JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 39, 353-373 (2023)
DOI: 10.6688/JISE.202303_39(2).0007

A Partition-induced Hereditary Metric for Equivalence Classes – Theory and Application*

RAY-MING CHEN
School of Mathematical Sciences
College of Science and Technology
Wenzhou-Kean University
Wenzhou, 325015 P.R. China
E-mail: copydrop4321@gmail.com

Measuring the distance between equivalence classes has its theoretical and practical merit, in particular, in the aspect of rough sets or the application on information systems. The typical metric for measuring the distance between partitions is the Hausdorff metric. Another candidate is the minimal matching metric which matches the pairwise minimal distance between the compartments. However, both methods need to involve or imbed Jaccard metric, which is essentially a static metric and less informative, since it scales the distance between 0 and 1. In this article, we devise a third metric which is defined inductively by some non-negative real functions. This mechanism enables its flexibility in applying metrics in real problems and delve deeper into the structures. We then apply this hereditary metric on two occasions: one with simulated data regarding algorithms and the other with real data regarding ontology population process. This metric per se is suitable for categorising procedures, methods, or other attributes.

Keywords: equivalence classes, information system, hereditary distance, ontology population, algorithms

1. INTRODUCTION

Seeking or finding the underlying correlation or causality between variables are vital in forming knowledge and decision making. However, in real world, one needs to involve the uncertainty when modelling the real situations. Traditionally people tend to use rough sets, fuzzy set or soft sets to deal with such uncertainty [1–3]. Other coordinating theories or tools, such as statistical methods, or the combination of these analytical techniques are normally required in order to combat some real complicated problems [4–6]. Correlation could be revealed either by similarity indexes or distance functions. For the similarity index parts, in addition to solely mathematical approaches or analysis, there exist other parametric and non-parametric statistical methods, such as Pearson/Spearman correlation, Paired t-test/Wilcoxon rank sum test, unpaired t test/Mann-Whitney U test [7, 8], *etc*. Regarding the distance function part, there are mainly three types of distance functions for partition space: minimal matching of two partitions [9], Hausdorff metric (or its variant [10]) embedded with Jaccard metric (indeed Hausdorff metric could be embedded with

Received December 14, 2021; revised March 27 & May 5, 2022; accepted May 11, 2022.
Communicated by Tzung-Pei Hong.
* This work is supported by the Humanities and Social Science Research Planning Fund Project under the Ministry of Education of China (Grant No. 20XJAGAT001).

any distance function, but for nominal data, Jaccard metric is a good candidate), and the hereditary metric devised in this article. The advantage for Hausdorff metric used in partition space is the computational convenience and its wide applications [11, 12]; however there exist some disadvantages for this metric, when one deals with nominal type data, in particular the embedded Jaccard metric, which is essentially a static metric and which squeezes the values of Hausdorff distance for partitions within 0 and 1 – this makes it more like a camouflage of similarity index. Moreover, Jaccard metric tends to ignore the data structures, or more precisely, it ditches the structures and purely focuses on the nominal difference between measured entities. On the other hand, the minimal matching method is very computational resource consuming, since it considers all the permutations between the measured entities. In addition, it is mainly designed for numerical positive data. For the nominal type data, it also needs to embed other set-related metric, such as Jaccard metric.

This paper is motivated by these disadvantages and by the setting of rough sets, in particular the treatment of information systems or data via equivalence classes or partitions. Therefore, we need to devise a metric that could properly measure the distance between equivalence classes that could minimise the above-mentioned disadvantages and provide much flexibility in real application. In this article, we devise a metric based on the hereditary values derived via three non-negative functions τ, σ and ρ: τ function tackles the measurement of subsets of a universe (or it could be deemed as a weighing function for compartments in a partition), which makes it capable of adaptable to real application, while σ and ρ functions record the hereditary difference between structures (or compartments). This also avoids the typical Hausdorff metric, which tends to ignore the structure difference.

A complete description and proof of this hereditary metric and its applications are presented in this article. In Section 2, we define the partition operation (union, intersection, and partial ordering) and their properties. In Section 3, we define a hereditary metric – a metric deductively defined via three non-negative real functions τ, σ and ρ – and show it satisfies all the axiomatic criteria for a metric. In Section 4, we apply the hereditary metric on an information system regarding algorithms and problems, and study the correlation between problems. In Section 5, we further apply this metric on some real data regarding ontology population process (OPP). We study the correlation between the procedures for OPP. There are several characteristics of this hereditary metric:

1. This metric generalizes the distance function δ between two non-empty pure sets A and B, where $\delta(A,B)$ is defined by $\delta(A,B) = \frac{|A|+|B|}{2} - |A \cap B|$. This format of metric could be further extended into or derived from other (spatial) structural metric [13, 14];

2. It serves as an alternative for Hausdorff metric over partition space. Our metric keeps some degree of flexibility by introducing several hierarchies into the structure of the distance function – this property also enables us to amend the metric according to the real problems and applications;

3. This metric enables one to assigning weights to compartments or partitions, which shall further enrich the application of rough sets.

Let us summarize the main contributions of this article:

1. We have devised a hereditary metric that measures the distance between any two equivalence classes. The proposed metric is much practical in real applications;

2. We have shown how to apply our distance function on information systems regarding problem classification and ontology population process procedure categorisation;

3. Our distance function could be further applied on classifying algorithms, clustering analysis, or even classifiers when they are treated as the attributes.

2. DEFINITIONS AND CLAIMS

Let S be an arbitrary non-empty finite set and $|S|$ be its cardinality. Let $P(S)$ denote the power set of S and $PP(S)$ denote the power set of $P(S)$. One observes that $PT(S) \subseteq PP(S)$. Let $PT(S)$ denote the set of all the partitions of S. For any set of sets $\mathbb{K} = \{K_1, K_2, ..., K_n\}$, let $\cup\mathbb{K}$ denote the union $K_1 \cup K_2 ... \cup K_n$. If all the elements in $\mathbb{K}$ are pairwise disjoint, we use the notation $\uplus\mathbb{K}$ to represent such union operation. Let $\mathbb{A}, \mathbb{B}, \mathbb{C} \in PT(S)$ and $\mathbb{U}, \mathbb{V} \in PP(S)$ be arbitrary throughout this article.

Definition 2.1. (operation $\sqcap$ on $PP(S)$) Define $\mathbb{U} \sqcap \mathbb{V} = \{U \cap V : U \cap V \neq \emptyset, U \in \mathbb{U}, V \in \mathbb{V}\}$.

Basically it is the set for two sets under $\cap$-associative law, in which $\cap$ corresponds to logical operator "and" under uncertain reasoning. The distance connects to the concept of overlapping: one could expect that two sets are closer if and only if their overlap is larger.

Example 1. Let us link the concept of equivalence classes with rough set theory. Suppose the universe U consists of 11 objects (or $U = \{b_1, b_2, \cdots, b_{11}\}$). Suppose there are three attributes: a_1, a_2 and a_3. The attribute values for a_1 are good, bad and normal; for a_2 are long and short; and for a_3 are yellow, red and green. The corresponding information system is shown in Table 1. The induced partitions based on the attribute-induced

Table 1. Information system.

U	attribute 1 or a_1	attribute 2 or a_2	attribute 3 or a_3
b_1	good	long	yellow
b_2	bad	long	red
b_3	good	short	green
b_4	normal	short	green
b_5	normal	short	yellow
b_6	bad	long	yellow
b_7	bad	short	red
b_8	normal	long	red
b_9	bad	short	green
b_{10}	bad	long	green
b_{11}	good	short	red

equivalence relations are:

- $U/a_1 = \{a_{11} = \{b_1, b_3, b_{11}\}, a_{12} = \{b_2, b_6, b_7, b_9, b_{10}\}, a_{13} = \{b_4, b_5, b_8\}\} \in PT(U)$;
- $U/a_2 = \{a_{21} = \{b_1, b_2, b_6, b_8, b_{10}\}, a_{22} = \{b_3, b_4, b_5, b_7, b_9, b_{11}\}\} \in PT(U)$;
- $U/a_3 = \{a_{31} = \{b_1, b_5, b_6\}, a_{32} = \{b_2, b_7, b_8, b_{11}\}, a_{33} = \{b_3, b_4, b_9, b_{10}\}\} \in PT(U)$.

By the above definition, one has

- $U/a_1 \sqcap U/a_2 = \{\{b_1\}, \{b_3, b_{11}\}, \{b_2, b_6, b_{10}\}, \{b_7, b_9\}, \{b_8\}, \{b_4, b_5\}\} \in PT(U)$;
- $U/a_1 \sqcap U/a_3 = \{\{b_1\}, \{b_3\}, \{b_{11}\}, \{b_6\}, \{b_2, b_7\}, \{b_9, b_{10}\}, \{b_5\}, \{b_8\}, \{b_4\}\} \in PT(U)$;
- $U/a_2 \sqcap U/a_3 = \{\{b_1\}, \{b_2, b_8\}, \{b_6\}, \{b_{10}\}, \{b_5\}, \{b_7, b_{11}\}, \{b_3, b_4, b_9\}\} \in PT(U)$.

Claim 1. $\mathbb{A} \sqcap \mathbb{A} = \mathbb{A}, \mathbb{A} \sqcap \mathbb{B} = \mathbb{B} \sqcap \mathbb{A}$, *and* $(\mathbb{A} \sqcap \mathbb{B}) \sqcap \mathbb{C} = \mathbb{A} \sqcap (\mathbb{B} \sqcap \mathbb{C})$.

Proof. The results follow directly from Definition 2.1. □

Claim 2. $(\uplus\mathbb{A}) \cap (\uplus\mathbb{B}) = S = \uplus(\mathbb{A} \sqcap \mathbb{B})$.

Proof. Since $\uplus\mathbb{A} = \uplus\mathbb{B} = S$, it suffices to show $(\uplus\mathbb{A}) \cap (\uplus\mathbb{B}) = \uplus(\mathbb{A} \sqcap \mathbb{B})$. Let $x \in (\uplus\mathbb{A}) \cap (\uplus\mathbb{B})$ be arbitrary. Then one has $x \in \uplus\mathbb{A}$ and $x \in \uplus\mathbb{B}$, *i.e.*, there exist $A \in \mathbb{A}$ and $B \in \mathbb{B}$ such that $x \in A \cap B$. This, by Definition 2.1, shows $x \in \uplus(\mathbb{A} \sqcap \mathbb{B})$. On the other hand, by assuming $x \in \uplus(\mathbb{A} \sqcap \mathbb{B})$, we could reach the result that $x \in (\uplus\mathbb{A}) \cap (\uplus\mathbb{B})$. □

Corollary 1. *(closure) $PT(S)$ is closed under $\sqcap$.*

Proof. Let $\mathbb{A}, \mathbb{B} \in PT(S)$ be arbitrary. We show $\mathbb{A} \sqcap \mathbb{B} \in PT(S)$. In Claim 2, we have shown $\cup(\mathbb{A} \sqcap \mathbb{B}) = S$. Hence it suffices to show every distinct $C, D \in \mathbb{A} \sqcap \mathbb{B}, C \cap D = \emptyset$. Suppose $C = A_C \cap B_C, D = A_D \cap B_D$; where $A_C, A_D \in \mathbb{A}$ and $B_C, B_D \in \mathbb{B}$. If $C \cap D \neq \emptyset$, then $A_C \cap A_D \neq \emptyset$, and $B_C \cap B_D \neq \emptyset$,*i.e.*, $A_C = A_D$ and $B_C = B_D$, *i.e.*, $C = D$, a contradiction. Therefore $C \cap D = \emptyset$. □

Definition 2.2. (refinement $\leq$ on $PP(S)$) Define $\mathbb{U} \leq \mathbb{V}$ if and only if $\forall V \in \mathbb{V} \exists \mathbb{Z} \subseteq \mathbb{U}$ such that $\uplus\mathbb{Z} = V$. We say $\mathbb{U}$ is a refinement of $\mathbb{V}$. If $\mathbb{U} \neq \mathbb{V}$ and $\mathbb{U} \leq \mathbb{V}$, we say $\mathbb{U}$ is a strict refinement of $\mathbb{V}$, denoted by $\mathbb{U} < \mathbb{V}$, *i.e.*, $\mathbb{U} \leq \mathbb{V}$ and

$$\exists V \in \mathbb{V} \exists \mathbb{Z} \subseteq \mathbb{U} [|\mathbb{Z}| > 1, \uplus\mathbb{Z} = V].$$

If $\mathbb{U}$ is not a refinement of $\mathbb{V}$, we use the notation $\mathbb{U} \not\leq \mathbb{V}$, *i.e.*,

$$\exists V \in \mathbb{V} \forall \mathbb{Z} \subseteq \mathbb{U} [\uplus\mathbb{Z} \neq V].$$

Claim 3. *Basic statements*

- *(1st Statement)* $\mathbb{A} \leq \mathbb{B} \Leftrightarrow \forall A \in \mathbb{A} \exists B \in \mathbb{B}[A \subseteq B]$*;*
- *(2nd Statement)* $\mathbb{A} < \mathbb{B} \Leftrightarrow [\mathbb{A} \leq \mathbb{B}$ *and* $\exists A \in \mathbb{A} \exists B \in \mathbb{B}(A \subsetneq B)]$.
- *(3rd Statement)* $\mathbb{A} \leq \mathbb{B} \Rightarrow [|\mathbb{A}| \geq |\mathbb{B}|$ *and* $(\mathbb{A} < \mathbb{B} \Rightarrow |\mathbb{A}| > |\mathbb{B}|)]$.

Proof. For the first statement: Let $A \in \mathbb{A}$ be arbitrary. Then $\exists B \in \mathbb{B}$ s.t. $A \cap B \neq \emptyset$. By $\mathbb{A} \leq \mathbb{B}$, it then follows $\exists \mathbb{Z} \subseteq \mathbb{A}$ such that $\uplus \mathbb{Z} = B$, *i.e.*, $A \cap \uplus \mathbb{Z} \neq \emptyset$, *i.e.*, $A \subseteq \uplus \mathbb{Z} = B$. On the other hand, assume

$$\forall A \in \mathbb{A} \exists B \in \mathbb{B}[A \subseteq B]. \tag{1}$$

Let $B \in \mathbb{B}$ be arbitrary. Let

$$\mathbb{Z} = \{A \in \mathbb{A} : A \cap B \neq \emptyset\} \subseteq \mathbb{A}.$$

Obviously, $\mathbb{Z} \neq \emptyset$ and by Eq. (1)

$$\forall A \in \mathbb{Z} \exists B_A \in \mathbb{B}[A \subseteq B_A], \tag{2}$$

i.e., $B_A \cap B \neq \emptyset$, *i.e.*, $\forall A \in \mathbb{A}[B_A = B]$, *i.e.*, by Eq. (2)

$$\forall A \in \mathbb{Z}[A \subseteq B],$$

i.e., by the definition $\uplus \mathbb{Z} = B$. For the second statement: Since $\mathbb{A} < \mathbb{B}$,

$$\exists B \in \mathbb{B} \exists \mathbb{Z} \subseteq \mathbb{A}[|\mathbb{Z}| > 1, \uplus \mathbb{Z} = B],$$

i.e., $\exists A \in \mathbb{A}[A \subsetneq B]$. On the other hand, choose $A \in \mathbb{A}, B \in \mathbb{B}$ such that $A \subsetneq B$. Let $\mathbb{Z} = \{A \in \mathbb{A} : A \cap B \neq \emptyset\} \subseteq \mathbb{A}$. Then $|\mathbb{Z}| > 1$ and $\uplus \mathbb{Z} = B$. For the third statement: It follows immediately from the above results and the definition. □

Claim 4. $\leq$ *is a partial ordering on* $PT(S)$.

Proof. It follows directly from Claim 3. □

Claim 5. *1.* $\mathbb{A} \sqcap \mathbb{B} \leq \mathbb{A}, \mathbb{B}$ *and* $|\mathbb{A} \sqcap \mathbb{B}| \geq |\mathbb{A}|, |\mathbb{B}|$;

2. $\mathbb{A} \neq \mathbb{B} \Leftrightarrow [\mathbb{A} \sqcap \mathbb{B} < \mathbb{A}$ *or* $\mathbb{A} \sqcap \mathbb{B} < \mathbb{B}]$;

3. $\mathbb{A} = \mathbb{B} \Leftrightarrow [\mathbb{A} \sqcap \mathbb{B} = \mathbb{A}$ *and* $\mathbb{A} \sqcap \mathbb{B} = \mathbb{B}]$;

4. $\mathbb{A} \neq \mathbb{B} \Leftrightarrow [|\mathbb{A} \sqcap \mathbb{B}| > |\mathbb{A}|$ *or* $|\mathbb{A} \sqcap \mathbb{B}| > |\mathbb{B}|]$;

5. $\mathbb{A} = \mathbb{B} \Leftrightarrow [|\mathbb{A} \sqcap \mathbb{B}| = |\mathbb{A}| = |\mathbb{B}|]$.

Proof. Firstly, by Corollary 1, $\mathbb{A} \sqcap \mathbb{B} \in PT(S)$ and then by Claim 3, the results follow. Secondly, let $\mathbb{A} \neq \mathbb{B}$. Then $\exists A \in \mathbb{A} \exists B \in \mathbb{B}$ s.t. $A \cap B \subsetneq A$ or $A \cap B \subsetneq B$, *i.e.*, by Claim 3, $\mathbb{A} \sqcap \mathbb{B} < \mathbb{A}$ or $\mathbb{A} \sqcap \mathbb{B} < \mathbb{B}$. For the third statement, it follows immediately from the first and second ones. The fourth and fifth ones come from Claim 3. □

Corollary 2. *1.* $|\mathbb{A} \sqcap \mathbb{B}| \geq max\{|\mathbb{A}|, |\mathbb{B}|\}$;

2. $\mathbb{A} \neq \mathbb{B}$ *iff* $|\mathbb{A} \sqcap \mathbb{B}| > min\{|\mathbb{A}|, |\mathbb{B}|\}$.

Proof. Both statements follow immediately from Claim 5. □

Claim 6. *For any* $\mathbb{S} \subseteq \mathbb{B}[\mathbb{A} \sqcap \mathbb{S} \subseteq \mathbb{A} \sqcap \mathbb{B}]$*, in particular,* $\mathbb{S} = \{B\}$ *for any* $B \in \mathbb{B}$.

Proof. It follows immediately from the definition. □

Claim 7. *If* $\mathbb{A} \leq \mathbb{B}, A \in \mathbb{A}, B \in \mathbb{B}$ *and* $A \cap B \neq \emptyset$*, then* $A \subseteq B$.

Proof. By the definition, $\exists \mathbb{Z} \subseteq \mathbb{A}$ s.t. $\uplus \mathbb{Z} = B$, *i.e.*, $A \cap (\uplus \mathbb{Z}) \neq \emptyset$, *i.e.*, $A \subseteq \uplus \mathbb{Z} = B$. □

3. SETTINGS OF A METRIC

In this section, we devise a metric which measures the distance between two partitions. This metric is well equipped with several characteristics:

- It would investigate the subtle differences between the two partitions via three non-negative real functions τ, σ and ρ. The partitions or their derived subsets are regarded as nodes, while τ, σ and ρ are deemed as weighting functions for the nodes or the edges.
- It derives a tree-like structure distance. The distance between partitions are defined by the assigned weights for their compartments, in which the assigned weights are assigned by the subsets of these compartments.
- The three-level weighting approaches could also be easily reduced to two-level, which is the typical method in devising distance functions.
- The three-level structures are easy to be linked to other probabilistic or fuzzy reasoning.

Suppose $\tau : P(S) \to \mathbb{R}^+$ is a non-negative real function with a property that $\tau(K) = 0$ if and only if K is an empty set. This function could play the role in assigning weights or probabilities for the nodes, which reflect the structure of the partitions.

Let $\mathbb{S}, \mathbb{T} \subseteq \mathbb{A}$ be arbitrary. Based on the hierarchical structure defined via $\subseteq$ and $\in$, we have the following functions.

Definition 3.1. (σ function) Define $\sigma : P(S) \to \mathbb{R}^+$ by $\sigma(A) := \sum_{H \in P(A)} \tau(H)$.

Definition 3.2. (ρ function) Define $\rho : PP(S) \to \mathbb{R}^+$ by $\rho(\mathbb{U}) := \sum_{U \in \mathbb{U}} \sigma(U)$.

Example 2. Suppose $\mathscr{A} = \{A_1, A_2, ..., A_n\} \subseteq P(S)$, *i.e.*, each A_j is a subset of S. Suppose the power set of each A_j, or $P(A_j) = \{H_{j1}, H_{j2}, ..., H_{jm_j}\} \subseteq A_j$, *i.e.*, each H_{jk} is a subset of A_j for all $j \in \{1, 2, ..., n\}$.

Then the hierarchical structure of these functions could be illustrated in Fig. 1.

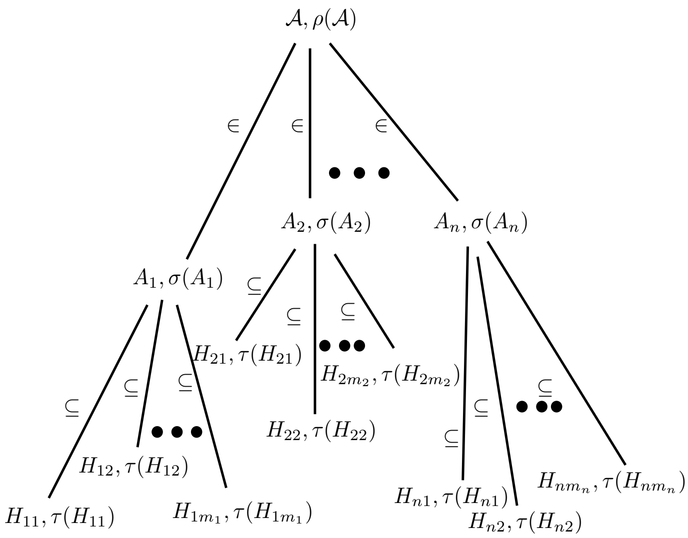

Fig. 1. τ, σ and ρ functions.

Definition 3.3. Define a $\mathbb{U}-$partition-induced subsets in total (or $PST(\mathbb{U})$) by $PST(\mathbb{U}) := \cup\{P(U) : U \in \mathbb{U}\}$, *i.e.*, $PST(\mathbb{U})$ is the union of all the subsets of the compartments in $\mathbb{U}$.

Claim 8. *1.* $\forall A, A' \in \mathbb{A}[A \neq A' \Leftrightarrow P(A) \cap P(A') = \emptyset]$;

2. $PST(\mathbb{T}) = \biguplus_{V \in \mathbb{T}} P(V)$.

Proof. Both follow immediately from the definition. □

Claim 9. *1.* $\rho(\mathbb{T}) = \sum_{H \in PST(\mathbb{T})} \tau(H)$;

2. *(monotonicity)* $\mathbb{U} \subseteq \mathbb{V} \Rightarrow PST(\mathbb{U}) \subseteq PST(\mathbb{V})$;

3. *(monotonicity)* $PST(\mathbb{A}) \subseteq PST(\mathbb{B}) \Rightarrow \rho(\mathbb{A}) \leq \rho(\mathbb{B})$.

Proof. By the definitions and Claim 8, the first statement holds via

$$\rho(\mathbb{T}) = \sum_{T \in \mathbb{T}} \sigma(T) = \sum_{T \in \mathbb{T}} \sum_{H \in P(T)} \tau(H) = \sum_{H \in PST(\mathbb{T})} \tau(H).$$

For the second statement, it follows immediately from the definition. For the third statement, it follows immediately from the first statement. □

Claim 10. *If* $\mathbb{K}, \mathbb{H} \subseteq \mathbb{A}$ *and* $\mathbb{K} \cap \mathbb{H} = \emptyset$*, then*

1. $PST(\mathbb{K} \uplus \mathbb{H}) = PST(\mathbb{K}) \uplus PST(\mathbb{H})$;

2. $\rho(\mathbb{K} \uplus \mathbb{H}) = \rho(\mathbb{K}) + \rho(\mathbb{H})$.

Proof. For the first statement: Let $U \in PST(\mathbb{K} \uplus \mathbb{H})$. Then $\exists V \in \mathbb{K} \uplus \mathbb{H}$ such that $U \in P(V)$. Since $\mathbb{K} \cap \mathbb{H} = \emptyset$, it follows $V \in \mathbb{K}$ or $V \in \mathbb{H}$, *i.e.*,

$$U \in PST(\mathbb{K}) \cup PST(\mathbb{H}).$$

Therefore, we have shown $PST(\mathbb{K} \uplus \mathbb{H}) \subseteq PST(\mathbb{K}) \cup PST(\mathbb{H})$. With this and Claim 9, it follows

$$PST(\mathbb{K} \uplus \mathbb{H}) = PST(\mathbb{K}) \cup PST(\mathbb{H}).$$

Next, we show $PST(\mathbb{K}) \cap PST(\mathbb{H}) = \emptyset$. Suppose $PST(\mathbb{K}) \cap PST(\mathbb{H}) \neq \emptyset$. Let $W \in PST(\mathbb{K}) \cap PST(\mathbb{H})$ be arbitrary. By definition,

$$\exists K \in \mathbb{K} \subseteq \mathbb{A}, H \in \mathbb{H} \subseteq \mathbb{A} \text{ s.t. } W \in P(K), W \in P(H),$$

i.e., $W \subseteq K \in \mathbb{A}$ and $W \subseteq H \in \mathbb{A}$, *i.e.*, $K \cap H \neq \emptyset$. Hence $K = H$, *i.e.*, $\mathbb{K} \cap \mathbb{H} \neq \emptyset$, a contradiction. For the second statement: By the first statement and Claim 9, one has $\rho(\mathbb{K} \uplus \mathbb{H}) =$

$$\sum_{H \in PST(\mathbb{K} \uplus \mathbb{H})} \tau(H) = \sum_{H \in PST(\mathbb{K})} \tau(H) + \sum_{H \in PST(\mathbb{H})} \tau(H) = \rho(\mathbb{K}) + \rho(\mathbb{H}).$$

□

In the following, we define a distance function d on $PT(S)$.

Definition 3.4. (distance function) Define the distance between $\mathbb{A}$ and $\mathbb{B}$ by

$$d(\mathbb{A}, \mathbb{B}) := \frac{1}{2} \cdot [\rho(\mathbb{A}) + \rho(\mathbb{B})] - \rho(\mathbb{A} \sqcap \mathbb{B}).$$

From the definitions, we know the distance depends also on the function τ.

Claim 11. *Suppose* $\mathbb{A} = \uplus\{A_i | 1 \leq i \leq m\}$, $\mathbb{B} = \uplus\{B_i | 1 \leq i \leq n\}$ *and* $\mathbb{A} \sqcap \mathbb{B} = \uplus\{D_i | 1 \leq i \leq v\}$ *for some* $v \in \mathbb{N}$. *If* $\tau(H) = |H|$ *for all* $H \in P(S)$, *then* $d(\mathbb{A}, \mathbb{B}) = \dfrac{\sum_{j=1}^{m}\sum_{k=0}^{|A_j|} |A_j| \cdot 2^{|A_j|-1} + \sum_{j=1}^{n}\sum_{k=0}^{|B_j|} |B_j| \cdot 2^{|B_j|-1}}{2} - \sum_{j=1}^{v}\sum_{k=0}^{|D_j|} |D_j| \cdot 2^{|D_j|-1}.$

Proof. Let C be the notation for combination. By the definition,
$d(\mathbb{A}, \mathbb{B}) = \dfrac{\sum_{j=1}^{m}\sum_{k=0}^{|A_j|} |A_j| \cdot C^k_{|A_j|} + \sum_{j=1}^{n}\sum_{k=0}^{|B_j|} |B_j| \cdot C^k_{|B_j|}}{2} - \sum_{j=1}^{v}\sum_{k=0}^{|D_j|} |D_j| \cdot C^k_{|D_j|}$, and the fact that $\sum_{i=0}^{m} i \cdot \binom{m}{i} = m \cdot 2^{m-1}$ and $\tau(H) = |H|$, the result follows. □

Claim 12. *(equality) If* $\mathbb{U}$ *is a set of disjoint non-empty sets and* $\mathbb{V}$ *is a disjoint non-empty sets, then*

$$PST(\mathbb{U}) = PST(\mathbb{V}) \Leftrightarrow \mathbb{U} = \mathbb{V}.$$

Proof. Let $U \in \mathbb{U}$ be arbitrary. Then by the definitions,

$$U \in P(U) \subseteq PST(\mathbb{U}) = PST(\mathbb{V}),$$

i.e., $U \in PST(\mathbb{V})$, *i.e.*,

$$\exists V \in \mathbb{V} \text{ s.t. } U \in P(V),$$

i.e., $U \subseteq V$. Similarly $\exists \tilde{U} \in \mathbb{U}$ s.t. $V \subseteq \tilde{U}$, *i.e.*, $U \subseteq V \subseteq \tilde{U}$. Hence

$$V = \tilde{U} = U \in \mathbb{V}.$$

Therefore, we have shown $\mathbb{U} \subseteq \mathbb{V}$ and by the same procedure, $\mathbb{V} \subseteq \mathbb{U}$. This completes our proof. □

Claim 13. *(monotonicity)*

1. $\mathbb{A} \leq \mathbb{B} \Leftrightarrow [PST(\mathbb{A}) \subseteq PST(\mathbb{B})]$;
2. $\mathbb{A} \leq \mathbb{B} \Rightarrow \rho(\mathbb{A}) \leq \rho(\mathbb{B})$;
3. $\mathbb{A} < \mathbb{B} \Rightarrow [PST(\mathbb{A}) \subsetneq PST(\mathbb{B})$ *and* $\rho(\mathbb{A}) < \rho(\mathbb{B})]$.

Proof. For the first statement: Let $H \in PST(\mathbb{A})$ be arbitrary. By Definition 3.3, $\exists A \in [H \in P(A)]$. Since $\exists B \in \mathbb{B}$ such that $A \cap B \neq \emptyset$, by Claim 7 $H \subseteq A \subseteq B$, *i.e.*, $H \in P(B)$, *i.e.*, $H \in PST(\mathbb{B})$. Hence we have shown

$$\forall A \in \mathbb{A} \exists B \in \mathbb{B}[A \subseteq B],$$

which by Claim 3 yields $\mathbb{A} \leq \mathbb{B}$. On the other hand, let $A \in \mathbb{A}$ be arbitrary. Then by the definition

$$P(A) \subseteq PST(\mathbb{A}) \subseteq PST(\mathbb{B}),$$

i.e., $A \in PST(\mathbb{B})$, *i.e.*, by the definition $\exists B \in \mathbb{B}$ s.t. $A \in P(B)$, *i.e.*, $A \subseteq B$. For the second statement: $\rho(\mathbb{A}) \leq \rho(\mathbb{B})$ follows immediately from the first statement and Claim 9. For the third statement: By the first and second statements and Claim 12, the results follow. □

Claim 14. *For any* $B, B' \in \mathbb{B}$

$$(\mathbb{A} \sqcap \{B\}) \cap (\mathbb{A} \sqcap \{B'\}) \neq \emptyset \textit{ iff } B = B'.$$

Proof. Let

$$K \in (\mathbb{A} \sqcap \{B\}) \cap (\mathbb{A} \sqcap \{B'\})$$

be arbitrary. Then $\exists A, \tilde{A} \in \mathbb{A}$ s.t. $K = A \cap B = \tilde{A} \cap B'$, *i.e.*, $B \cap B' \neq \emptyset$, *i.e.*, $B = B'$. □

Lemma 3.1. 1. $\mathbb{A} \sqcap \mathbb{B} = \biguplus_{B \in \mathbb{B}} (\mathbb{A} \sqcap \{B\})$;

2. $\rho(\mathbb{A}\sqcap\mathbb{B}) = \sum_{B\in\mathbb{B}} \rho(\mathbb{A}\sqcap\{B\})$ *and* $\rho(\mathbb{B}\sqcap\mathbb{C}) = \sum_{B\in\mathbb{B}} \rho(\mathbb{C}\sqcap\{B\})$;
3. $\rho(\mathbb{B}) = \sum_{B\in\mathbb{B}} \rho(\{B\})$;
4. $\rho(\mathbb{A}\sqcap\mathbb{B}\sqcap\mathbb{C}) = \sum_{B\in\mathbb{B}} \rho(\mathbb{A}\sqcap\{B\}\sqcap\mathbb{C})$.

Proof. For the first statement: $\mathbb{A}\sqcap\mathbb{B} = \biguplus_{B\in\mathbb{B}}(\mathbb{A}\sqcap\{B\})$ follows immediately from Claim 6 and 14. For the second statement: We show the front one. By the first statement and Claim 10, it follows

$$\rho(\mathbb{A}\sqcap\mathbb{B}) = \rho(\biguplus_{B\in\mathbb{B}}(\mathbb{A}\sqcap\{B\})) = \sum_{B\in\mathbb{B}} \rho(\mathbb{A}\sqcap\{B\}).$$

For the third statement: By Claim 10

$$\rho(\mathbb{B}) = \rho(\biguplus_{B\in\mathbb{B}}\{B\}) = \sum_{B\in\mathbb{B}} \rho(\{B\}).$$

The fourth statement follows immediately from the above statements. □

Given $B\in\mathbb{B}$, we use K_{ij} to denote the set $A_i\cap B\cap C_j$, whenever $A_i\in\mathbb{A}$ and $C_j\in\mathbb{C}$.

Remark 1. If $K_{ij}\neq\emptyset$ and $K_{i'j'}\neq\emptyset$, then one has $K_{ij}\cap K_{i'j'}\neq\emptyset$ if and only $i=i', j=j'$. Furthermore, $P(K_{ij})\cap P(K_{i'j'})\neq\emptyset$ if and only if $i=i'$ and $j=j'$.

Based on Remark 1 and Claim 10, we have the following proofs.

Claim 15. *1.* $PST(\mathbb{A}\sqcap\{B\}) = \biguplus_{i=1}^{|\mathbb{A}|} P(\biguplus_{j=1}^{|\mathbb{C}|} K_{ij})$;

2. $PST(\mathbb{C}\sqcap\{B\}) = \biguplus_{j=1}^{|\mathbb{C}|} P(\biguplus_{i=1}^{|\mathbb{A}|} K_{ij})$;

3. $PST(\{B\}) = P(\biguplus_{i=1}^{|\mathbb{A}|}\biguplus_{j=1}^{|\mathbb{C}|} K_{ij})$;

4. $PST(\mathbb{A}\sqcap\{B\}\sqcap\mathbb{C}) = \biguplus_{i=1}^{|\mathbb{A}|}\biguplus_{j=1}^{|\mathbb{C}|} P(K_{ij})$.

Proof. The first statement follows immediately from the following inferences:

$$\begin{aligned}
&H\in PST(\mathbb{A}\sqcap\{B\})\\
&\Leftrightarrow \exists A_i\in\mathbb{A} \text{ s.t. } H\in P(A_i\cap B)\\
&\Leftrightarrow \exists A_i\in\mathbb{A} \text{ s.t. } H\subseteq A_i\cap B\subseteq A_i\cap B\cap S = A_i\cap B\cap(\uplus\mathbb{C})\\
&= \biguplus_{j=1}^{|\mathbb{C}|}(A_i\cap B\cap C_j).
\end{aligned}$$

This leads to the necessary and sufficient conditions that

$$\exists A_i \in \mathbb{A} \text{ s.t. } H \in P(\biguplus_{j=1}^{|\mathbb{C}|}(A_i \cap B \cap C_j)) \subseteq \biguplus_{i=1}^{|\mathbb{A}|} P(\biguplus_{j=1}^{|\mathbb{C}|}(A_i \cap B \cap C_j))$$
$$= \biguplus_{i=1}^{|\mathbb{A}|} P(\biguplus_{j=1}^{|\mathbb{C}|} K_{ij}).$$

Similarly, one could show the second statement. The third statement follows from the following inferences:

$$P(B) = P(B \cap S \cap S) = P(B \cap (\uplus\mathbb{A}) \cap (\uplus\mathbb{C}))$$
$$= P(B \cap (\biguplus_{i=1}^{|\mathbb{A}|} A_i) \cap (\biguplus_{j=1}^{|\mathbb{C}|} C_j)) = P(\biguplus_{i=1}^{|\mathbb{A}|}\biguplus_{j=1}^{|\mathbb{C}|} A_i \cap B \cap C_j)$$
$$= P(\biguplus_{i=1}^{|\mathbb{A}|}\biguplus_{j=1}^{|\mathbb{C}|} K_{ij}).$$

The fourth statement follows from the following inferences:

$$H \in PST(\mathbb{A} \sqcap \{B\} \sqcap \mathbb{C})$$
$$\Leftrightarrow H \in P(A_i \cap B \cap C_j) = P(K_{ij}) \text{ for some } A_i \in \mathbb{A}, C_j \in \mathbb{C}$$
$$\subseteq \biguplus_{i=1}^{|\mathbb{A}|}\biguplus_{j=1}^{|\mathbb{C}|} P(K_{ij}).$$

□

Claim 16. $P(\biguplus_{j=1}^{|\mathbb{C}|} A_i \cap B \cap C_j) \cap PST(\mathbb{C} \sqcap \{B\}) = \biguplus_{j=1}^{|\mathbb{C}|} P(A_i \cap B \cap C_j)$ *for all* $A_i \in \mathbb{A}$.

Proof. Let $A_i \in \mathbb{A}$ be arbitrary. Then the statement follows from the following inferences:

$$H \in P(\biguplus_{j=1}^{|\mathbb{C}|} A_i \cap B \cap C_j) \cap PST(\mathbb{C} \sqcap \{B\})$$
$$\Leftrightarrow H \subseteq \biguplus_{j=1}^{|\mathbb{C}|} A_i \cap B \cap C_j, H \in PST(\mathbb{C} \sqcap \{B\})$$
$$\Leftrightarrow H \subseteq A_i \cap B \cap (\biguplus_{j=1}^{|\mathbb{C}|} C_j) = A_i \cap B \cap S = A_i \cap B \text{ and}$$
$$H \subseteq C_k \cap B \text{ for some } C_k \in \mathbb{C}.$$
$$\Leftrightarrow H \subseteq A_i \cap B \cap C_k \text{ for some } C_k \in \mathbb{C}.$$
$$\Leftrightarrow H \in P(A_i \cap B \cap C_k) \text{ for some } C_k \in \mathbb{C}$$
$$\subseteq \biguplus_{j=1}^{|\mathbb{C}|} P(A_i \cap B \cap C_j).$$

Claim 17. $PST(\mathbb{A}\sqcap\{B\})\cap PST(\mathbb{C}\sqcap\{B\}) = PST(\mathbb{A}\sqcap\{B\}\sqcap\mathbb{C})$. □

Proof. It follows immediately from the following inferences:

$$\begin{aligned}
&PST(\mathbb{A}\sqcap\{B\})\cap PST(\mathbb{C}\sqcap\{B\})\\
&=\biguplus_{i=1}^{|\mathbb{A}|} P(\biguplus_{j=1}^{|\mathbb{C}|} A_i\cap B\cap C_j)\cap PST(\mathbb{C}\sqcap\{B\})(\text{ by Claim 15})\\
&=\biguplus_{i=1}^{|\mathbb{A}|} [P(\biguplus_{j=1}^{|\mathbb{C}|} A_i\cap B\cap C_j)\cap PST(\mathbb{C}\sqcap\{B\})]\\
&=\biguplus_{i=1}^{|\mathbb{A}|}\biguplus_{j=1}^{|\mathbb{C}|} P(A_i\cap B\cap C_j)(\text{ by Claim 16 })\\
&=PST(\mathbb{A}\sqcap\{B\}\sqcap\mathbb{C})(\text{ by Claim 15})
\end{aligned}$$

Lemma 3.2. *1.* $d(\mathbb{A},\mathbb{B})\geq 0$*;* □

2. $d(\mathbb{A},\mathbb{B})=0$ *if and only if* $\mathbb{A}=\mathbb{B}$*;*

3. $d(\mathbb{A},\mathbb{B})=d(\mathbb{B},\mathbb{A})$.

Proof. For the first two statements: If $\mathbb{A}=\mathbb{B}$, then by the definition of d and Claim 1, it follows $d(\mathbb{A},\mathbb{B})=0$. If $\mathbb{A}\neq\mathbb{B}$, then by Claim 5, $\mathbb{A}\sqcap\mathbb{B}<\mathbb{A}$ or $\mathbb{A}\sqcap\mathbb{B}<\mathbb{B}$. By Claim 13,

$$\rho(\mathbb{A}\sqcap\mathbb{B})<\rho(\mathbb{A})\text{ or }\rho(\mathbb{A}\sqcap\mathbb{B})<\rho(\mathbb{B}).$$

Therefore, one has

$$2\cdot\rho(\mathbb{A}\sqcap\mathbb{B})<\rho(\mathbb{A})+\rho(\mathbb{B}),$$

and thus $d(\mathbb{A},\mathbb{B})>0$. For the third statement: it follows immediately from the definition of d and Claim 1. □

Claim 18. *For all* $B\in\mathbb{B}$

$$\rho(\{B\})+\rho(\mathbb{A}\sqcap\{B\}\sqcap\mathbb{C})-\rho(\mathbb{A}\sqcap\{B\})-\rho(\mathbb{C}\sqcap\{B\})\geq 0.$$

Proof. By Claim 9 and 15, we have the following inferences:

$$\begin{aligned}
&\rho(\{B\})+\rho(\mathbb{A}\sqcap\{B\}\sqcap\mathbb{C})-\rho(\mathbb{A}\sqcap\{B\})-\rho(\mathbb{C}\sqcap\{B\})\\
&=\sum_{H\in PST(\{B\})}\tau(H)+\sum_{H\in PST(\mathbb{A}\sqcap\{B\}\sqcap\mathbb{C})}\tau(H)\\
&\qquad-\sum_{H\in PST(\mathbb{A}\sqcap\{B\})}\tau(H)-\sum_{H\in PST(\mathbb{C}\sqcap\{B\})}\tau(H). \quad (3)
\end{aligned}$$

Since

$$\mathbb{A}\sqcap\{B\}\sqcap\mathbb{C}\subseteq\mathbb{A}\sqcap\{B\}\subseteq\{B\},$$

$$\mathbb{A} \sqcap \{B\} \sqcap \mathbb{C} \subseteq \mathbb{C} \sqcap \{B\} \subseteq \{B\},$$

by Claim 9,

$$\begin{aligned} &PST(\mathbb{A} \sqcap \{B\} \sqcap \mathbb{C}) \subseteq PST(\mathbb{A} \sqcap \{B\}) \subseteq PST(\{B\}), \\ &\qquad PST(\mathbb{A} \sqcap \{B\} \sqcap \mathbb{C}) \subseteq PST(\mathbb{C} \sqcap \{B\}) \subseteq PST(\{B\}). \end{aligned} \tag{4}$$

Now define

$$S_1 = PST(\mathbb{A} \sqcap \{B\}) - PST(\mathbb{A} \sqcap \{B\} \sqcap \mathbb{C}),$$

$$S_2 = PST(\mathbb{C} \sqcap \{B\}) - PST(\mathbb{A} \sqcap \{B\} \sqcap \mathbb{C}).$$

Then $S_1, S_2 \subseteq PST(\{B\})$. Moreover, by Claim 17, $S_1 \cap S_2 = \emptyset$, *i.e.*,

$$S_1 \uplus S_2 \uplus PST(\mathbb{A} \sqcap \{B\} \sqcap \mathbb{C}) \subseteq PST(\{B\}). \tag{5}$$

Hence by Eqs. (3), (4) and (5)

$$\begin{aligned} &\rho(\{B\}) + \rho(\mathbb{A} \sqcap \{B\} \sqcap \mathbb{C}) - \rho(\mathbb{A} \sqcap \{B\}) - \rho(\mathbb{C} \sqcap \{B\}) \\ &\geq \sum_{H \in S_1 \uplus S_2 \uplus PST(\mathbb{A} \sqcap \{B\} \sqcap \mathbb{C})} \tau(H) + \sum_{H \in PST(\mathbb{A} \sqcap \{B\} \sqcap \mathbb{C})} \tau(H) \\ &- \sum_{H \in S_1 \uplus PST(\mathbb{A} \sqcap \{B\} \sqcap \mathbb{C})} \tau(H) - \sum_{H \in S_2 \uplus PST(\mathbb{A} \sqcap \{B\} \mathbb{C})} \tau(H) \\ &= 0. \end{aligned}$$

□

Lemma 3.3. $d(\mathbb{A}, \mathbb{B}) + d(\mathbb{B}, \mathbb{C}) \geq d(\mathbb{A}, \mathbb{C})$.

Proof. Since

$$\begin{aligned} &d(\mathbb{A}, \mathbb{B}) + d(\mathbb{B}, \mathbb{C}) - d(\mathbb{A}, \mathbb{C}) \\ &= \rho(\mathbb{B}) + \rho(\mathbb{A} \sqcap \mathbb{C}) - \rho(\mathbb{A} \sqcap \mathbb{B}) - \rho(\mathbb{B} \sqcap \mathbb{C}), \text{ by Definition 3.4} \\ &\geq \rho(\mathbb{B}) + \rho(\mathbb{A} \sqcap \mathbb{B} \sqcap \mathbb{C}) - \rho(\mathbb{A} \sqcap \mathbb{B}) - \rho(\mathbb{B} \sqcap \mathbb{C}), \text{ by Claim 13} \\ &= \sum_{B \in \mathbb{B}} [\rho(\{B\}) + \rho(\mathbb{A} \sqcap \{B\} \sqcap \mathbb{C}) - \rho(\mathbb{A} \sqcap \{B\}) - \rho(\mathbb{C} \sqcap \{B\})], \end{aligned}$$

by Lemma 3.1. it suffices to show for all $B \in \mathbb{B}$

$$\rho(\{B\}) + \rho(\mathbb{A} \sqcap \{B\} \sqcap \mathbb{C}) - \rho(\mathbb{A} \sqcap \{B\}) - \rho(\mathbb{C} \sqcap \{B\}) \geq 0.$$

It holds via Claim 18 and this completes our proof. □

Theorem 3.4. *$(PT(S), d)$ is a metric space.*

Proof. It follows immediately from Lemmas 3.2 and 3.3. □

4. APPLICATION ON PROBLEM CLASSIFICATION

In this section, we apply the metric of equivalence classes on rough set theory. Suppose the universe $U = \{alg_1, alg_2, \cdots, alg_{12}\}$, where alg_k indicates the kth given algorithm. The attributes are defined by three problems which are prepared largely for the algorithms. The attribute values for Problem 1 are correct, wrong and N/A, which stands for the corresponding algorithms for the problem are correctly solved, wrongly solved and not available (namely, can't be solved by the algorithm), respectively. The problem 2 is solvable by all the algorithms, but with various handling speed. The attribute values for Problem 2 are slow, medium and fast, which indicate how efficient the corresponding algorithms are in solving Problem 2. Problem 3 is implementable by computer programs. The attribute values for Problem 3 are C, C++, JAVA and PYTHON. Induced by the problems as attributes and the attributes as equivalence relations, the information system is presented in Table 2. The partitions are

Table 2. Information system.

U	$Prob_1$	$Prob_2$	$Prob_3$
alg_1	correct	slow	C
alg_2	correct	slow	C++
alg_3	wrong	medium	C
alg_4	N/A	fast	JAVA
alg_5	correct	medium	PYTHON
alg_6	N/A	fast	C
alg_7	N/A	fast	JAVA
alg_8	wrong	slow	C++
alg_9	correct	medium	JAVA
alg_{10}	wrong	slow	PYTHON
alg_{11}	correct	fast	C
alg_{12}	wrong	fast	C

- $U/Prob_1 = \{[correct] = \{alg_1, alg_2, alg_5, alg_9, alg_{11}\}, [wrong] = \{alg_3, alg_8, alg_{10}, alg_{12}\}, [N/A] = \{alg_4, alg_6, alg_7\}\}$;

- $U/Prob_2 = \{[slow] = \{alg_1, alg_2, alg_8, alg_{10}\}, [medium] = \{alg_3, alg_5, alg_9\}, , [fast] = \{alg_4, alg_6, alg_7, alg_{11}, alg_{12}\}\}$;

- $U/Prob_3 = \{[C] = \{alg_1, alg_3, alg_6, alg_{11}, alg_{12}\}, [C{+}{+}] = \{alg_2, alg_8\}, [JAVA] = \{alg_4, alg_7, alg_9\}, [PYTHON] = \{alg_5, alg_{10}\}\}$.

Furthermore,

- $U/Prob_1 \sqcap U/Prob_2 = \{[correct] \sqcap [slow] = \{alg_1, alg_2\}, [correct] \sqcap [medium] = \{alg_5, alg_9\}, [correct] \sqcap [fast] = \{alg_{11}\}, [wrong] \sqcap [slow] = \{alg_8, alg_{10}\}, [wrong] \sqcap [medium] = \{alg_3\}, [wrong] \sqcap [fast] = \{alg_{12}\}, [N/A] \sqcap [slow] = \emptyset, [N/A] \sqcap [medium] = \emptyset, [N/A] \sqcap [fast] = \{alg_4, alg_6, alg_7\}\}$;

- $U/Prob_1 \sqcap U/Prob_3 = \{[correct] \sqcap [C] = \{alg_1, alg_{11}\}, [correct] \sqcap [C++] = \{alg_2\}, [correct] \sqcap [JAVA] = \{alg_9\}, [correct] \sqcap [PYTHON] = \{alg_5\}, [wrong] \sqcap [C] = \{alg_3, alg_{12}\}, [wrong] \sqcap [C++] = \{alg_8\}, [wrong] \sqcap [JAVA] = \emptyset, [wrong] \sqcap [PYTHON] = \{alg_{10}\}, [N/A] \sqcap [C] = \{alg_6\}, [N/A] \sqcap [C++] = \emptyset, [N/A] \sqcap [JAVA] = \{alg_4, alg_7\}, [N/A] \sqcap [PYTHON] = \emptyset\}$;

- $U/Prob_2 \sqcap U/Prob_3 = \{[slow] \sqcap [C] = \{alg_1\}, [slow] \sqcap [C++] = \{alg_2, alg_8\}, [slow] \sqcap [JAVA] = \emptyset, [slow] \sqcap [PYTHON] = \{alg_{10}\}, [medium] \sqcap [C] = \{alg_3\}, [medium] \sqcap [C++] = \emptyset, [medium] \sqcap [JAVA] = \{alg_9\}, [medium] \sqcap [PYTHON] = \{alg_5\}, [fast] \sqcap [C] = \{alg_6, alg_{11}, alg_{12}\}, [fast] \sqcap [C++] = \emptyset, [fast] \sqcap [JAVA] = \{alg_4, alg_7\}, [fast] \sqcap [PYTHON] = \emptyset\}\}$.

Now we choose the $\tau : P(U) \to \mathbb{R}^+$ by $\tau(H) = 1$, which indicates the compatible algorithms within the same compartment shall enforce and strengthen the attribute with which the compartment is endowed. The more the combinations are, the higher the weights assigned to the compartment are. Take the compartment $[correct]$ for example. With our device, we could further consider the possible interaction between the algorithms $alg_1, alg_2, alg_5, alg_9$ and alg_{11}. Since they are in the same category with the same attribute, one shall expect the combination of the algorithms among them shall enhance the weights of the compartment. Unlike typical measurement for partition distance function, our device could further delve into the property of the compartments (or categories). Now we calculate the values for function σ:

- $\sigma([correct]) = 2^{|[correct]|} - 1 = 31; \sigma([wrong]) = 2^{|[wrong]|} - 1 = 31; \sigma([N/A]) = 2^{|[N/A]|} - 1 = 7; \sigma([slow]) = 2^{|[slow]|} - 1 = 31; \sigma([medium]) = 2^{|[medium]|} - 1 = 7; \sigma([fast]) = 2^{|[fast]|} - 1 = 31; \sigma([C]) = 2^{|[C]|} - 1 = 31; \sigma([C++]) = 2^{|[C++]|} - 1 = 3; \sigma([JAVA]) = 2^{|[JAVA]|} - 1 = 7; \sigma([PYTHON]) = 2^{|[PYTHON]|} - 1 = 3$;

- $\sigma([correct] \sqcap [slow]) = 3, \sigma([correct] \sqcap [medium]) = 3, \sigma([correct] \sqcap [fast]) = 1, \sigma([wrong] \sqcap [slow]) = 3, \sigma([wrong] \sqcap [medium]) = 1, \sigma([wrong] \sqcap [fast]) = 1, \sigma([N/A] \sqcap [slow]) = 0, \sigma([N/A] \sqcap [medium]) = 0, \sigma([N/A] \sqcap [fast]) = 7$;

- $\sigma([correct] \sqcap [C]) = 3, \sigma([correct] \sqcap [C++]) = 1, \sigma([correct] \sqcap [JAVA]) = 1, \sigma([correct] \sqcap [PYTHON]) = 1, \sigma([wrong] \sqcap [C]) = 3, \sigma([wrong] \sqcap [C++]) = 1, \sigma([wrong] \sqcap [JAVA]) = 0, \sigma([wrong] \sqcap [PYTHON]) = 1; \sigma([N/A] \sqcap [C]) = 1, \sigma([N/A] \sqcap [C++]) = 0, \sigma([N/A] \sqcap [JAVA]) = 3, \sigma([N/A] \sqcap [PYTHON]) = 0$;

- $\sigma([slow] \sqcap [C]) = 1, \sigma([slow] \sqcap [C++]) = 3, \sigma([slow] \sqcap [JAVA]) = 0, \sigma([slow] \sqcap [PYTHON]) = 1, \sigma([medium] \sqcap [C]) = 1, \sigma([medium] \sqcap [C++]) = 0, \sigma([medium] \sqcap [JAVA]) = 1, \sigma([medium] \sqcap [PYTHON]) = 1, \sigma([fast] \sqcap [C]) = 7, \sigma([fast] \sqcap [C++]) = 0, \sigma([fast] \sqcap [JAVA]) = 3, \sigma([fast] \sqcap [PYTHON]) = 0$.

Next, we calculate the values for ρ:

- $\rho(U/Prob1) = \sigma([correct]) + \sigma([wrong]) + \sigma([N/A]) = 69; \rho(U/Prob2) = \sigma([slow]) + \sigma([medium]) + \sigma([fast]) = 69; \rho(U/Prob3) = \sigma([C]) + \sigma([C++]) + \sigma([JAVA]) + \sigma([PYTHON]) = 44$;

- $\rho(U/Prob_1 \sqcap U/Prob_2) = \sigma([correct] \sqcap [slow]) + \sigma([correct] \sqcap [medium]) + \sigma([correct] \sqcap [fast]) + \sigma([wrong] \sqcap [slow]) + \sigma([wrong] \sqcap [medium]) + \sigma([wrong] \sqcap [fast]) + \sigma([N/A] \sqcap [slow]) + \sigma([N/A] \sqcap [medium]) + \sigma([N/A] \sqcap [fast]) = 19$;
- $\rho(U/Prob_1 \sqcap U/Prob_3) = \sigma([correct] \sqcap [C]) + \sigma([correct] \sqcap [C++]) + \sigma([correct] \sqcap [JAVA]) + \sigma([correct] \sqcap [PYTHON]) + \sigma([wrong] \sqcap [C]) + \sigma([wrong] \sqcap [C++]) + \sigma([wrong] \sqcap [JAVA]) + \sigma([wrong] \sqcap [PYTHON]) + \sigma([N/A] \sqcap [C]) + \sigma([N/A] \sqcap [C++]) + \sigma([N/A] \sqcap [JAVA]) + \sigma([N/A] \sqcap [PYTHON]) = 15$;
- $\rho(U/Prob_2 \sqcap U/Prob_3) = \sigma([slow] \sqcap [C]) + \sigma([slow] \sqcap [C++]) + \sigma([slow] \sqcap [JAVA]) + \sigma([slow] \sqcap [PYTHON]) + \sigma([medium] \sqcap [C]) + \sigma([medium] \sqcap [C++]) + \sigma([medium] \sqcap [JAVA]) + \sigma([medium] \sqcap [PYTHON]) + \sigma([fast] \sqcap [C]) + \sigma([fast] \sqcap [C++]) + \sigma([fast] \sqcap [JAVA]) + \sigma([fast] \sqcap [PYTHON] = 18$.

Finally, we calculate the distances between the partitions:

- $d(U/Prob_1, U/Prob_2) = \frac{\rho(U/Prob_1)+\rho(U/Prob_2)}{2} - \rho(U/Prob_1 \sqcap U/Prob_2) = 50$;
- $d(U/Prob_1, U/Prob_3) = \frac{\rho(U/Prob_1)+\rho(U/Prob_3)}{2} - \rho(U/Prob_1 \sqcap U/Prob_3) = 42.5$;
- $d(U/Prob_2, U/Prob_3) = \frac{\rho(U/Prob_2)+\rho(U/Prob_3)}{2} - \rho(U/Prob_2 \sqcap U/Prob_3) = 38.5$.

The conclusion is Problems 2 and 3 are much correlated.

Remark 2. In this application, we take the universe be a set of algorithms and the problems as the attributes. One could also redo the other way around by taking the universe of problems and the attributes of algorithms. In this case, one could further classify the algorithms, *i.e.*, attribute classification or form the clusters based on the hereditary metric, *i.e.*, attribute clustering. If the attributes are associated with a set of classifiers, one could also categorise these classifiers, or classifier categorisation.

Remark 3. We could also endow the three function τ, σ and ρ with probabilistic settings. Let Φ be the power set of U, but excluding empty set. This set represents all the possible combinations of the 12 algorithms. In order to study the probabilistic correlation between the attributes, we restrict their individual sample space to the maximal consistent equivalence classes induced by the types of problems, for example Problems 1 and 2. The maximal consistent combinations for Problem 1 is $PW[correct] \cup PW[wrong] \cup PW[N/A] - \{\emptyset\}$, while the ones for Problem 2 is $\Omega_2 = PW[slow] \cup PW[medium] \cup PW[fast] - \{\emptyset\}$, where PW denotes the power set operation. If τ is defined by $\frac{1}{|\Omega|}$, where $|\Omega| = |\Omega_1|$ for Problem 1, $|\Omega| = |\Omega_2|$ for Problem 2, and $|\Omega| = |\Omega_{12}|$ for the intersected structure (or partition) Ω_{12}. Then the distance function reveals the dissimilar distance between two probabilistic structures regarding Problems 1 and 2.

5. APPLICATION ON ONTOLOGY POPULATION PROCESS

Ontology population process is an interesting and practical issues in information science [15, 16]. In this part, we run a real data [17] based analysis via our hereditary metric. Let the universe U be a collection of 14 ontology population systems (OPS). The attributes are various automatic ontology process procedures. The assigned attributes are the patterns (or parameters) applied within the attribute. The detailed description regarding the choice design of these ontology population systems are presented in the information system in Table 3. In the table, "att_1" represents "Type of concept instances"; "att_2" represents "Type of relation instances"; "att_3" represents "Domain dependency"; "att_4" represents "Consistency and redundancy checks"; "att_5" represents "Input documents type"; "att_6" represents "Expert intervention"; "att_7" represents "Used method". In the column att_2, 'all' means both taxonomic and non-taxonomic relations, including verb and noun relations. Now we intend to find out the correlation between these attributes, or ontology population process procedures.

Table 3. Information system for ontology population systems.

OPS	att_1	att_2	att_3	att_4
1 ≡ Hearst	Noun_phrase	Taxonomic	Independent	Not_solved
2 ≡ Group 1	Noun_phrases	All	Independent	Not_solved
3 ≡ Ibrahim	Noun_phrases	All	Dependent	Not_solved
4 ≡ Artequakt	Exact_entities	All_except_noun- based	Dependent	Redundancy_ only
5 ≡ Makki	Exact_entities	All_except _noun- based	Independent	Solved
6 ≡ Faria	Exact_entities	All_except _noun- based	Independent	Not_solved
7 ≡ SOBA	Exact_entities	All	Dependent	Redundancy_only
8 ≡ ISOLDE	Exact_entities	All	Independent	Solved
9 ≡ Web → KB	Web_pages	All	Web-centred	Not_solved
10 ≡ Group 2	Noun_phrases	All	Independent	Not_solved
11 ≡ Group 3	Exact_entities	All	Dependent	Not_solved
12 ≡ BOEMIE	Exact_entities	All	Independent	Solved
13 ≡ Group 4	Exact_entities	None	Independent	Redundancy_only
14 ≡ Yoon	Exact_entities	None	Independent	Redundancy_only
OPS	att_5	att_6	att_7	
1 ≡ Hearst	Unstructured_text	Automatic	Rule-based	
2 ≡ Group 1	Unstructured text	Semi-automatic	Rule-based	
3 ≡ Ibrahim	Unstructured_text	Semi-automatic	Rule-based	
4 ≡ Artequakt	Unstructured_text	Automatic	Rule-based	
5 ≡ Makki	Unstructured_text	Semi-automatic	Rule-based	
6 ≡ Faria	Unstructured_text	Automatic	Rule-based	
7 ≡ SOBA	Unstructured_text,	Automatic	Rule-based	
8 ≡ ISOLDE	Unstructured_text	Automatic	Rule-based	
9 ≡ Web → KB	Unstructured_text	Automatic	Machine_ learning	
10 ≡ Group 2	Unstructured_text	Automatic	Machine_learning	
11 ≡ Group 3	Unstructured_text	Automatic	Machine_learning	
12 ≡ BOEMIE	Heterogeneous	Semi-automatic	Hybrid	
13 ≡ Group 4	Unstructured_text	Automatic	Statistical	
14 ≡ Yoon	Only_structured	Semi-automatic	Statistical	

- $U/att_1 = \{[Noun\_phrase] = \{1,2,3,10\}, [Exact\_entities] = \{4,5,6,7,8,11,12,13,14\}, [Web\_pages] = \{9\}\}$;

- $U/att_2 = \{[Taoxonomic] = \{1\}, [All] = \{2,3,7,8,9,10,11,12\}, [All\_except\_noun-based] = \{4,5,6\}, [None] = \{13,14\}\}$;
- $U/att_3 = \{[Independent] = \{1,2,5,6,8,10,12,13,14\}, [Dependent] = \{3,4,7,11\}, [Web\_centred] = \{9\}\}$;
- $U/att_4 = \{[Not\_solved] = \{1,2,3,6,9,10,11\}, [Redundancy\_only] = \{4,7,13,14\}, [Solved] = \{5,8,12\}\}$;
- $U/att_5 = \{[Unstructured\_text] = \{1,2,3,4,5,6,7,8,9,10,11,13\}, [Heterogeneous] = \{12\}, [Only\_structured] = \{14\}\}$;
- $U/att_6 = \{[Automatic] = \{1,4,6,7,8,9,10,11,13\}, [Semi-auotmatic] = \{2,3,5,12,14\}\}$;
- $U/att_7 = \{[Rule-based] = \{1,2,3,4,5,6,7,8\}, [Machine\_learning] = \{9,10,11\}, [Hybrid] = \{12\}, [Statistical] = \{13,14\}\}$.

Moreover, the intersected partitions from the pairs of partitions are listed calculated and listed by their representing vector in Appendix titled "Pairwise intersected partitions". Now suppose the compartment weight is induced by the function $\tau : P(U) \to \mathbb{R}^+$, which is defined by $\tau(H) := |H|^2$. Then the values for function σ are calculated as follows:

- $\sigma([Noun\_phrase]) = \sum_{k=1}^{|[Noun\_phrase]|} k^2 \cdot \binom{|[Noun\_phrase]|}{k} = (|[Noun\_phrase]| + |[Noun\_phrase]|^2) \cdot 2^{|[Noun\_phrase]|-2} = 80$; similarly, $\sigma([Exact\_entities]) = 11520$; $\sigma([Web\_pages]) = 1$;
- $\sigma([Taoxonomic]) = 1$; $\sigma([All]) = 4608$; $\sigma([All\_except\_noun-based]) = 24$; $\sigma([None]) = 6$;
- $\sigma([Independent]) = 11520$; $\sigma([Ddependent]) = 80$; $\sigma([Web\_centred]) = 1$;
- $\sigma([Not\_solved]) = 1792$;$\sigma([Redundancy\_only]) = 80$;$\sigma([Solved]) = 24$;
- $\sigma([Unstructured_text]) = 159744$;$\sigma([Heterogenous]) = 1$;$\sigma([Only\_structured]) = 1$;
- $\sigma([Automatic]) = 11520$;$\sigma([Semi-automatic]) = 240$;
- $\sigma([Rule-based]) = 4608$;$\sigma([Machine_learning]) = 24$;$\sigma([Hybrid]) = 1$; $\sigma([Statistical]) = 6$.

As for the values for σ function over intersected partitions are listed in Appendix titled "Values for σ function over intersected partitions". Now the distances between ontology population process procedures are calculated and presented in Table 4. The most correlated pair among all the attributes is Attributes 2 and 4, and the second most correlated pair among all the attributes is Attributes 4 and 7. This indicates "Type of relation instances" is closely related to "Consistency and redundancy checks" in terms of level of automation for the ontology system. Therefore the information extraction engine regarding type of relation of instances is directly responsible for the consistency of data extraction and the soundness of knowledge base.

Table 4. Distance matrix for ontology population process procedures.

$d(U/att_i, U/att_j)$	U/att_1	U/att_2	U/att_3	U/att_4	U/att_5	U/att_6	U/att_7
U/att_1	0	7984	10879	6557.5	83798.5	10971.5	7846
U/att_2	7984	0	8001	**3010.5**	80372.5	7925.5	4503
U/att_3	10879	8001	0	6625.5	83798.5	11334.5	7846
U/att_4	6557.5	**3010.5**	6625.5	0	78997	6550	**3144.5**
U/att_5	83798.5	80372.5	83798.5	78997	0	74207	77557.5
U/att_6	10971.5	7925.5	11334.5	6550	74207	0	7908.5
U/att_7	7846	4503	7846	**3144.5**	77557.5	7908.5	0

6. CONCLUSIONS

Firstly, we have devised a metric that is suitable for measuring the distances between equivalence classes or the partitions, which are basic units in uncertain reasoning. Our device extends the concept of overlap between two sets. In uncertain reasoning, to identify or compare the differences between the units, one relies on the set intersection operations. Secondly, in the article, we give a complete proof for the distance functions. Distance functions are much more intuitive in explanation and interpretation of experimental or observational results. Thirdly, we show how to apply our metric on the real problems - in particular categorisation of problems or processing procedures. Lastly, based on these applications, one could find that this metric is flexible in coupling with other algorithms or methods. This offers us a much more efficient way tackling real problems, in particular those related to uncertain situations or environments. It is worth comparing our method with typical measurement of attribute dependency (MAD) in rough set theory:

- Our distance function measures mainly the correlation between attributes, while MAD measures the dependency or functionality of one attribute on the other;
- Distance function is a symmetric measurement, which is closer to the concept of similarity index, while MAD is an asymmetric measurement;
- Our distance function utilises three functions across three levels, which render some extra power in manipulating the weights on the compartments, while MAD utilises only two levels;
- The range of our distance function could be any positive real numbers, while MAD is normalised by the size of the universe and lies between 0 and 1;
- A larger value produced by our distance function indicates how uncorrelated two attributes are, while a larger value produced by MAD indicates a stronger dependency of one attribute on the other;
- Our distance function exploits mainly intersection operation, while MAD exploits intersection, union and subset operations.

When considering the correlation between attributes via our metric, one could further test whether the correlation is statically justifiable via non-parametric statistical inference.

One takes the product of partition space generated from the universe as the sample space and takes our method as a statistic to produce the range of the statics and a probability density function. Then one compares the empirical distance with the critical distance to obtain a hypothetical testing regarding the significance of the correlation between the attributes. However, one should be warned that the statistic per se depends on the functions τ, σ and ρ, and thus one should properly design the three functions to better reflect the realistic correlation between attributes. Another issue is such statistical testing normally consumes a lot of computational resources, since the product of all the partitions are considered.

REFERENCES

1. Z. Pawlak, *Rough Sets: Theoretical Aspects of Reasoning about Data*, Kluwer Academic Publishers, Netherlands, 1991.
2. L. A. Zadeh, "Fuzzy sets," *Information Science and Engineering*, Vol. 8, 1965, pp. 338-353.
3. A. Kharal and B. Ahmad, "Mappings on soft classes," *New Mathematics & Natural Computation*, Vol. 7, 2011, pp. 471-481.
4. D. Meng, X. Zhang, and K. Qinc, "Soft rough fuzzy sets and soft fuzzy rough sets," *Computers & Mathematics with Applications*, Vol. 62, 2011, pp. 4635-4645.
5. X. Liu, "Entropy, distance measure and similarity measure of fuzzy sets and their relations," *Fuzzy Sets and Systems*, Vol. 52, 1992, pp. 305-318.
6. C. Xu, "Improvement of the distance between intuitionistic fuzzy sets and its applications," *Journal of Intelligent & Fuzzy Systems*, Vol. 33, 2017, pp. 1563-1575.
7. G. W. Divine, H. J. Norton, A. E. Barón, and E. Juarez-Colunga, "The Wilcoxon-Mann-Whitney procedure fails as a test of medians," *The American Statistician*, Vol. 72, 2018, pp. 278-286.
8. M. P. Fay and M. A. Proschan, "Wilcoxon-Mann-Whitney or *t*-test? On assumptions for hypothesis tests and multiple interpretations of decision rules," *Statistics Surveys*, Vol. 4, 2010, pp. 1-39.
9. R. M. Chen, "A metric for finite power multisets of positive real numbers based on minimal matching," *Axioms*, Vol. 7, 2018.
10. O. U. Aydin, A. A. Taha, A. Hilbert, A. A. Khalil, *et al*., "On the usage of average Hausdorff distance for segmentation performance assessment: hidden error when used for ranking," *European Radiology Experimental*, Vol. 5, 2021.
11. M. Beauchemin, K. P. B. Thomson, and G. Edwards, "On the Hausdorff distance used for the evaluation of segmentation results," *Information Science and Engineering*, Vol. 24, 1998, pp. 3-8.
12. D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, "Comparing images using the Hausdorff distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 15, 1993, pp. 850-863.
13. R. M. Chen, "On COVID-19 country containment metrics: a new approach," *Journal of Decision Systems*, Vol. 31, 2022, pp. 116-133.

14. R. M. Chen, “Randomness for nucleotide sequences of SARS-CoV-2 and its related subfamilies,” *Computational and Mathematical Methods in Medicine*, Vol. 2020, 2020.
15. C. Faria, I. Serra, and R. Girardi, “A domain-independent process for automatic ontology population from text,” *Science of Computer Programming*, Vol. 95, 2014, pp. 26-43.
16. M. Shamsfard and A. A. Barforoush, “The state of the art in ontology learning: a framework for comparison,” *The Knowledge Engineering Review*, Vol. 18, 2003, pp. 293-316.
17. N. S. A. M. Lubani, M. and R. Mahmud, “Ontology population: Approaches and design aspects,” *Journal of Information Science*, Vol. 45, 2018, pp. 502-515.

Ray-Ming Chen received the MS degree in Mathematical Logic and the Theory of Computation from Manchester University, UK in 2006, and the Ph.D. degree in the Department of Mathematics, Leeds University, UK in 2011. Currently, he is an Assistant Professor at School of Mathematical Sciences, College of Science and Technology, Wenzhou-Kean University, China. His research interests include mathematics, data analysis, computational theories, statistics, and social science.