illustration of the greedy selection strategy.

Table 3: Computational complexity of Lines 1–4.

Measure	Computation times	Computational complexity	Note
Overlap	$\frac{nm(n-1)}{2}$	$O(n^2m)$	–
Eskin	$\frac{nm(n-1)}{2}$	$O(n^2m)$	number of values
Goodall3	$\frac{nm(n+1)}{2}$	$O(n^2m)$	frequency of occurrence
Goodall4	$\frac{nm(n+1)}{2}$	$O(n^2m)$	frequency of occurrence
IOF	$\frac{nm(n+1)}{2}$	$O(n^2m)$	frequency of occurrence
OF	$\frac{nm(n+1)}{2}$	$O(n^2m)$	frequency of occurrence

is contained in the dataset, and we need to spend more time computing this statistical information. The similarities of each pair of instances are computed, that is, every instance needs to be compared with the other $(n-1)$ instances on each attribute. It is noteworthy that these measures are symmetric, that is, $sim(x,y) = sim(y,x)$. Moreover, the time complexity for $sim(x,y)$ computation is $O(\frac{nm(n-1)}{2})$ for all these measures. To obtain the information of distribution $f(v)$, we need to traverse universe U and add $n \times m$ computations. Thus, the *Goodall3*, *Goodall4*, *IOF*, and *OF* measures need more than $n \times m$ times computations than the *Overlap* and *Eskin* measures. Despite that, $O(n^2m)$ is the computational complexity for all these similarity measures. Table 3 lists the aforementioned information.

2. As described in Algorithm 1 lines 5–14, the most similar but contradictory instance determines threshold θ^*. When the similarity matrix has been computed, we determine threshold θ^* using a sequential search approach; that is, there are n similarity values that need to be compared for each instance. Hence, the time complexity is $O(n^2)$ in this step, regardless of

which similarity measure is adopted.

3. Third, as described in Algorithm 1 lines 15–22, any instance y belongs to $N^*(x)$ if $sim(x, y) > \theta_x^*$. The similarities between instance x and the other instances should be compared with θ_x^*. When the neighborhood of each instance is determined, there are $n \times (n-1)$ times calculations that should be performed, and the time complexity is $O(n^2)$.
4. A greedy strategy is used to select the representatives. We must compare the neighborhoods of these instances before we select the first representative. These different positive areas $POS(d)$ comprise universe U, and to cover a corresponding positive area, n times calculations should be performed. When we select the second instance, we need to compare the neighborhoods of the remaining instances. When p representatives are selected, we need to execute $n + (n-1) + \ldots + (n-p+1) = \frac{np(2n-p+1)}{2}$ times comparisons in total. Generally, number of representatives p is far fewer than number of instances n, and the time complexity in this step is $O(n^2 p)$.

□

To summarize, let n be the number of instances, m the number of attributes in the training set, and p the number of representatives. Because $p \leq n$, the time complexity of Algorithm 1 is

$$O(n^2 m) + O(n^2) + O(n^2) + O(n^2 p) = O(n^2(p+m)). \tag{17}$$

3.2. Representative-based classification

The RC algorithm considers each representative and its neighborhood as a decision rule. If a new instance is located in one or a few neighborhoods, the decision rules of these neighborhoods determine the predicted class label. Otherwise the rule-based approach [58] is useless. Therefore, the RC algorithm is inspired by the kNN algorithm; thus, a kNN-like approach is used to solve this problem.

The core issue of kNN-like algorithms is the distance definition or function. For most kNN-based classification algorithms, the distance is usually replaced by a similarity. However, in the RC algorithm, the roles of selected representatives are different; a powerful representative usually has a greater effect on classification. The differences are reflected in neighborhood threshold θ^*; a smaller threshold means a stronger generalization ability. Let R be the set of representatives. The distance between unlabeled instance x' and representative $x \in R$ is

$$distance(x', x) = \frac{1}{sim(x', x)} - \frac{1}{\theta_x^*}. \tag{18}$$

In this Euclidean space, according to Equation 18, a negative distance means that x' is located in the neighborhood of x; otherwise x' is out of the range of this neighborhood.

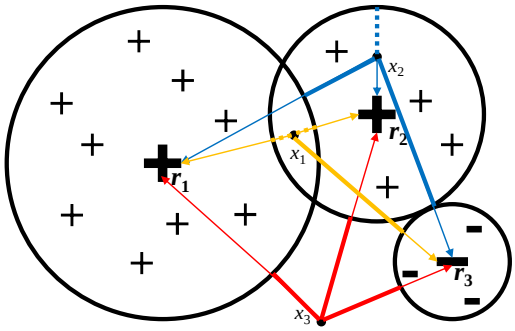


Figure 3: Distance between test instances and representatives.

Moreover, only the closest representatives have the permission to predict the class label. The minimum distance between unlabeled instance x' and representatives x is defined as

$$mds(x', R) = \min\{distance(x', x)|x \in R\}. \tag{19}$$

The set of *electoral representatives* is then

$$E(x') = \{x \in R|distance(x', x) = mds(x', R)\}. \tag{20}$$

This distance function considers not only similarity but also minimum neighborhood threshold θ^*, which is an input of the problem. Note that the distance cannot simply be interpreted as that in Euclidean space. It may be negative or zero, which means that x' is in the range of $N^*(x)$. If the distance is positive, then x' is beyond the control range.

If there is more than one electoral representative, then they predict the class label of x' in concert. For example, three-way decision models [11, 24, 54], the standard voting method, or other approaches can be implemented to solve this conflict. In this case, with set of electoral representatives E, the predicted class label of x' is

$$d'(x') = \underset{1 \le i \le |V_d|}{\arg\max} |\{x \in E(x')|d(x) = i\}|. \tag{21}$$

Note that we select just one representative, but in some other contexts it could be useful to return to the user the entire list of possible labels as a type of upper approximation, which is typical of rough set approaches. For instance, a physician could be interested in knowing not only a specific illness automatically detected given some clinical tests, but a set of possible illnesses on which to restrict his/her attention.

In Fig. 3, it is easy to observe that there are three representatives and two class labels: '$+$' and '$-$'. As an outlier, x_3 does not fall into any neighborhood of these representatives. However, the distance between x_3 and r_1 is the shortest,

Algorithm 2 Representative-based classification *RC)*

Input: Unlabeled instance x', set of representatives R, and covering $CR = \{(x, \theta_x^*)|x \in R\}$.
Output: Predicted class label of x'.

1: $DIS = 0$;
2: $MDS =$ MAX_VALUE;
3: $E = \emptyset$;
4: **for** (each $x \in R$) **do**
5: Compute $sim(x', x)$;
6: Compute distance $DIS = distance(x', x)$ according to Equation (18);
7: **if** $(DIS < MDS)$ **then**
8: $MDS = DIS$;
9: $E = \{x\}$;
10: **else** $\{(DIS = MDS)\}$
11: $E = E \cup \{x\}$;
12: **end if**
13: **end for**
14: Compute $d'(x')$ according to Equation (21);
15: Return $d'(x')$.

and the class label of x_3 is consistent with r_1. For x_2, it falls into the neighborhood of r_2, and has the same class label as r_2. Thus, the unlabeled instance is assigned the class label of the only electoral representative. By contrast, x_1 falls into the neighborhoods of r_1 and r_2 at the same time, and it has the same shortest distance as r_1 and r_2. Hence, r_1 and r_2 are the electoral representatives to predict the class label of x_1 using the vote method.

Algorithm 2 provides the processes for the RC algorithm. The representative with less than the MDS distance can be viewed as a candidate electoral representative. Any much closer representative resets electoral representative set E. Lines 7–12 show that the RC algorithm ensures that only the electoral representatives can be retained.

Proposition 2. *The computational complexity of Algorithm 2 is O(pm), where p is the number of representatives and m is the number of conditional attributes.*

Proof. In Algorithm 2, each unlabeled instance should be compared with all representatives. The distance between an unlabeled instance and representatives is based on their similarity. Regardless of which similarity measure was adopted in the training stage, the similarities between an unlabeled instance and representatives are computed using the *Overlap* measure. For each similarity calculation, m conditional attributes should be computed. The time cost of generating this similarity vector is $p \times m$. The time complexity of determining the electoral representatives is $O(p)$. Thus, there are $p \times (m + 1)$ times computations that should be executed, and the time complexity of Algorithm 2 is

$O(pm)$.

□

4. Experimental evaluation

In this section, we conduct a series of experiments to address the following questions:

1. What type of similarity measure is more appropriate for the RC algorithm?
2. How do the similarity measures influence the classification?
3. Does the RC algorithm with an appropriate similarity measure outperform classical classification algorithms?

4.1. Experimental setup

Experiments were undertaken on 15 real-world datasets from the UCI Repository of Machine Learning Databases [2]. The details about these 15 experimental datasets are listed in Table 4. Among these datasets, Ionosphere, Iris, Penbased, Sonar, WDBC, Waveform, and Wine are continuous. Because the RC algorithm was designed for nominal or categorical datasets, the continuous datasets were pre-discretized using the WEKA [14] software tool. Note that all these datasets had only one decision attribute, as mentioned previously. When we counted the number of attributes, the conditional and decision attributes were included. Moreover, experiments were conducted on a training-testing scenario; the class label of the testing instance was invisible until its class label was predicted.

The general experiment was same for each dataset. Each dataset was divided into two parts randomly using a given division percentage: one used for the training set and the other for the testing set. We linearly increased the scale of the training sets to obtain their classification trends using the RC algorithm. For different datasets, we adopted different division percentages. To help this, the number of instances was used to determine the division strategy. Car, Kr-vs-kp, Waveform, Mushroom, and Penbased have more than 1,000 instances, and we varied the proportions for these data from 0.01 to 0.1 in increments of 0.01. Other datasets were relatively small, and we varied the proportions of them from 0.05 to 0.5 in increments of 0.05. We conducted 100 repeated experiments for each dataset for each division and computed the average classification accuracy.

4.2. Results

In the training stage, similarities between instances were computed for every measure. In the testing stage, the conditions for computing the *Eskin*, *Goodall3*, *Goodall4*, *IOF*, and *OF* measure similarities were unavailable. Because only the set of selected representatives were retained, compared with the training set, the knowledge of the distribution and frequency of the representatives was completely different; that is, the similarities between representatives and unlabeled instances were computed using the *Overlap* measure.

Table 4: Description of the 15 UCI datasets.

Data set	Domain	Instance	Attribute	Class	
Zoo	UCI	Life	101	17	8
Promoters	UCI	Life	106	58	2
Iris	UCI	Life	150	5	3
Wine	UCI	Physical	178	14	3
Sonar	UCI	Physical	208	61	2
Ionosphere	UCI	Physical	351	34	2
Dermatology	UCI	Life	366	34	6
Voting	UCI	Social	435	17	2
WDBC	UCI	Life	569	31	2
Tic-Tac-Toe	UCI	Game	958	10	2
Car	UCI	N/A	1,728	7	4
Kr-vs-kp	UCI	Game	3,196	37	2
Waveform	UCI	Physical	5,000	22	3
Mushroom	UCI	Life	8,124	23	2
Penbased	UCI	Computer	10,992	17	7

Fig. 4 illustrates the average classification accuracies at each given percentile. With increasing numbers of training instances and decreasing testing instances, the classification accuracies should have improved in theory. From Figs. 4 (a) to (o), all the similarity measures adopted in the RC algorithm corresponded to this trend, except the *Goodall4* measure. For example, with a large training set, the *Goodall4*-based RC algorithm had a lower classification accuracy, illustrated in Figs. 4 (d), (f), (h), and (i). Moreover, on the remaining datasets, the classification trends were unusual, and seemed to be independent of the scale of the training set. The results demonstrate another significant phenomenon: the classification trend of the *Goodall3*-based RC algorithm was similar to that of other measures, but the values were lower than theirs at each point (i.e., Zoo, Wine, WDBC, and Mushroom). For the other similarity measures, there was no common or significant difference between them for either their accuracy trends or values.

While the changing trend is one important factor, accuracy is as important. Table 5 lists the accuracy information of these datasets for different similarity measures. To help the comparison and analysis, the best classification results are shown in bold. It needs to be stressed that each accuracy value is the average value of 10 experiments, with the different given division percentages. For example, 0.8554 ± 0.221 is the average value of 10 accuracies computed when the scale of the training set is 0.05, 0.1, $\cdots$ 0.5. The deviation value describes the difference between the best and worst accuracies. A higher average accuracy indicates better veracity, and a lower deviation indicates better stability. From the information in Table 5, the *Goodall3*-based and *Goodall4*-based RC algorithms outperformed no other algorithms; there was no dataset that obtained the best classification using these two algorithms. We found that the *Overlap*-based RC

Table 5: Information of the average classification accuracies.

Data set	*Overlap*		*Eskin*		*Goodall3*		*Goodall4*		*IOF*		*OF*	
	Accuracy	Variance	Accuracy	Variance	Accuracy	Variance	Accuracy	Variance	Accuracy	Variance	Accuracy	Variance
Zoo	0.855	7.95E-3	**0.857**	6.65E-3	0.616	5.40E-3	0.662	**4.92E-4**	0.856	9.49E-3	0.856	6.63E-3
Promoters	0.695	3.90E-3	0.717	2.71E-3	0.684	4.22E-3	0.512	**6.00E-5**	0.714	3.01E-3	**0.719**	2.57E-3
Iris	0.905	1.65E-3	**0.940**	1.04E-3	0.861	1.71E-3	0.697	1.88E-3	0.920	5.70E-3	0.935	**9.81E-4**
Wine	0.915	4.31E-3	0.906	3.84E-3	0.782	6.39E-3	0.478	5.12E-3	**0.915**	5.31E-3	0.914	**3.80E-3**
Sonar	**0.832**	2.77E-3	0.800	2.26E-3	0.780	5.57E-3	0.729	**8.86E-4**	0.829	2.57E-3	0.818	2.09E-3
Ionosphere	**0.903**	**5.61E-4**	0.796	1.70E-3	0.828	1.78E-3	0.390	2.66E-3	0.893	1.39E-3	0.856	1.20E-3
Dermatology	0.860	1.55E-3	0.874	2.58E-3	0.540	3.10E-4	0.555	**2.64E-5**	0.887	2.36E-3	**0.907**	1.72E-3
Voting	0.913	2.12E-4	0.886	2.25E-4	0.901	4.38E-4	0.675	1.00E-3	**0.915**	2.29E-4	0.904	**1.58E-4**
WDBC	0.951	1.30E-4	0.939	**9.03E-5**	0.902	8.78E-4	0.870	2.43E-4	**0.952**	1.00E-4	0.944	2.02E-4
Tic-Tac-Toe	0.773	2.93E-3	**0.803**	3.52E-3	0.736	1.14E-3	0.646	**3.08E-4**	0.762	2.03E-3	0.790	2.35E-3
Car	**0.733**	1.98E-4	0.712	1.27E-3	0.726	2.64E-4	0.698	**1.86E-5**	0.729	1.37E-3	0.719	1.44E-3
Kr-vs-kp	**0.749**	2.06E-3	0.747	1.89E-3	0.621	1.67E-3	0.667	**3.32E-4**	0.749	1.90E-3	0.744	1.88E-3
Waveform	**0.756**	3.66E-4	0.708	1.65E-4	0.753	5.52E-4	0.542	**1.28E-4**	0.752	2.90E-4	0.715	2.17E-4
Mushroom	0.982	**1.67E-4**	0.965	2.79E-4	0.857	1.08E-3	0.806	3.04E-5	0.982	2.04E-4	**0.984**	1.85E-4
Penbased	0.865	2.52E-3	0.869	2.60E-3	0.857	2.77E-3	0.146	**2.22E-5**	0.866	2.38E-3	**0.877**	2.42E-3
Mean Accuracy	0.854	3.12E-3	0.844	3.08E-3	0.756	3.41E-3	0.602	**1.32E-3**	**0.856**	3.83E-3	0.855	2.78E-3
Mean Rank	1.8666		2.6666		4.5333		5.8000		**1.6000**		1.8666	

algorithm obtained its best classification results on five datasets and outperformed other similarity measures. Additionally, the *OF*-based, *IOF*-based, and *Eskin*-based RC algorithms obtained the highest accuracies on four, three, and three datasets, respectively.

The *Overlap* measure appears to be the most appropriate measure for the RC algorithm if we count the best results in Table 5. However, this simple statistical result is not strong evidence to demonstrate the superiority of the *Overlap* measure. Through observation, some different measures can obtain classification results that are very close to the best results. Moreover, the gap between some accuracies is significant. Thus, we used the mean accuracy and mean rank to evaluate the performance of these measures. We ranked and assigned the six similarity measures using their accuracies: the best measure was assigned 1, the second-best was assigned 2, and so on. Another aspect to note is that if the difference between two accuracies was less than 1%, then these two measures had the same ranking and were assigned the same score. Then, the differences between all of these accuracies were amplified. According to the mean rank, the *IOF* measure was followed by the *Overlap*, *OF*, *Eskin*, *Goodall3*, and *Goodall4* measures. Through statistical analysis, the mean accuracy of the *IOF*-based RC algorithm was 0.8568 ± 0.120, and the mean rank was 1.6000. Among all these measures, *IOF* was the most appropriate measure for the RC algorithm. By contrast, the *Overlap* and *IOF* measures were comparable with the *OF* measure, for both their mean accuracies and mean ranks.

Combining the general conclusions from Fig. 4 and Table 5, we can answer the first question: The *Overlap*, *OF*, and *Eskin* measures are good candidate measures, and the *IOF* measure is more appropriate for the RC algorithm.

The classification differences come from both the training and testing stages. Figs. 5 and 6 illustrate the similarity measure influences from the training and testing stages, respectively. In the training stage, the most similar contradictory instance determined threshold θ of each labeled instance. As Fig. 5 shows, $\theta^*_{r_i}$ was 0.6, the next candidate threshold 0.5 was not valid because the 0.5 threshold made representative r_1 contain some contradictory instances that belonged to $N^*(r_2)$ or $N^*(r_3)$. Hence, with this similarity measure, four instances were selected as the representatives. However, another measure generated a candidate threshold of 0.55, so that $N^*(r_1)$ could cover $N^*(r_4)$, and finally, three instances were selected as representatives. In the testing stage, the distances computed using similarities and thresholds determined the class label prediction. As Fig. 6 shows, unlabeled instance x had the same minimum distance from r_1 and r_3 based on the similarity measures. If we replaced the measure, unlabeled instance x had only one minimum distance from r_3, and the class label prediction was changed. Generally, these different similarity measures calculate different thresholds and influence the classification.

Because these similarity values were computed from different measures, it is unfair to compare these values directly. For example, the similarity of the *Eskin* measure belongs to [1/3, 1] and the similarity of the *Overlap* measure belongs to [0, 1]. Two instances that have an *Eskin* measure similarity value of 0.5 may be not more similar than two instances that have an *Overlap* measure similarity

value of 0.4. To compare these similarities, we set up an equivalent model of the above similarities that referred to the *Overlap* measure. We determined the boundary instances of one representative using one of these similarity measures, and we recomputed the *Overlap* measure similarities between those instances. Hence, we normalized the similarity values of the boundary instances based on the *Overlap* measure. Let the *Eskin* similarity between a boundary instance x and a representative y be 0.8, the *Overlap* similarity might be 0.6. Similarly, we recomputed the *IOF*, *OF*, *Goodall3*, and *Goodall4* measures with respect to the *Overlap* measure, which were 0.45, 0.55, 0.51, and 0.55, respectively. Thus, we compared these different similarities in a fair manner.

For simplicity, we used the largest training set (50% for smaller datasets and 10% for larger datasets) for the experiment and statistics. Fig. 7 illustrated the initial and converted average values of the thresholds. These converted thresholds apparently make no great difference compared with the initial values; however, a small difference can make a widely different classifier. Additionally, Fig. 7 only shows the average θ values on the largest training set; for the individual representatives, the differences were more significant. The *Goodall4* measure obtained the minimum average thresholds on 13 datasets, except for Iris and Tic-Tac-Toe. Combined with Table 5, *Goodall4* was the worst measure for the RC algorithm. For other measures, those that obtained the maximum average threshold almost obtained the best classification accuracies, such as Zoo, Promoters, Sonar, Ionosphere, Voting, WDBC, Car, Waveform, and Mushroom. Thus, for the *Eskin*, *Overlap*, *IOF*, and *OF* measures, the higher the threshold value, the higher the classification accuracy. We can answer the second question regarding how the measures influence the classification accuracy by generating different representatives.

We analyzed the classification trends when the scale of the training set was continuously enlarged. *IOF* was the most appropriate similarity measure for the RC algorithm when the training set was small. To verify whether the RC algorithm with different similarity measures outperformed other classical classification algorithms (i.e., ID3, C4.5, and Naïve Bayes), the 10-fold cross-validation method [3] was applied to all the experimental datasets. Table 6 shows the details of these experiments. The best classification results for the RC classifier with different measures are shown in italics, and the best classification results of the nine classifiers is shown in bold. We ranked and assigned these classifiers using their values; the strategy was the same as that mentioned previously. Clearly, Naïve Bayes achieved the highest accuracy on seven out of the 15 datasets, which was better than ID3 and C4.5. However, the Naïve Bayes classifier was worse than the *Overlap*-based RC classifier, considering the mean accuracy and mean rank. This phenomenon indicates that some classifiers outperformed slightly on some datasets, but had significant disadvantages on other datasets. For RC-based algorithms, the *Overlap*, *IOF*, and *OF* measures performed similarly, and the *Overlap* measure was better than the other two measures. According to the mean accuracy, the RC algorithm with the *Overlap* measure outperformed the classical algorithms: the RC algorithm with the *IOF* measure was better than ID3 and Naïve Bayes, and was comparable

to C4.5. Thus, we can answer the third question regarding whether the RC algorithm with an appropriate similarity measure is better than some classical classification algorithms.

Table 6: Average classification accuracies using 10-fold cross-validation.

Data set	Accuracy								
	RC						ID3	C4.5	Naïve Bayes
	Overlap	*Eskin*	*Goodall3*	*Goodall4*	*IOF*	*OF*			
Zoo	*0.9504*	0.9286	0.7346	0.5609	0.9453	0.9334	**0.9661**	0.9241	0.9387
Promoters	*0.8496*	0.8089	0.8392	0.5000	0.8296	0.8191	0.7440	0.7863	**0.9113**
Iris	0.9400	0.9593	0.9226	0.7026	0.9566	***0.9606***	0.9446	0.9366	0.9500
Wine	0.9775	0.9736	0.8818	0.3997	0.9701	*0.9787*	0.9573	0.9298	**0.9882**
Sonar	**0.9115**	0.8687	0.9055	0.7858	0.9066	0.8777	0.7410	0.8366	**0.9365**
Ionosphere	0.9299	0.8561	0.9148	0.3589	***0.9373***	0.9162	0.9088	0.9201	0.9017
Dermatology	0.9153	0.9513	0.5683	0.5576	0.9407	*0.9608*	0.9211	0.9418	**0.9784**
Voting	0.9292	0.9018	0.9243	0.6088	*0.9319*	0.9183	0.9361	**0.9521**	0.9007
WDBC	0.9622	0.9588	0.9045	0.7987	***0.9655***	0.9620	0.9481	0.9578	0.9583
Tic-Tac-Toe	0.9532	***0.9615***	0.8622	0.6535	0.9109	0.8948	0.8524	0.8525	0.6963
Car	*0.8526*	0.6658	0.7017	0.7002	0.7614	0.7799	0.8976	**0.9245**	0.8560
Kr-vs-kp	0.8573	*0.8596*	0.6404	0.6540	0.8530	0.8583	**0.9964**	0.9943	0.8774
Waveform	0.7938	0.7464	***0.8019***	0.4997	0.7768	0.7480	0.7358	0.7765	0.7934
Mushroom	0.9999	0.9959	0.9176	0.8356	***1.0000***	***1.0000***	**1.0000**	**1.0000**	0.9547
Penbased	0.9651	0.9662	0.9638	0.1331	0.9643	***0.9707***	0.9414	0.9496	0.8574
Mean Accuracy	***0.9191***	0.8935	0.8322	0.5832	0.9100	0.9052	0.8993	0.9121	0.8999
Mean Rank	***2.2666***	3.9333	5.4000	8.7333	2.6000	3.2666	4.8666	3.9333	3.5333

5. Conclusion and further work

In this paper, we compared the classification accuracy of the RC algorithm with six similarity measures. For the RC algorithm, classifier generation and label prediction were based on the similarities between instances. Although these measures have their own focuses, the core methods of influencing the classification are the same.

Based on the experimental results in Tables 5 and 6, the RC algorithm with the *Goodall3* measure performed better when the training set was sufficiently large. However, the RC algorithm with the *Goodall4* measure had a poor ability to classify, regardless of the dataset or scale of the training set. Thus, we can conclude that the *Goodall3* measure is more appropriate for the RC algorithm when the training set is very large, and the *Goodall4* measure is inappropriate for the RC algorithm. Similarly, the *Goodall3*, *Goodall4*, *IOF*, and *OF* measures are all concerned with the frequency of attribute values, and the latter two outperformed the former two in most cases. Therefore, if the frequency of attribute values should be considered, then the *IOF* measure may be the most appropriate similarity measure for the RC algorithm. The RC algorithm with

the *Eskin* measure typically obtained better accuracies on the Zoo, Tic-Tac-Toe, and Iris datasets. These three datasets have similar or the same number of values of conditional attributes. When the dataset had the same or similar numbers of values for each of its conditional attributes, the *Eskin* measure was the most appropriate measure for the RC algorithm. By contrast, if there was no particular requirement, then the *Overlap* measure was the best choice because of its stability and universality. Additionally, there was no measure that was applied to every dataset; an appropriate measure should be analyzed in each specific case.

In future work, other scenarios can be considered, such as the cost-sensitive [27, 28, 21, 57] scenario and active learning [42] scenario. We also plan to compare the similarity measures with inconsistent datasets or missing value datasets. Finally, we would like to design a mechanism to determine the most appropriate similarity measure for the RC algorithm.

Acknowledgements

This work was supported by the National Natural Science Foundation of China [grant numbers 61379089, 61573321, 41631179]; and the Zhejiang Provincial Natural Science Foundation of China [grant number LY18F030017].

[1] U. Agharese, R. Osaseri, Diabetes diagnosis model using rough set and k-nearest neighbor classifier, in: International Conference on Artificial Intelligence & Computer Science, 2016.

[2] K. Bache, M. Lichman, UCI machine learning repository, http://archive.ics.uci.edu/ml/ (2013).

[3] Y. Bengio, Y. Grandvalet, No unbiased estimator of the variance of k-fold cross-validation, Journal of Machine Learning Research 5 (2004) 1089–1105.

[4] D. Bianucci, G. Cattaneo, Information entropy and granulation co-entropy of partitions and coverings: a summary, LNCS Transactions on Rough Sets X (2009) 15–66.

[5] D. Bianucci, G. Cattaneo, D. Ciucci, Entropies and co-entropies of coverings with application to incomplete information systems, Fundamenta Informaticae 75 (1-4) (2007) 77–105.

[6] S. Boriah, V. Chandola, V. Kumar, Similarity measures for categorical data: a comparative evaluation, in: SIAM International Conference on Data Mining, 2008.

[7] D.-G. Chen, X.-X. Zhang, W.-L. Li, On measurements of covering rough sets based on granules and evidence theory, Information Sciences 317 (2015) 329–348.

[8] H.-M. Chen, T.-R. Li, C. Luo, J. Hu, Dominance-based neighborhood rough sets and its attribute reduction, in: International Joint Conference on Rough Sets, 2015.

[9] M. S. Chen, J.-W. Han, P. S. Yu, Data mining: an overview from database perspective, IEEE Transactions on Knowledge and Data Engineering 8 (6) (1996) 866–883.

[10] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, S. Stolfo, A geometric framework for unsupervised anomaly detection, in: Applications of Data Mining in Computer Security, Springer US, 2002, pp. 77–101.

[11] C. Gao, Y. Y. Yao, Actionable strategies in three-way decisions, Knowledge-Based Systems 133 (2017) 141–155.

[12] D. W. Goodall, A new similarity index based on probability, Biometrics 22 (4) (1966) 882–907.

[13] G.-D. Guo, A. K. Jain, W.-Y. Ma, H.-J. Zhang, Learning similarity measure for natural image retrieval with relevance feedback, IEEE Transactions on Neural Networks 13 (4) (2001) 811–820.

[14] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten, The WEKA data mining software: an update, ACM SIGKDD Explorations Newsletter 11 (1) (2009) 10–18.

[15] Q.-H. Hu, D.-R. Yu, J.-F. Liu, C.-X. Wu, Neighborhood rough set based heterogeneous feature subset selection, Information Sciences 178 (18) (2008) 3577–3594.

[16] Q.-H. Hu, D.-R. Yu, Z.-X. Xie, Neighborhood classifiers, Expert Systems with Applications 34 (2) (2008) 866–876.

[17] S. J. Karen, A statistical interpretation of term specificity and its application in retrieval, Journal of Documentation 60 (5) (2004) 493–502.

[18] S. B. Kotsiantis, Supervised machine learning: a review of classification techniques, Informatica 31 (2007) 249–268.

[19] S. S. Kumar, H. H. Inbarani, A. T. Azar, K. Polat, Covering-based rough set classification system, Neural Computing & Applications 28 (10) (2017) 2879–2888.

[20] G.-M. Lang, D.-Q. Miao, T. Yang, M.-J. Cai, Knowledge reduction of dynamic covering decision information systems when varying covering cardinalities, Information Sciences 346–347 (2016) 236–260.

[21] H.-X. Li, X.-Z. Zhou, J.-B. Zhao, B. Huang, Cost-sensitive classification based on decision-theoretic rough set model, in: International Conference on Rough Sets and Knowledge Technology, Springer Berlin Heidelberg, 2012.

[22] J.-H. Li, Y. Ren, C.-L. Mei, Y.-H. Qian, X.-B. Yang, A comparative study of multigranulation rough sets and concept lattices via rule acquisition, Knowledge-Based Systems 91 (2016) 152–164.

[23] W.-W. Li, Z.-Q. Huang, X.-Y. Jia, X.-Y. Cai, Neighborhood based decision-theoretic rough set models, International Journal of Approximate Reasoning 69 (2016) 1–17.

[24] X.-N. Li, H.-J. Yi, Y.-H. She, B.-Z. Sun, Generalized three-way decision models based on subset evaluation, International Journal of Approximate Reasoning 83 (2017) 142–159.

[25] C.-H. Liu, D.-Q. Miao, J. Qian, On multi-granulation covering rough sets, International Journal of Approximate Reasoning 55 (6) (2014) 1404–1418.

[26] L.-W. Ma, Two fuzzy covering rough set models and their generalizations over fuzzy lattices, Fuzzy Sets & Systems 294 (2016) 1–17.

[27] F. Min, H.-P. He, Y.-H. Qian, W. Zhu, Test-cost-sensitive attribute reduction, Information Sciences 181 (22) (2011) 4928–4942.

[28] F. Min, W. Zhu, Attribute reduction of data with error ranges and test costs, Information Sciences 211 (2012) 48–67.

[29] A. Mitra, S. R. Satapathy, S. Paul, Clustering analysis in social network using covering based rough set, in: IEEE International Advance Computing Conference, 2013.

[30] H. Mohammadzadeh, T. Gottron, F. Schweiggert, G. Heyer, Finder: extracting the headline of news web pages based on cosine similarity and overlap scoring similarity, in: Proceedings of the Workshop on Web Information and Data Management, 2012.

[31] K. P. Murphy, Machine learning: a probabilistic perspective, MIT Press, 2012.

[32] Z. Pawlak, Rough sets: theoretical aspects of reasoning about data, Kluwer Academic Publishers, 1992.

[33] Y. Ping, Y.-H. Lu, Neighborhood rough set and SVM based hybrid credit scoring classifier, Expert Systems with Applications 38 (9) (2011) 11300–11304.

[34] P. Prabhavathy, B. K. Tripathy, An integrated covering-based rough fuzzy set clustering approach for sequential data, International Journal of Reasoning-based Intelligent Systems 7 (3/4) (2015) 296–304.

[35] Y.-H. Qian, J.-Y. Liang, Y. Y. Yao, C.-Y. Dang, MGRS: a multi-granulation rough set, Information Sciences 180 (6) (2010) 949–970.

[36] J. R. Quinlan, Induction of decision trees, Machine Learning 1 (1) (1986) 81–106.

[37] J. R. Quinlan, C4.5: programs for machine learning, Elsevier, 2014.

[38] I. Rish, An empirical study of the naïve bayes classifier, Journal of Universal Computer Science 1 (2) (2001) 41–46.

[39] C. Stanfill, D. L. Waltz, Toward memory-based reasoning, Communications of the ACM 29 (12) (1986) 1213–1228.

[40] L.-R. Su, W. Zhu, Closed-set lattice and modular matroid induced by covering-based rough sets, International Journal of Machine Learning & Cybernetics 8 (1) (2017) 191–201.

[41] C.-Z. Wang, Q. He, D.-G. Chen, Q.-H. Hu, A novel method for attribute reduction of covering decision systems, Information Sciences 254 (2014) 181–196.

[42] M. Wang, F. Min, Y.-X. Wu, Z.-H. Zhang, Active learning through density clustering, Expert Systems with Applications 85 (2017) 305–317.

[43] R. Wang, D.-G. Chen, S. Kwong, Fuzzy-rough-set-based active learning, IEEE Transactions on Fuzzy Systems 22 (6) (2014) 1699–1704.

[44] S.-L. Wang, X.-L. Li, J.-F. Xia, X.-P. Zhang, Weighted neighborhood classifier for the classification of imbalanced tumor dataset, Journal of Circuits Systems & Computers 19 (1) (2010) 259–273.

[45] S.-L. Wang, X.-L. Li, S.-W. Zhang, Neighborhood rough set model based gene selection for multi-subtype tumor classification, in: Advanced Intelligent Computing Theories and Applications, Springer Berlin Heidelberg, 2008.

[46] S.-P. Wang, W. Zhu, Q.-X. Zhu, F. Min, Characteristic matrix of covering and its application to boolean matrix decomposition, Information Sciences 263 (1) (2012) 186–197.

[47] I. H. Witten, E. Frank, M. A. Hall, Data mining: practical machine learning tools and techniques, 2nd ed., Morgan Kaufmann Publishers Inc., 2011.

[48] S.-P. Xu, X.-B. Yang, E. C. C. Tsang, E. A. Mantey, Neighborhood collaborative classifiers, in: International Conference on Machine Learning and Cybernetics, 2017.

[49] T. Yang, Q.-G. Li, Reduction about approximation spaces of covering generalized rough sets, International Journal of Approximate Reasoning 51 (3) (2010) 335–345.

[50] Y. Y. Yao, Rough sets, neighborhood systems and granular computing, in: IEEE Canadian Conference on Electrical and Computer Engineering, vol. 3, 1999.

[51] Y. Y. Yao, B.-X. Yao, Covering based rough set approximations, Information Sciences 200 (1) (2012) 91–107.

[52] W. Zakowski, Approximations in the space (u, π), Demonstratio mathematica 16 (3) (1983) 761–769.

[53] B.-W. Zhang, F. Min, D. Ciucci, Representative-based classification through covering-based neighborhood rough sets, Applied Intelligence 43 (4) (2015) 840–854.

[54] H.-R. Zhang, F. Min, B. Shi, Regression-based three-way recommendation, Information Sciences 378 (2017) 444–461.

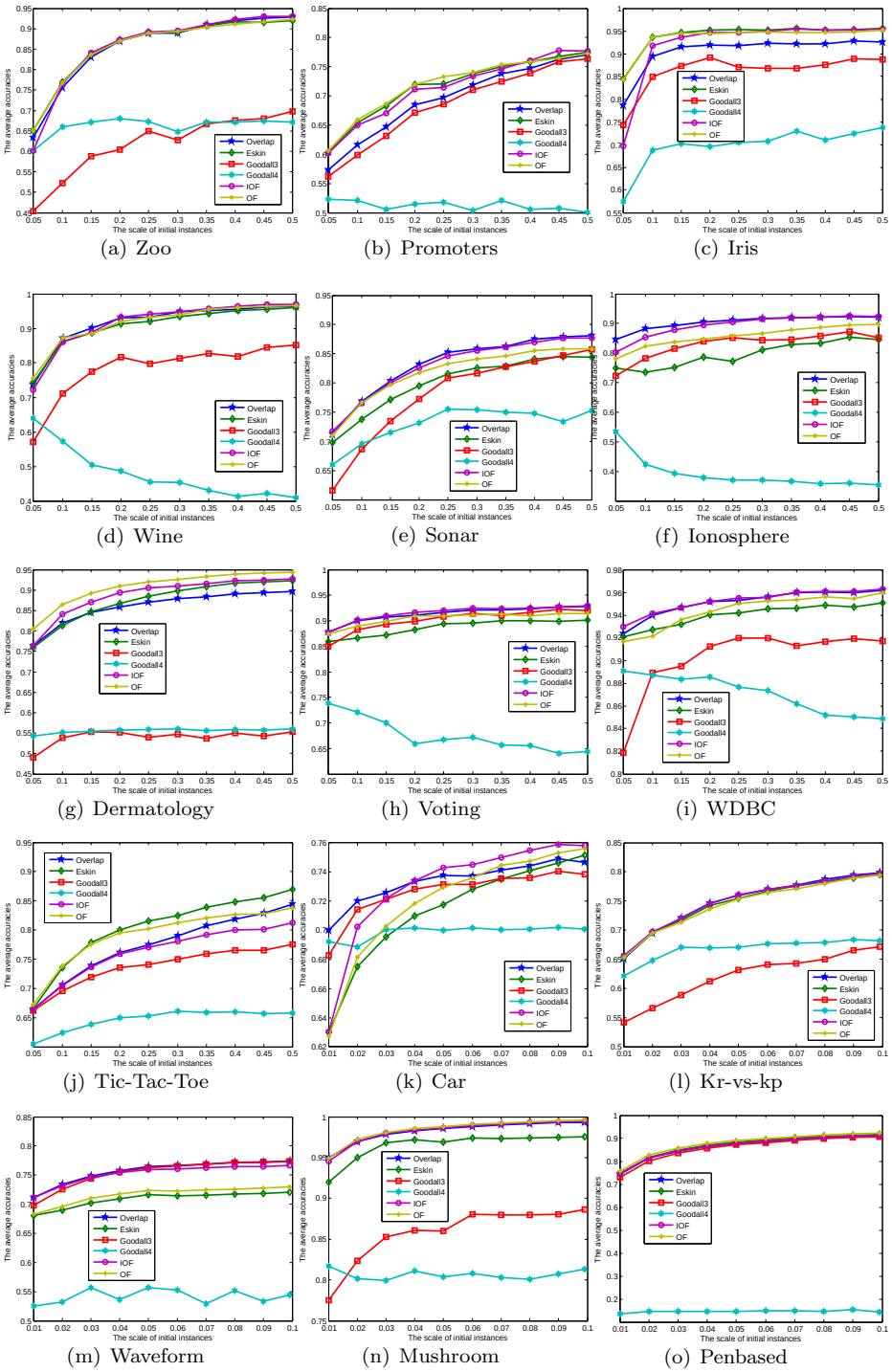
[55] J.-B. Zhang, T.-R. Li, H.-M. Chen, Composite rough sets for dynamic data mining, Information Sciences 257 (2014) 81–100.

[56] H. Zhao, P.-F. Zhu, P. Wang, Q.-H. Hu, Hierarchical feature selection with recursive regularization, in: Proceedings of the 26th International Joint Conference on Artificial Intelligence, 2017.

[57] H. Zhao, W. Zhu, Optimal cost-sensitive granularization based on rough sets for variable costs, Knowledge-Based Systems 65 (2014) 72–82.

[58] S.-Y. Zhao, E. C. C. Tsang, D.-G. Chen, X.-Z. Wang, Building a rule-based classifier – a fuzzy-rough set approach, IEEE Transactions on Knowledge & Data Engineering 22 (5) (2010) 624–638.

[59] W. Zhu, Topological approaches to covering rough sets, Information Sciences 177 (6) (2007) 1499–1508.



(a) Zoo (b) Promoters (c) Iris

(d) Wine (e) Sonar (f) Ionosphere

(g) Dermatology (h) Voting (i) WDBC

(j) Tic-Tac-Toe (k) Car (l) Kr-vs-kp

(m) Waveform (n) Mushroom (o) Penbased

Figure 4: Comparison of average accuracies for six similarity measures.

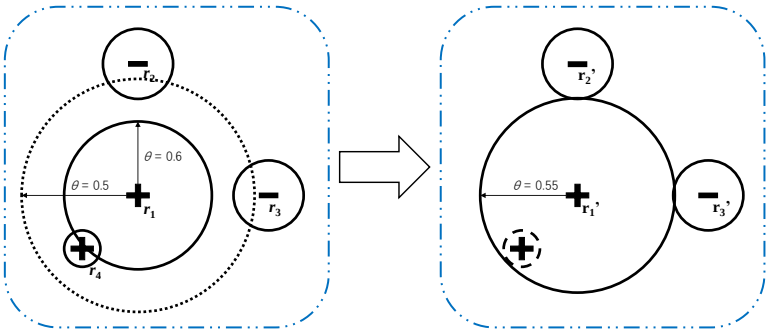


Figure 5: Illustration of influence in the training stage.

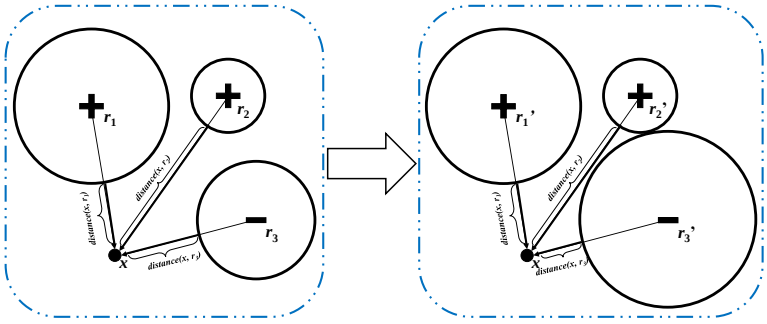
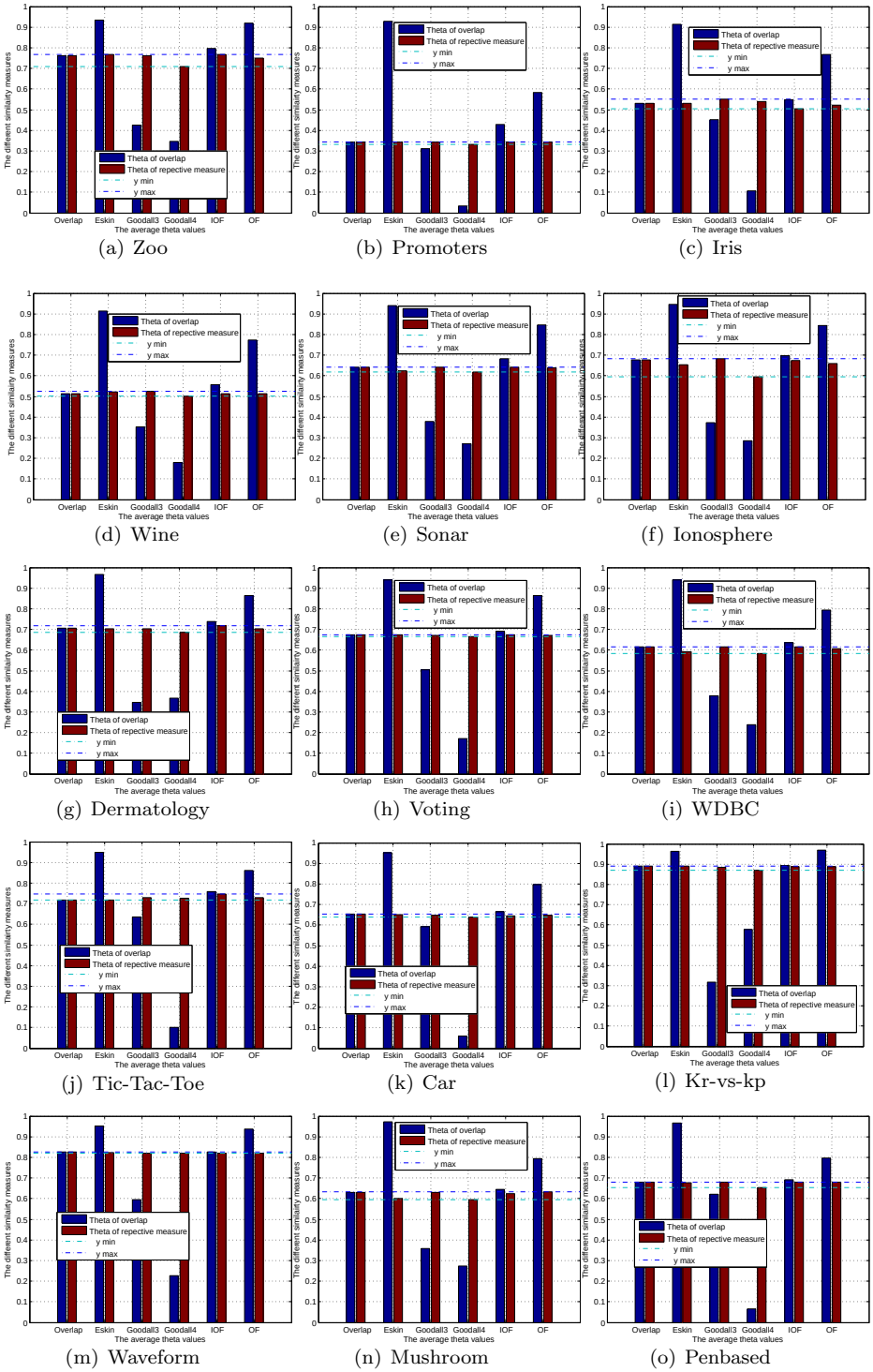


Figure 6: Illustration of influence in the testing stage.



(a) Zoo (b) Promoters (c) Iris

(d) Wine (e) Sonar (f) Ionosphere

(g) Dermatology (h) Voting (i) WDBC

(j) Tic-Tac-Toe (k) Car (l) Kr-vs-kp

(m) Waveform (n) Mushroom (o) Penbased

Figure 7: Comparison of average θ values for six similarity measures.