

SOME MEASURES RELATING PARTITIONS USEFUL FOR COMPUTATIONAL INTELLIGENCE

Ronald R. YAGER

*Machine Intelligence Institute, Iona College
New Rochelle, NY 10801, USA
E-mail: yager@panix.com*

Received October 13th, 2007
Revised November 6th, 2007

We investigate a number of measures relating partitions. One class of measures we consider are congruence measures. These measures are used to calculate the similarity between two partitionings. We provide a number of examples of this type of measure. Another class of measures we investigate are prognostication measures. This type of measure, closely related to a concept of containment between partitions, is useful in indicating how well knowledge of an objects class in one partitioning indicates its class in a second partitioning. We apply our measures to some data mining applications. One example is in choosing the appropriate level of a concept hierarchy. We also introduce a measure of the non-specificity of a partition. This measures a feature of a partition related to the granularity of the constituent classes of the partition.

Keywords: partitioning, clustering, granular computing data Mining, concept hierarchy, computational

1. Introduction

Partitioning of objects is a pervasive activity in the human cognition and conceptualization. For example a notable feature of human language is the implicit partitioning. Terms like small, medium and large essentially imply a partitioning of the objects over which they are applied. Concepts such as cold, warm and hot involve a partitioning of temperatures. Humans find it easier to reason with the type of granules that result from partitioning^{1,2}.

Modern technologies agenda to develop human like computational intelligence requires us to have sophisticated tools for managing objects that are partitions^{3,4,5}. In addition many applications in data mining involve the use of partitioning^{6,7,8}. The emerging field of granular computing^{9,10,11} is one attempt to provide these tools.

Our purpose here is to investigate some measures

that can be used in relating partitions.

One class of measures are what we call congruence measures. These measures are used to calculate the similarity between two partitionings. For example if we partition a set of people by education and openness to new technology we are interested in calculating how similar these two partitions are. Another application is comparing the results of two clustering algorithms. Another class of measures we investigate are what we refer to as prognostication measures. This type of measure, closely related to a concept of containment between partitions, is useful in indicating how well knowledge of an objects class in one partitioning indicates its class in a second partitioning. We also introduce a measure of the non-specificity of a partition. This measures a feature of a partition related to the granularity or size of the constituent classes of the partition.

2. Comparing Partitions

Assume X is a finite collection of objects. These objects can be people, countries, companies or any other class of objects. A common process performed within many tasks in modern intelligent computational technology is partitioning these objects. We recall this involves obtaining a collection of distinct subsets of X , A_1, \dots, A_q , called classes such that $A_i \cap A_j = \emptyset$ and $\bigcup_{j=1}^q A_j = X$. In this case each element $x_k \in X$ lies in exactly one of these classes. The data mining task of clustering^{12,13,14} involves a partition of the objects. The use of concept hierarchies¹⁵ involves a partition of object. The field of rough sets is based on a partitioning¹⁶. Many other tasks involve this type of partition operation.

Within this context it becomes interesting to compare partitions. Thus if $P_1 = \langle A_1, \dots, A_q \rangle$ and $P_2 = \langle B_1, \dots, B_p \rangle$ are two partitions a natural question is how similar or congruent are these two partitions. A example of this arises when we cluster objects by different features and are interested whether the resulting partitions are the same.

One goal here is to investigate and suggest a measures of congruence between partitions. Assume \mathbf{P} is the set of all partitions over the space X we are interested in obtaining a mapping $\text{Cong} : \mathbf{P} \times \mathbf{P} \rightarrow [0, 1]$ indicating the degree of congruence or similarity of two partitions. Assume P_1 and P_2 are two partitions we want this mapping to satisfy

- 1) $\text{Cong}(P_1, P_2) = \text{Cong}(P_2, P_1)$
- 2) $\text{Cong}(P_1, P_1) = 1$

That is we require that Cong is similarity relationship¹⁷.

We note that we can associate with any set X two special partitions. The first is the universal partition where we just have one set. We shall denote this $P^* = \langle X \rangle$. The other special partition is the one in which each element in X is in a distinct class. We denote this $P_* = \langle A_1, \dots, A_n \rangle$ where $A_i = \{x_i\}$. Conceptually these are the very opposite partitionings, they are the least similar of any pair of partitions Thus another property we desire of Cong is

- 3) $\text{Cong}(P^*, P_*) = \text{Min}[\text{Cong}(P_1, P_2)]$

We now turn to the formulation of Cong .

3. Equivalence Class Approach to Comparing Partitions

It is well known that any partitioning of X can be uniquely associated with an equivalence relation on X . Here we shall consider the formulation of the congruence mapping using this important connection.

We recall an equivalence relation E on X is a mapping $E : X \times X \rightarrow \{0, 1\}$ such that

- 1) $E(x, x) = 1$ Identity
- 2) $E(x, y) = E(y, x)$ Symmetry
- 3) $E(x, z) \geq \text{Min}_y [E(x, y), E(y, z)]$ Transitivity

Condition three implies that if $E(x, y) = 1$ and $E(y, z) = 1$ then $E(x, z) = 1$.

It is well known that if A_1, \dots, A_q is a partitioning of X then we can obtain an equivalence relation E such that $E(x, y) = 1$ if x and y are in the same class and $E(x, y) = 0$ if they are in different classes. Alternatively if E is an equivalence relation on X we can obtain a partitioning by assigning element x and y to the same class if $E(x, y) = 1$ and to different classes if $E(x, y) = 0$. From this perspective we can associate with each $x \in X$ an equivalence class A_x such that $y \in A_x$ if $E(x, y) = 1$.

We now note some special equivalence relations and the associated partitioning they induce. If E is such that $E(x, y) = 1$ for all x and y then E partitions X into one set $A_1 = X$. This is the partitioning we denoted as P^* . We denote this equivalence relation as E^* . At the other extreme is the case E_* where $E_*(x, y) = 0$ for $x \neq y$. In this case our classes are $A_i = \{x_i\}$. Here our are classes are n singleton sets. This is the partitioning we denoted as P_* .

Assume E_1 and E_2 are two equivalence relations. We shall say $E_1 \subseteq E_2$ if $E_1(x, y) \leq E_2(x, y)$ for all x and y . It can be shown that for any equivalence relation E we have $E_* \subseteq E \subseteq E^*$.

We also note that if $E_1 \subseteq E_2$ then if x and y are in the same equivalence class is under E_1 then are also in the same class under E_2 . Furthermore if E_1 induces the partitioning $\langle A_1, \dots, A_q \rangle$ and E_2 induces the partitioning $\langle B_1, \dots, B_p \rangle$ then $q \geq p$. That is E_1 generally has more equivalence classes.

We note that if for $x \neq y$ we indicate by $\langle x, y \rangle$ an

unordered pair, $\langle x, y \rangle = \langle y, x \rangle$, then if X has n elements we have $\binom{n}{2} = \frac{(n)(n-1)}{2}$ unordered pairs.

In the following we shall use U to indicate the set of all distinct unordered pairs from X .

In the following we shall find it convenient at times to express E as a subset of the space U of unordered pairs. Here we shall use the notation $U \cdot E$ to indicate the subset of U related to E . In this case $U \cdot E(\langle x, y \rangle) = E(x, y) = E(y, x)$. Furthermore we may find it useful at times to refer to $\binom{n}{2}$ as \mathbf{n}^c2 to indicate it is the combination of n elements taken two at a time.

We now suggest a measure of congruence between partitions which we express in terms of their underlying equivalence relations. Assume $P_1 : \langle A_1, \dots, A_q \rangle$ and $P_2 : \langle B_1, \dots, B_p \rangle$ are two partitions of X based on the equivalence relations E_1 and E_2 respectively. We define

$$\begin{aligned} \text{Cong}_1(P_1, P_2) &= \text{Cong}_1(E_1, E_2) \\ &= 1 - \frac{1}{\binom{n}{2}} \sum_U |U \cdot E_1(\langle x, y \rangle) - U \cdot E_2(\langle x, y \rangle)| \neq 0 \end{aligned}$$

Here $\sum_U |U \cdot E_1(\langle x, y \rangle) - U \cdot E_2(\langle x, y \rangle)|$ is the number of pairs that have different values in E_1 and E_2 .

We easily observe that $0 \leq \text{Cong}_1(E_1, E_2) \leq 1$. It is also clear that $\text{Cong}_1(E_1, E_2) = \text{Cong}_1(E_2, E_1)$.

Consider two relations E_1 and E_2 such that $E_1 = E_2$, that is $E_1(x, y) = E_2(x, y)$ for all x and y . Here they have the same partitions. In this case $U \cdot E_1(\langle x, y \rangle) - U \cdot E_2(\langle x, y \rangle) = 0$ for all $\langle x, y \rangle$ and therefore $\text{Cong}_1(E_1, E_1) = 1$. Thus if the classes of two partitions are the same then their congruence is one. We further note that if $E_1 \neq E_2$, there exists some x and y such that $E_1(x, y) \neq E_2(x, y)$ then $\sum_U |U \cdot E_1(\langle x, y \rangle) - U \cdot E_2(\langle x, y \rangle)| \neq 0$ and hence $\text{Cong}_1(E_1, E_2) < 1$. Thus $\text{Cong}_1(E_1, E_2) = 1$ if and only if $E_1 = E_2$.

Consider the equivalence relation E^* , here all $E^*(x, y) = 1$. This results in a partition with one

class $A_1 = X$. Let E be any other equivalence relation, partitioning of the space. It is clear that

$$\text{Cong}_1(E^*, E) = 1 - \frac{1}{\binom{n}{2}} (\#U \cdot E = 0)$$

where $\#U \cdot E = 0$ is number of elements in U that have zero membership in $U \cdot E$. From this it easily follows that if E_1 and E_2 are two equivalence relations such that $E_1 \subseteq E_2$ then $\text{Cong}_1(E^*, E_2) \geq \text{Cong}_1(E^*, E_1)$.

Analogously we see that with E_* the equivalence relation such that $E_*(x, y) = 0$ for $x \neq y$ then $\text{Cong}(E_*, E) = 1 - \frac{1}{\binom{n}{2}} (\#E = 1)$.

From this it follows that if E_1 and E_2 are such that $E_1 \subseteq E_2$ then $\text{Cong}(E_*, E_2) \leq \text{Cong}(E_*, E_1)$.

Consider the two partitions E^* and E_* . In this case $|U \cdot E^*(\langle x, y \rangle) - U \cdot E_*(\langle x, y \rangle)| = 1$ for all $\langle x, y \rangle$ and hence $\text{Cong}_1(E^*, E_*) = 0$. Thus these are the two complete opposite partitions. We can further see that for any other two pair of partitions $\text{Cong}_1(E_1, E_2) > 0$. Thus $\text{Cong}_1(E_1, E_2) = 1$ iff $E_1 = E_2$ and $\text{Cong}_1(E_1, E_2) = 0$ iff $E_1 = E^*$ and $E_2 = E_*$.

We also note here that for any partition E always has less equivalence classes than E_* and more than E^* .

Consider two equivalence relations E_1 and E_2 where $E_1 \subseteq E_2$, $E_1(x, y) \leq E_2(x, y)$. Let $P_1 : \langle A_1, \dots, A_q \rangle$ and $P_2 : \langle B_1, \dots, B_p \rangle$ be their respective partitionings. We observe that for any A_j there exists a B_k such that $A_j \subseteq B_k$. We see this as follows. Assume x and y are such that $E_1(x, y) = 0$ and $E_2(x, y) = 1$. Here x and y are in different equivalence classes under E_1 we denote these A_x and A_y . However under E_2 they are in the same equivalence class, we denote this B_k . Since all the elements equivalent to x under E_1 are also equivalent under E_2 and the same for y , then B_k must be such that such that $A_x \cup A_y \subseteq B_k$. We also note in this case $q > p$, the partitioning P_2 has less classes than the partitioning P_1 .

We can provide an alternative and sometimes

useful expression for

$$\sum_U |U \cdot E_1(\langle x, y \rangle) - U \cdot E_2(\langle x, y \rangle)|.$$

Let G be a subset of U defined such that

$$G = (U \cdot E_1 \cap \overline{U \cdot E_2}) \cup (\overline{U \cdot E_1} \cap U \cdot E_2)$$

Thus G is the set of elements in U which have

$$U \cdot E_1(\langle x, y \rangle) = 1 \text{ and } U \cdot E_2(\langle x, y \rangle) = 0$$

or

$$U \cdot E_1(\langle x, y \rangle) = 0 \text{ and } U \cdot E_2(\langle x, y \rangle) = 1$$

It is easy to see that

$$\text{Card}(G) = \sum_U |U \cdot E_1(\langle x, y \rangle) - U \cdot E_2(\langle x, y \rangle)|$$

hence we can express

$$\text{Cong}_1(E_1, E_2) = 1 - \frac{1}{\binom{n}{2}} \text{Card}(G)$$

We further observe that if we let $G_a = U \cdot E_1 \cap \overline{U \cdot E_2}$ and $G_b = \overline{U \cdot E_1} \cap U \cdot E_2$ then $G = G_a \cup G_b$ and $G_a \cap G_b = \emptyset$, G_a and G_b are disjoint. From this it follows that

$$\text{Card}(G) = \text{Card}(G_a) + \text{Card}(G_b)$$

This formulation will allow us to prove the following important property about the measure Cong_1 .

Theorem 1. *Assume E_1, E_2 and E_3 are three equivalence relations such that $E_1 \subset E_2 \subset E_3$ then*

$$\text{Cong}_1(E_3, E_2) \geq \text{Cong}_1(E_3, E_1).$$

Proof. First we observe that from the fact that $E_1 \subset E_2 \subset E_3$ we have $U \cdot E_1 \subset U \cdot E_2 \subset U \cdot E_3$. We shall denote $G_a = U \cdot E_3 \cap \overline{U \cdot E_2}$ and $G_b = \overline{U \cdot E_3} \cap U \cdot E_2$ and $H_a = U \cdot E_3 \cap \overline{U \cdot E_1}$ and $H_b = \overline{U \cdot E_3} \cap U \cdot E_1$. In this case

$$\text{Cong}_1(E_3, E_2) = 1 - \frac{1}{\binom{n}{2}} [\text{Card}(G_a) + \text{Card}(G_b)]$$

$$\text{Cong}_1(E_3, E_1) = 1 - \frac{1}{\binom{n}{2}} [\text{Card}(H_a) + \text{Card}(H_b)]$$

hence

$$\begin{aligned} & \text{Cong}_1(E_3, E_2) - \text{Cong}_1(E_3, E_1) \\ &= \frac{1}{\binom{n}{2}} [(\text{Card}(H_a) - \text{Card}(G_a)) \\ & \quad + (\text{Card}(H_b) - \text{Card}(G_b))] \end{aligned}$$

Consider now the term $G_b = \overline{U \cdot E_3} \cap U \cdot E_2$, it is the elements that are in $U \cdot E_2$ but not in $U \cdot E_3$, however since $E_2 \subset E_3$ then $G_b = \emptyset$. Similarly $H_b = \overline{U \cdot E_3} \cap U \cdot E_1$ are the elements in $U \cdot E_1$ that are not in $U \cdot E_3$ again since $E_1 \subset E_2$, we have $H_b = \emptyset$. Thus

$$\begin{aligned} & \text{Cong}_1(E_3, E_2) - \text{Cong}_1(E_3, E_1) \\ &= \frac{1}{\binom{n}{2}} (\text{Card}(H_a) - \text{Card}(G_a)) \end{aligned}$$

Consider now $H_a = U \cdot E_3 \cap \overline{U \cdot E_1}$ and $G_a = U \cdot E_3 \cap \overline{U \cdot E_2}$. Since $U \cdot E_1 \subseteq U \cdot E_2$ then $\overline{U \cdot E_2} \subseteq \overline{U \cdot E_1}$ and hence $G_a \subseteq H_a$ hence $\text{Card}(H_a) \geq \text{Card}(G_a)$ and the result follows. \square

Consider the space U of unordered pairs $\langle x, y \rangle$.

Consider now an $\binom{n}{2}$ -dimensional space Z whose

coordinates are (z_1, \dots, z_m) , here we let $m = \binom{n}{2}$.

Assume a bijective mapping M associating with each element $\langle x, y \rangle$ a dimension in the space Z , $M : U \rightarrow Z$. Using this we can associate with any equivalence relation E a vector $V \cdot E$ in this space Z such that the i^{th} component of the vector $V \cdot E(z_i) = U \cdot E(\langle x, y \rangle)$ where $\langle x, y \rangle$ is the pair corresponding to the dimension z_i . More formally $V \cdot E(z_1) = U \cdot E(M^{-1}(\langle x, y \rangle))$. Thus the z_i coordinate in $V \cdot E$ takes the value one if the unordered pair $\langle x, y \rangle$ associated with z_i is in E otherwise it takes the value zero.

Consider now two equivalence relations E_1 and E_2 defined on X . We can associate with each of these a vector as defined above, we denote these $V \cdot E_1$ and $V \cdot E_2$. Consider now the term $\sum_{\langle x, y \rangle \in U} |U \cdot E_1(\langle x, y \rangle) - U \cdot E_2(\langle x, y \rangle)|$. Letting $m =$

$\binom{n}{2}$ we see that this can be alternatively expressed as $\sum_{j=1}^m |V \cdot E_1(z_j) - V \cdot E_2(z_j)|$. This alternative expression is a well-known measure of distance, a metric, between the two vectors. It is an example of the more general Minkowski metric

$$L_p[(a_1, \dots, a_m), (b_1, \dots, b_m)] = \left(\sum_{i=1}^m |a_i - b_i|^p \right)^{\frac{1}{p}}$$

for $p > 0$. In the case where $p = 1$ this is called city block metric. We note that if the arguments are restricted to the unit interval then for any p this assumes its maximal value for the case when $a_i = 0$ and $b_i = 1$ for all i .

Thus we can view $\sum_{j=1}^m |V \cdot E_1(z_j) - V \cdot E_2(z_j)|$ as a distance between two vectors. We denote this $L_1(V \cdot E_1, V \cdot E_2)$. It is well known that $L_1(V \cdot E_1, V \cdot E_2)$ satisfies the properties of a metric.

Let \mathbf{E} be the set of all equivalence relations on X . Let $\mathbf{V} \cdot \mathbf{E}$ be the set of all associated vectors as defined above. Then L_p is a mapping

$$L_p : \mathbf{V} \cdot \mathbf{E} \longrightarrow \mathbf{V} \cdot \mathbf{E} \longrightarrow R$$

such that for any $V \cdot E_1, V \cdot E_2$ and $V \cdot E_3$ associated with the equivalence relations E_1, E_2 and E_3 we have

- 1) $L_p(V \cdot E_1, V \cdot E_2) = L_p(V \cdot E_2, V \cdot E_1)$ Symmetry
- 2) $L_p(V \cdot E_1, V \cdot E_2) = 0$ if $V \cdot E_1 = V \cdot E_2$
- 3) $L_p(V \cdot E_1, V \cdot E_3) \leq L_p(V \cdot E_1, V \cdot E_2) + L_p(V \cdot E_2, V \cdot E_3)$

Furthermore since $L_p(V \cdot E_1, V \cdot E_2)$ is a metric then

$$d(E_1, E_2) = \frac{1}{\binom{n}{2}} L_p(V \cdot E_1, V \cdot E_2)$$

is also a metric, distance measure, however it maps into the unit interval. From this we see that for any p the term $1 - d(E_1, E_2)$ is a similarity relation and can be used as a measure of congruence between E_1 and E_2 . Furthermore we note for the special equivalence relation E^* all the components in the vector $V \cdot E^*$ are ones while for the special equivalence relation E_* all the components in the vector $V \cdot E_*$ are zeros. From this it follows that $1 - d(E_*, E^*)$ will have the smallest congruence for any two equivalence relations.

4. Rand Index

In¹⁸ Rand suggested an approach for comparing the partitions that resulted from using different methods of clustering. This is called the Rand index.

Again suppose $P_1 : \langle A_1, \dots, A_q \rangle$ and $P_2 : \langle B_1, \dots, B_p \rangle$ are two partitions of X . In¹⁹ it was noted there are four different relationships that can exist among any of the $\binom{n}{2}$ pairs of distinct elements x and y of X .

- 1) The objects x and y are in the same class in both P_1 and P_2
- 2) The object x and y are in different classes in both P_1 and P_2
- 3) The objects x and y are in the same class in P_1 but in different classes in P_2
- 4) The objects x and y are in the different classes in P_1 but in the same class in P_2

If we let $a = \#$ of pairs of type 1, $b = \#$ of pairs of type 2, $c = \#$ of pairs of type 3 and $d = \#$ of pairs of type 4 then $a + b + c + d = \binom{n}{2}$, the total number of distinct pairs. Using this notation the Rand index R of the two partitions P_1 and P_2 is

$$R = \frac{a+b}{\binom{n}{2}} = \frac{a+b}{a+b+c+d}$$

Let E_1 and E_2 be the two equivalence relations on X defining the two partitions. Let U again be the space of unordered distinct pairs $\langle x, y \rangle$, $|U| = \binom{n}{2}$. Let $U \cdot E_1$ be a subset of U defined by $U \cdot E_1(\langle x, y \rangle) = E_1(x, y)$. Similarly $U \cdot E_2$ is defined as $U \cdot E_2(\langle x, y \rangle) = E_2(x, y)$. We note that $\text{Card}(U \cdot E_1)$ is the number of pairs elements that are in the same class in P_1 . Similarly $\text{Card}(U \cdot E_2)$ is the number of pairs of elements that are in the same class in P_2 . Furthermore $U \cdot E_1 \cap U \cdot E_2$ are all pairs that are in the same class in both partitionings. Hence $a = \text{Card}(U \cdot E_1 \cap U \cdot E_2)$. On the other hand $\overline{U \cdot E_1}$

is the subset of pairs that are in different classes in P_1 . Similarly $\overline{U \cdot E_2}$ is the set of pairs that in different classes is P_2 . Here then $\overline{U \cdot E_1} \cap \overline{U \cdot E_2}$ is the subset of pairs that are in different classes in both partitions. Thus we have $b = \text{Card}(\overline{U \cdot E_1} \cap \overline{U \cdot E_2})$. Using this we that we can express the Rand index as

$$R = \frac{a+b}{\binom{n}{2}} = \frac{\text{Card}(U \cdot E_1 \cap U \cdot E_2) + \text{Card}(\overline{U \cdot E_1} \cap \overline{U \cdot E_2})}{\binom{n}{2}}$$

This is equivalent to the Cong_1 measure.

5. An Alternative Measure of Congruence

In the preceding we have introduced a measure of similarity, congruence, between two partitions using the underlying equivalence relations. Here we now consider formulating a congruence measure from the perspective of the partitions themselves.

Assume we have two partitions of the space X ,

$$P_1 = A_1, \dots, A_q$$

$$P_2 = B_1, \dots, B_p$$

here

$$X = \bigcup_{j=1}^q A_j \text{ and } A_i \cap A_j = \emptyset \text{ for } i \neq j$$

and

$$X = \bigcup_{j=1}^p B_j \text{ and } B_i \cap B_j = \emptyset \text{ for } i \neq j$$

Without loss of generality we shall assume $q = p$. If $q > p$ we can augment the partition P_2 by adding $q - p$ subsets, $B_{p+1} = B_{p+2} = \dots = B_q = \emptyset$. Thus in the following we assume the two partitions have the same number of classes, q .

We now introduce an operation called a pairing of P_1 and P_2 , denoted $g(P_1, P_2)$, which associates with each subset A_i of P_1 a unique partner B_j from P_2 . Formally if $Q = \{1, 2, \dots, q\}$ then a pairing is a mapping $g : Q \rightarrow Q$ that is bijective, one to one and onto. Essentially g is a permutation of Q . We then have that a pairing $g(P_1, P_2)$ is a collection of q

pairs, $(A_j, B_{g(j)})$. We shall now associate with each pairing a score, $\text{Score}(g(P_1, P_2))$, defined as follows. Denoting $D_{g,j} = A_j \cap B_{g(j)}$ for $j = 1$ to q , we obtain

$$\text{Score}(g(P_1, P_2)) = \frac{\sum_{j=1}^q \text{Card}(D_{g,j})}{\text{Card}(X)}$$

In the following we illustrate the process of determining $\text{Score}(g(P_1, P_2))$.

Example. Let $X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$.

Assume P_1 is $A_1 = \{x_1, x_2, x_3\}$, $A_2 = \{x_4, x_5, x_6\}$ and $A_3 = \{x_7\}$.

Assume P_2 is $B_1 = \{x_2, x_4, x_6\}$ and $B_2 = \{x_1, x_3, x_5, x_7\}$.

We first augment P_2 with $B_3 = \emptyset$.

One pairing is $g(j) = j$ in this case we get the pairs (A_1, B_1) , (A_2, B_2) , (A_3, B_3) . From this

$$D_{g,1} = A_1 \cap B_1 = \{x_2\}$$

$$D_{g,2} = A_2 \cap B_2 = \{x_5\}$$

$$D_{g,3} = A_3 \cap B_3 = \emptyset$$

In this case $\text{Score}(g(P_1, P_2)) = \frac{2}{7}$.

Another pairing is $g(1) = 3$, $g(2) = 1$, $g(3) = 2$ here our pairs are

$$(A_1, B_3), (A_2, B_1), (A_3, B_2)$$

In this case

$$D_{g,1} = A_1 B_3 = \emptyset$$

$$D_{g,2} = A_2 B_1 = \{x_4, x_6\}$$

$$D_{g,3} = A_3 B_2 = \{x_7\}$$

In this case $\text{Score}(g(P_1, P_2)) = \frac{3}{7}$.

We now shall use this to obtain a measure of congruence, $\text{Cong}_2(P_1, P_2)$. Let G be the set of all pairings, $g \in G$. We define

$$\text{Cong}_2(P_1, P_2) = \mathbf{Max}_{g \in G} [\text{Score}(g(P_1, P_2))]$$

Thus this measure of congruence is the score of the largest pairing.

We see that for any pairing g ,

$$0 \leq \sum_{j=1}^q \text{Card}(D_{g,j}) \leq \text{Card}(X)$$

From this it follows that $0 \leq \text{Cong}_2(P_1, P_2) \leq 1$.

More precisely since for any two partitions we can

always find a pairing g in which $\sum_{j=1}^q \text{Card}(D_{g,j}) \geq 1$

we see that

$$\frac{1}{\text{Card}(X)} \leq \text{Cong}_2(P_1, P_2) \leq 1$$

Let us look at some special cases of the measure of congruence. In the following we assume $\text{Card}(X) = n$. First consider the case when $P_1 = P_2$ that is

$$P_1 : A_1, A_2, \dots, A_q$$

$$P_2 : B_1, B_2, \dots, B_q$$

where $A_j = B_j$. We see the pairing g such that $g(j) = j$ has the pairs (A_j, B_j) for $j = 1$ to q . In this case $D_{g,j} = A_j \cap B_j = A_j$. We get that $\sum_{j=1}^q \text{Card}(D_{g,j}) = \sum_{j=1}^q \text{Card}(A_j) = n$. For this g we

have $g(P_1, P_2) = \frac{n}{n} = 1$. From this we get that $\text{Cong}_2(P_1, P_2) = \mathbf{Max}_{g \in G} [\text{Score}(g(P_1, P_2))] = 1$. Thus

if the partitions are the same the congruence is 1. It can be shown that $\text{Cong}_2(P_1, P_2) = 1$ if and only if $P_1 = P_2$.

Consider now the situation in which one of the partitions P^* is simply X the whole space, it is the grand partition and the other partition $P = A_1, \dots, A_q$. We see here that there are only q distinct pairing mappings possible. Each one corresponds to an association of X with one of the A_j and the other classes in P are associated with \emptyset . For the pairing g_j that associates X with A_j we have $X \cap A_j = A_j$. In this case we get $\text{Score}(g_j) = \frac{\text{Card}(A_j)}{n}$. Thus here

then $\text{Cong}_2(P^*, P) = \mathbf{Max}_j \left[\frac{\text{Card}(A_j)}{n} \right]$. Thus for the grand partition P^* its congruence with any other partition is simply $\mathbf{Max}_j [\text{Card}(A_j)]$, its the proportion of elements in the largest class of P .

Consider now the case in which one partition, P_* , is the partition consisting of classes which are singletons, $P_* = A_1, \dots, A_n$, where $A_i = \{x_i\}$. Let $P = B_1, \dots, B_q$ be any arbitrary partitioning. We can easily see that in this case $\text{Cong}_2(P_*, P) = q/n$ where q is the number of classes in P . In the special case where P is P_* we get that $\text{Cong}_2(P_*, P_*) = 1/n$.

6. Concept Hierarchies

An important application of partitioning is the case of concept hierarchies. Let X be a set of objects. A concept hierarchy is a collection of partitions, P_1, \dots, P_r . Here P_k is called the k^{th} level partition. The fundamental property of the concept hierarchy is that each class (granular or cluster) in a lower level partition is fully contained in one class of the next higher level partitioning. The partitioning becomes more coarse as we go up. In Fig. 1 we illustrate the idea of a concept hierarchy.

P₄	A ₄						
P₃	A ₃₁		A ₃₂				
P₂	A ₂₁	A ₂₂		A ₂₃			
P₁	A ₁₁	A ₁₂	A ₁₃	A ₁₄	A ₁₅	A ₁₆	A ₁₇

Fig. 1. Concept Hierarchy

Formally here we have a collection of partitions

$$P_1 : A_{11}, A_{12}, \dots, A_{1q_1}$$

$$P_2 : A_{21}, A_{22}, \dots, A_{2q_2}$$

⋮

$$P_r : A_{r1}, A_{r2}, \dots, A_{rq_r}$$

We note that for $m > k$ we have $q_m < q_k$.

The fundamental property of a concept hierarchy is expressed in the following. Let k and m be two levels such that $m > k$, m is higher than k . For any class in the k^{th} level, A_{kj} , there exists a class in the m^{th} level, A_{mi} such that $A_{kj} \subseteq A_{mi}$. Essentially here then with $m > k$ we have for any A_{mi} that $A_{mi} = \bigcup_{j \in S_{mi/k}} A_{kj}$ where $S_{mi/k} \subseteq \{1, \dots, q_k\}$.

We note the most extreme possible upper partitioning is the one consisting of the whole space while the most extreme possible lower partitioning is the one whose classes are singletons.

We now consider the calculation of the congruence between partitions in this concept hierarchy. We let $\text{Cong}_2(P_k, P_m)$ be the congruence between

the k^{th} and m^{th} level partitionings where we assume $m > k$. Here again we let $S_{mi/k}$ be the collection of class in P_k that are contained in A_{mi} . We now let $\text{Max}_{j \in S_{mi/k}} [\text{Card}(A_{kj})] = \text{Big}_{mi/k}$, it is the cardinality of the largest class in the partition P_k that is contained in A_{mi} . Using this we see that

$$\text{Cong}_2(P_k, P_m) = \frac{1}{n} \sum_{i=1}^{q_m} \text{Big}_{mi/k}$$

Thus it is the sum of the cardinalities of the largest class in P_k associated with a class of P_m .

A special case of determining congruence in concept hierarchies is when $m = k + 1$. Here we are considering the similarity between adjacent partitionings. In this case we can note some features that make adjacent partitions more similar. Without loss of generality assume the classes in P_k have been indexed such that $\text{Card}(A_{ki}) \geq \text{Card}(A_{kj})$ for $i < j$. Thus we index the P_k in decreasing order of cardinality. Consider now the construction of the partitioning P_{k+1} where we shall allow q_{k+1} categories. We recall $q_{k+1} < q_k$.

We see that we can maximize $\text{Cong}_2(P_k, P_{k+1})$ by assigning the q_{k+1} largest classes in P_k to different classes in P_{k+1} . Here then we let $A_{ki} \subseteq A_{k+1i}$ for $i = 1$ to q_{k+1} . Thus each of the largest classes i in P_k should be the focus of one of the classes in P_{k+1} . We also observe that the congruence increases as the size of q_{k+1} increases.

7. Optimal Solution

In the general case the calculation of $\text{Cong}_2(P_1, P_2)$ requires the solution of an optimization problem, that is we must determine the pairing g that maximize the score. In the following we shall provide a view of this optimization problems that allows for a classical solution.

Here then we have two partitions $P_1 : A_1, \dots, A_q$ and $P_2 : B_1, \dots, B_q$ where we have assumed the inclusion of null sets if necessary. We can associate with these two partitions the matrix N show below. In this matrix $n_{ij} = \text{Card}(A_i \cap B_j)$, it is the number of elements that A_i and B_j have in common. We note that the sum of the elements in this i^{th} row is equal

to $\text{Card}(A_i)$ while the sum of the element is the j^{th} column is $\text{Card}(B_j)$. The sum of all the elements in the matrix is n , $\text{Card}(X)$.

$$\begin{array}{c} \\ A_1 \\ \\ A_i \\ \\ A_q \end{array} \begin{array}{ccc} B_1 & B_j & B_q \\ \left[\begin{array}{ccc} n_{11} & & n_{1q} \\ & n_{ij} & \\ n_{q1} & & n_{qq} \end{array} \right] \end{array}$$

The problem of calculating $\text{Cong}_2(P_1, P_2)$ can be viewed using the perspective of the matrix. A pairing g is an association of each row A_i with a unique column B_j , each pairing results in the selection of q elements consisting of one from each row and one from each column. Furthermore each pair (A_i, B_j) in a pairing g has associated value n_{ij} . The score of the pairing g is the sum of these q associated values. The problem of determining $\text{Cong}_2(P_1, P_2)$ is then to find the selection of these q values that provide the largest total. This problem is a well-known optimization problem in operation research, it is called the assignment problem²⁰. Techniques have been developed for solving this problem. We note that if q is small then we can solve this problem by enumeration. We note that if g is an assignment function, $g(i)$ assigns $B_{g(i)}$ to A_i , then the score of g equals $\sum_{i=1}^q n_{ig(i)}$.

Here we shall present a view of this optimization problem that allows for easy solution by enumeration if q is small or by use of genetic algorithms²¹ if q is large.

To calculate $\text{Cong}_2(P_1, P_2)$ we must calculate $\text{Max}_{g \in G} [\text{Score}(g(P_1, P_2))]$. Using a mapping g the pairs are $(A_i, B_{g(i)})$ and the score is

$$\text{Score}(g(P_1, P_2)) = \sum_{i=1}^q \text{Card}[A_i \cap B_{g(i)}] = \sum_{i=1}^q n_{ig(i)}$$

Consider the simple assignment $g(i) = i$ here then A_i and B_i are assigned as pairs. In this case

$$\text{Score}(g(P_1, P_2)) = \sum_{i=1}^q n_{ii}$$

it is the sum of the elements on the diagonal of the matrix N .

We note that any mapping $g : Q \rightarrow Q$ can be viewed as a permutation of the B_j . We see that for any permutation g of $\langle B_1, \dots, B_q \rangle$ we get an interchange of the columns of N . For this permutation, however, its score can be obtained as the sum of the elements on the main diagonal.

Thus one way to obtain the value of $\text{Cong}_2(P_1, P_2)$ is to do all the possible permutations of the columns. Total up the values of the main diagonal for each permutation. The largest of these totals is the value of $\text{Cong}_2(P_1, P_2)$. In cases where q is small this can be easily done by enumeration. In cases where q is large the total number of permutations, $\text{Card}(G)$, becomes too large for enumeration. Here we can use genetic algorithms to get an optimal solution [5, 21].

We can express this optimization problem in the form that can be solved using a genetic algorithm. Consider our genomes to be permutations of the $\{1, 2, \dots, q\}$. In the case where $q = 6$ an example of this would be $\hat{g} = [2\ 3\ 4\ 5\ 6\ 1]$.

Each \hat{g} genome can be seen to induce a matrix in which the i^{th} column is $B_{\hat{g}(i)}$. Given a particular genome \hat{g} our fitness function is

$$F = \sum_{i=1}^q n_{i\hat{g}(i)}$$

We can now search that the whole space of genomes G using the established procedures for searching a permutation space with genetic algorithms.

8. Measure of Prognostication

In the following we shall investigate a different type of measure of the relationship between two partitions. Assume P_1 and P_2 are two partitions of X ,

$$P_1 : A_1, \dots, A_q$$

$$P_2 : B_1, \dots, B_r$$

Here we will try to measure how well we are able to tell what class an object is in the partition P_2 from our knowledge of the class it is in partition P_1 . This can be seen as a kind of prognostication or prediction. We call denote this measure $\text{Prog}(P_1, P_2)$. As we shall see this will not be symmetrical, that is $\text{Prog}(P_1, P_2) \neq \text{Prog}(P_2, P_1)$ but it will be such that $\text{Prog}(P_j, P_j)$ will assume the highest value. As we

shall subsequently see this concept is closely related to the idea of containment between the partitions. In the following we shall assume $\text{Card}(X) = n$.

We first define what we shall call an association. Let $Q = \{1, 2, \dots, q\}$ and let $R = \{1, 2, \dots, r\}$ we define an association to be a mapping $h : Q \rightarrow R$. Thus for each $i \in Q$ we get $h(i) \in R$. We note that h need not be bijective. This for $i \neq k$ we can have $h(i) = h(k)$. Essentially this mapping associates with each class A_i in partition P_1 a class B_j in partition P_2 . While more than one A_i can be associated with each B_j , each A_i is only associated with only one B_j . For a given h we define $\text{Fam}(j) = \{A_i \mid h(i) = j\}$, it is the family of classes in P_1 that map into B_j in partition P_2 . We let H be the set of all association mappings.

We now obtain for any association mapping h a value

$$\text{Value}(h) = \frac{1}{n} \sum_{j=1}^r \text{Card} \left(B_j \cap \bigcup_{i \in \text{Fam}(j)} A_i \right)$$

Using these values we, define

$$\text{Prog}(P_1, P_2) = \text{Max}_{h \in H} [\text{Value}(h)]$$

Thus the degree of prognostication of P_1 for P_2 is the largest value of any association.

We now make some observations about the properties of $\text{Value}(h)$ and $\text{Prog}(P_1, P_2)$. First we observe that for any h , $\text{Value}(h) \geq 0$ and $\text{Value}(h) \leq 1$. From this we conclude that

$$0 \leq \text{Prog}(P_1, P_2) \leq 1.$$

Let us now consider some special cases of P_1 and P_2 . First consider the case where $P_1 = P_2$. In this case we have

$$P_1 : A_1, A_2, \dots, A_q$$

$$P_2 : B_1, B_2, \dots, B_q$$

where $B_i = A_i$. Consider now the association mapping $h(i) = i$. In this case

$$\text{Value}(h) = \frac{1}{n} \sum_{j=1}^q \text{Card}(A_j \cap B_j) = 1$$

From this we can conclude that $\text{Prog}(P_1, P_2) = 1$.

Another interesting special case is where P_2 is P^* it just consists of one set X ,

$$P_1 : A_1, \dots, A_q$$

$$P_2 : B_1$$

where $B_1 = X$. Here there is only one association mapping, $h(i) = 1$ for all i . In this case

$$\text{Value}(h) = \frac{1}{n} \text{Card} \left(X \cap \bigcup_{j=1 \text{ to } q} A_j \right) = 1$$

Thus here again we get

$$\text{Prog}(P, P^*) = 1.$$

Consider now another special case. Here we assume P_1 is P^* , each $A_i = \{x_i\}$ it is a singleton set and P_2 is any arbitrary partition. Thus in this case

$$\begin{aligned} P_1 &: A_1, A_2, \dots, A_n \\ P_2 &: B_1, B_2, \dots, B_r \end{aligned}$$

Consider the association mapping $h(i)$ such that $h(i) = j$ where $x_i \in B_j$. In this case

$$\text{Value}(h) = \frac{1}{n} \sum_{j=1}^r \text{Card} \left(B_j \cap \bigcup_{x_i \in B_j} A_i \right) = 1$$

Again here we get $\text{Prog}(P_*, P) = 1$.

Actually these three preceding situations can be seen to be special cases of a more general relationship between P_1 and P_2 . This relationship is a hierarchical or containment relationship between the two partitions. We define this in the following. Assume

$$\begin{aligned} P_1 &: A_1, A_2, \dots, A_q \\ P_2 &: B_1, B_2, \dots, B_r \end{aligned}$$

we say $P_1 \subseteq P_2$ if for each A_i there exists a B_j such that $A_i \subseteq B_j$. It is clear that $r \leq q$.

For the situation in which $P_1 \subseteq P_2$ consider the association mapping h where $h(i) = j$ where $A_i \subseteq B_j$. In this case $\bigcup_{i \in \text{Fam}(j)} A_i = B_j$ and hence

$$\begin{aligned} \text{Value}(h) &= \frac{1}{n} \sum_{j=1}^r \text{Card} \left(B_j \cap \bigcup_{h(i)=j} A_i \right) \\ &= \frac{1}{n} \sum_{j=1}^r \text{Card}(B_j) = 1 \end{aligned}$$

Thus in this case of $P_1 \subseteq P_2$ we always get $\text{Prog}(P_1, P_2) = 1$.

Here then we can view $\text{Prog}(P_1, P_2)$ essentially as a measure of containment of the cluster P_1 in P_2 .

Let us look in the opposite direction. Assume are partitions are

$$\begin{aligned} P_1 &: A_1 = X \\ P_2 &: B_1, \dots, B_r \end{aligned}$$

In this case there exists r possible association mappings h_1, \dots, h_r where for h_j we have $h_j(1) = j$. For any of these mappings we have $\text{Value}(h_j) = \frac{1}{n} \text{Card}(B_j)$. Thus here $\text{Prog}(P^*, P_2) = \text{Max}_j [\text{Card}(B_j)]$, it is the proposition of the elements in the largest cluster of P_2 .

Consider now the case where P_2 is singleton, $B_j = \{x_j\}$ and P_1 is arbitrary

$$\begin{aligned} P_1 &: A_1, \dots, A_q \\ P_2 &: B_1, \dots, B_n \end{aligned}$$

With some thought it can be seen there the maximum of $\text{Value}(h)$ occurs for any mapping h that associates A_i with one of the sets B_j whose element is in A_i , $x_j \in A_i$. In the case we get

$$\text{Value}(h) = \frac{1}{n} \sum_{j=1}^n \text{Card} \left(B_j \cap \bigcup_{h(i)=j} A_i \right) = \frac{q}{n}$$

Thus $\text{Prog}(P_1, P_*) = \frac{q}{n}$.

We can provide a lower bound on $\text{Prog}(P_1, P_2)$ for any two partitionings. Again assume

$$\begin{aligned} P_1 &: A_1, \dots, A_q \\ P_2 &: B_1, \dots, B_r \end{aligned}$$

Consider a mapping h_j such that for each i , $h_j(i) = j$. Here every class in P_1 is associated with B_j . In this case

$$\begin{aligned} \text{Value}(h_j) &= \text{Card}(B_j \cap X) \\ &= \text{Card} \left(B_j \cap \bigcup_{i=1}^q A_i \right) = \text{Card}(B_j) \end{aligned}$$

From this we can conclude that $\text{Prog}(P_1, P_2) \geq \text{Max}_j [\text{Card}(B_j)]$.

We now shall provide a view of the situation which will greatly simplify the calculation of $\text{Prog}(P_1, P_2)$. Again assume we have two partitions $P_1 : \langle A_1, \dots, A_q \rangle$ and $P_2 : \langle B_1, \dots, B_r \rangle$. Consider the matrix N shown below. In this matrix the value n_{ij} is the number of elements A_i and B_j have in common, that is $\text{Card}(A_i \cap B_j) = n_{ij}$. Here the sum of the elements in the i^{th} row is $\text{Card}(A_i)$ and the sum of the

elements in the j^{th} column is $\text{Card}(B_j)$.

$$\begin{array}{c} A_1 \\ A_i \\ A_q \end{array} \begin{bmatrix} B_1 & B_j & B_r \\ & n_{ij} & \\ & & \end{bmatrix}$$

We recall that any association mapping h assigns to each class A_i in P_1 a class in P_2 which we denote as $B_{h(i)}$. Consider now

$$\text{Value}(h) = \frac{1}{n} \sum_{j=1}^r \text{Card} \left(B_j \cap \bigcup_{i \text{ s. t. } h(i)=j} A_i \right)$$

We observe $A_i \cap A_j = \emptyset$, they are disjoint, thus we can rewrite the above as

$$\text{Value}(h) = \frac{1}{n} \sum_{j=1}^r \left(\sum_{i \text{ s. t. } h(i)=j} \text{Card}(B_j \cap A_i) \right)$$

Furthermore since A_i is associated with only one B_j we can reformulate this as

$$\begin{aligned} \text{Value}(h) &= \frac{1}{n} \sum_{j=1}^r \left(\sum_{i \text{ s. t. } h(i)=j} \text{Card}(B_{h(i)} \cap A_i) \right) \\ &= \frac{1}{n} \sum_{i=1}^q \text{Card}(B_{h(i)} \cap A_i) \\ \text{Value}(h) &= \frac{1}{n} \sum_{i=1}^q n_{ih(i)} \end{aligned}$$

Thus for a given h its $\text{Value}(h)$ is the sum of the $n_{ih(i)}$. Essentially for a given h for each A_i we select one element in N from its corresponding row and then add these up. From this observation the determination of h^* such that $\text{Val}(h^*) = \text{Max}_h[\text{Value}(h)]$ becomes clear. We select from each row the largest element in N . From this we easily calculate the value of Prog

$$\text{Prog}(P_1, P_2) = \frac{1}{n} \sum_{i=1}^q \text{Max}_j [n_{ij}]$$

Thus the calculation of $\text{Prog}(P_1, P_2)$ simply requires us to determine the largest value in each row of N .

In the following we shall let $g : Q \rightarrow R$ be the unique association such that $n_{ig(i)} = \text{Max}_j [n_{ij}]$. We now clearly understand the situation which we refer to as $P_1 \subseteq P_2$. In this case for each A_i there exists one B_j such that $A_i \subseteq B_j$. Let us denote this $g(i)$, thus $A_i \subseteq B_{g(i)}$. In this case $A_i \cap B_{g(i)} = A_i$ and hence $n_{ig(i)} = \text{Card}(A_i)$ and furthermore $n_{ij} = 0$ for $j \neq g(i)$. Thus each row has only one non-zero entry and the value of that is $\text{Card}(A_i)$. It is clear how we pick these elements and this gives us $\text{Prog}(P_1, P_2) = 1$.

Thus the determination of $\text{Prog}(P_1, P_2)$ can be seen as type of assignment problem. We assign each A_i to one B_j , the one with the maximal common elements. Since we can assign multiple A_i to same B_j there is no conflict and each assignment can be done independently. Thus this is easier than the previous problem.

Let us define $U_i = |A_i| - n_{ig(i)} = \sum_{j \neq g(i)} n_{ij}$. It is the number of elements in A_i not in $B_{g(i)}$. We now see that

$$\begin{aligned} \text{Prog}(P_1, P_2) &= \frac{1}{n} \sum_{i=1}^q n_{ig(i)} \\ &= \frac{1}{n} \sum_{i=1}^q |A_i| - U_i = 1 - \frac{1}{n} \sum_{i=1}^q U_i \end{aligned}$$

it is one minus the proportion of the element not in the assignment set.

We see the following the following theorem holds.

Theorem 2. $\text{Prog}(P_1, P_2) = 1$ if and only if for each A_i there exists an B_j such that $A_i \subseteq B_j$.

Proof. This is the necessary and sufficient for $\text{Prog}(P_1, P_2) = 1$. \square

Definition 1. Assume $P_1 : \langle A_1, \dots, A_q \rangle$ is a partition of X . Assume $P_2 : \langle B_1, \dots, B_p \rangle$ is another partition of X where $q \geq p$. If for each A_i there exists a B_j such that $A_i \subseteq B_j$ then we say that P_1 is a *refinement* of P_2 . Alternatively we say P_2 is a *coarsing* of P_1 .

Note. Earlier we showed if P_1 is a refinement of P_2 then $\text{Prog}(P_1, P_2) = 1$.

Theorem 3. Assume P_1 is a refinement of P_2 . Let P_3 be any arbitrary partition of X then

$$\text{Prog}(P_1, P_3) = \text{Prog}(P_2, P_3).$$

Proof. Assume $P_1 : \langle A_1, \dots, A_q \rangle$, $P_2 : \langle B_1, \dots, B_p \rangle$ and $P_3 : \langle D_1, \dots, D_m \rangle$. Here $q \geq p$. Let g_2 be the optimal mapping between P_2 and P_3 hence $\text{Prog}(P_2, P_3) = \sum_{j=1}^p |B_j \cap D_{g_2(j)}|$. Because of the relationship between P_1 and P_2 each B_j is the union of a subset of classes from P_1 . We shall denote the subset of these classes making up B_j as $C(j)$ and indicate an element in $C(j)$ as A_{k_j} . Using this notation then $B_j = \bigcup_{k_j \in C(j)} A_{k_j}$

$$B_j = \bigcup_{k=1}^{n_j} A_{jk}$$

Using this we see that

$$\begin{aligned} \text{Prog}(P_2, P_3) &= \sum_{j=1}^p |B_j \cap D_{g_2(j)}| \\ &= \sum_{j=1}^p \left(\sum_{k_j \in C(j)} |A_{k_j} \cap D_{g_2(j)}| \right) \end{aligned}$$

Let g_1 be a mapping associating a class in P_1 with a class in P_3 such that for any A_{k_j} we have $|A_{k_j} \cap D_{g_1(k_j)}| = \mathbf{Max}_{\text{all } D_i} |A_{k_j} \cap D_i|$. From this we have

$$\text{Prog}(P_1, P_3) = \sum_{i=1}^q |A_i \cap D_{g_1(i)}| \text{ however}$$

$$\sum_{i=1}^q |A_i \cap D_{g_1(i)}| = \sum_{j=1}^p \left(\sum_{k_j \in C(j)} |A_{k_j} \cap D_{g_1(j)}| \right)$$

and therefore

$$\text{Prog}(P_1, P_3) = \sum_{j=1}^p \left(\sum_{k_j \in C(j)} |A_{k_j} \cap D_{g_1(j)}| \right)$$

Since $|A_{k_j} \cap D_{g_1(k_j)}| \geq |A_{k_j} \cap D_{g_2(k_j)}|$ the result follows. \square

9. Some Applications

One application of $\text{Prog}(P_1, P_2)$ is to provide a measure of congruence. In particular we can define

$$\text{Cong}_3(P_1, P_2) = \text{Prog}(P_1, P_2) \wedge \text{Prog}(P_2, P_1)$$

where $\wedge = \text{Min}$.

Let us see what are the properties of this definition. First we see that it is symmetric

$$\text{Cong}_3(P_1, P_2) = \text{Cong}_3(P_2, P_1)$$

Furthermore since $\text{Prog}(P_1, P_1) = 1$ then we have $\text{Cong}_3(P_1, P_1) = 1$. We further observe that if $P_1 \subseteq P_2$ then we showed $\text{Prog}(P_1, P_2) = 1$ and hence $\text{Cong}_3(P_1, P_2) = \text{Prog}(P_2, P_1)$.

Consider the case where P_1 is the partition into singleton, P_* , and P_2 is the whole space, P^* . Here $\text{Prog}(P_*, P^*) = 1$ while $\text{Prog}(P_*, P_*) = 1/n$ and hence $\text{Cong}_3(P_*, P^*) = 1/n$.

Another application of $\text{Prog}(P_1, P_2)$ is in helping build relations between variables. Again assume X is a collection of objects. Let U and V be two attributes associated with these objects. Let $P_1 : \langle A_1, \dots, A_r \rangle$ be a partition of the objects in X based upon the attribute U . Let $P_2 : \langle B_1, \dots, B_q \rangle$ be a partition of the object in X based upon the second attribute V . Here we are interested being able to determine in which class in P_2 an object falls given our knowledge of which class in the partition P_1 the object lies. Here we want relations like

If an objects U value is in A_i then its V value is in B_j .

Here the quantity $\text{Prog}(P_1, P_2)$ helps us determine which is the best association to make. The function g that maximizes $\text{Prog}(P_1, P_2)$ provides the best association. In this case the value of $\text{Prog}(P_1, P_2)$ provides some measure of the usefulness or quality of the relation obtained.

An important example of this problem is the following. Assume we have some preferred partitioning, $P_2 : \langle B_1, \dots, B_r \rangle$, of the objects in X based upon some attribute V . Let F be a family of partitions of X based upon another attribute U . Here $F = \{P_{11}, \dots, P_{1s}\}$ where each P_{1k} is a partition of X

based on U . That is $P_{1k} : \langle A_{k1}, \dots, A_{kq_k} \rangle$. Our objective here is to find the partition in F that gives the best rules for the connection between the U and V attributes. Here we must first search through each P_{1k} to find its best association function, g_k , and obtain the value for $\text{Prog}(P_{1k}, P_2)$. Using this we then select the partition P_{1k^*} such that $\text{Prog}(P_{1k^*}, P_2) = \text{Max}_{k \in F} [\text{Prog}(P_{1k}, P_2)]$. This type of problem naturally arises in situations in which a decision maker knows what action to perform if the object is in any one of the classes of P_2 , B_j , but makes observations about the objects in terms of attribute U . A typical example of this occurs in the medical domain. In this type of problem the decision maker, doctor, would know from his training what treatment to recommend if a patient has a particular disease. However he usually doesn't directly observe a disease, instead he observes measurable physical attributes of a patient. Here U would correspond to measurable physical attributes of a patient and the partition P_2 would correspond to different diseases. In order to prescribe the treatment he must be able to determine the current disease from the symptoms U . The goal in selecting the optimal partition of U is of course to accurately determine the disease from the measurements of the physical attributes U . In order to accomplish this task we must enable the decision to comprehend the values of the attributes. Human comprehension of data is generally enhanced by using granularization to view the values of the attributes. For example blood pressure of 180 is better viewed as high blood pressure. Body temperature of 99.3 is most usefully viewed as normal. With this understanding we see that each partition of P_{1k} of U corresponds to a different granularization of the attribute. We then see that F provides a collection of different granularizations which are comprehensible to the decision making. Thus by initially introducing F we are essentially trying to constrain the space in which we are looking for the optimal partitioning to be those which are cognitively comprehensible to the decision maker. Once having introduced F our objective then becomes selecting from these different comprehensible granulations the one that leads to the most accurate determination of the disease from the systems.

The following Fig. 2 illustrates the essential features of this paradigm.

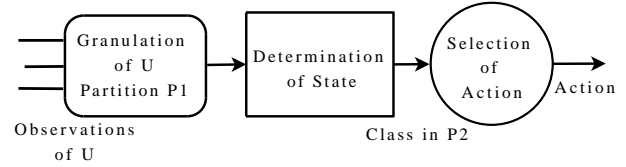


Fig. 2. Basic Paradigm

Examples of this paradigm also occur in the financial environment. Here experts have “rules of thumb” describing what action to take if they know the state of the economy (recession, inflation etc.). However they observe primary data about the economy and from this this must determine the state in order to know what action to take.

10. Non-Specificity of the Partition

The concepts of specificity and its complement non-specificity have been investigated in considerable detail particularly by Yager²² and Klir²³. It is closely related to the idea of generality.

In²³ Klir discusses one measure of non-specificity of a set. In particular if A is a subset of X then Klir's measure of non-specificity is $\log_2(\text{Card}(A))$. In the following we shall use $|A|$ to denote $\text{Card}(A)$. When A is a singleton, $A = \{x\}$ we get $\log_2(|A|) = 0$. For the case when $A = X$, $\log_2(|A|) = \log_2(n)$, where $|X| = n$. Since $\log_2(|A|)$ is monotonic with respect to $|A|$ we see that

$$0 \leq \log_2(|A|) \leq \log_2(n)$$

A normalized version of Klir's measure of non-specificity is $\text{NS}_K(A) = \frac{\log_2(|A|)}{\log_2(n)}$. Here then for any subset $A \subseteq X$ we have $0 \leq \text{NS}_K(A) \leq 1$.

Based on Yager's^{22,23,24,25} work on specificity we obtain an alternative definition for non-specificity

$$\text{NS}_Y(A) = \frac{|A| - 1}{n - 1}$$

This takes values in the unit interval. It attains its maximal value of one for the case when $A = X$ and its minimum value of zero for the case when A is a singleton.

It can be seen that these definitions are cointensive, they are both measures of the granularity of the subset A . The larger the subset the closer the non-specificity is to one. For singletons we get a non-specificity of zero. In using these definitions we will only be considering non-empty sets. The choice between these two definitions of non-specificity depends is context dependent, it depends upon what application and what features are of interest. In the following we shall use the neutral term NS when we have no need to distinguish between these two definitions.

For our purposes we see that the measure of non-specificity can be used to provide a measure of generality. The larger the set A (the more non-specific) the more general the concept the set is representing.

We now briefly consider the extension of the concept of non-specificity from sets to partitions. Our interest here is in quantifying the idea of generality. We are interested in quantifying how general are the classes comprising a partitioning.

Assume $P : \langle A_1, \dots, A_q \rangle$ is a partition of X . We are now interested in the calculation of the non-specificity of P , $NS(P)$. In the following we shall propose such a definition. This definition will make use of the measure of non-specificity of the individual classes in P . The definition we shall use is the following is

$$NS(P) = \sum_{i=1}^q \frac{|A_i|}{n} NS(A_i)$$

We see it is a weighted average of the non-specificities of the component classes in P . The weights are determined by the number of elements in the class.

We see that this definition is independent of the indexing of the classes, each class is treated in the same manner. We also note that it is idempotent, if for all i , $NS(A_i) = a$ then $NS(P) = a$. This if all the classes have the same degree of non-specificity then this is the non-specificity of the partition as a whole. We also note that since each of $NS(A_i) \in [0, 1]$ then $NS(P) \in [0, 1]$.

Let us look at some notable special cases. Consider the case where $P = P^*$, that is $q = 1$ and

$A_1 = X$. We see that $AS(A_1) = 1$ and $|A_1| = n$ and therefore $NS(P^*) = 1$. Thus the non-specificity of a partition consisting of just the whole space is one.

At the opposite extreme is the case when the classes are just singletons P_* , here we have n classes with $A_i = \{x_i\}$. In the case $NS(A_i) = 0$ and hence from the idempotency we get $NS(P) = 0$.

We should note that fact that $\bigcup_{i=1}^q A_i = X$ places some constraints on the possible manifestations of $NS(P)$.

An interesting property holds between the measures of non-specificity of two partitions P_1 and P_2 when $P_1 \subseteq P_2$.

Theorem 4. *If P_1 and P_2 are two partitions such that $P_1 \subseteq P_2$ then $NS(P_1) \leq NS(P_2)$.*

Proof. Assume $P_1 : \langle A_1, \dots, A_q \rangle$ and $P_2 : \langle B_1, \dots, B_r \rangle$. The fact that $P_1 \subseteq P_2$ means that $r \leq q$ and that for each A_i there exists a unique B_j such that $A_i \subseteq B_j$. We shall let $Fam(j)$ be the collection of A_i since that $A_i \subseteq B_j$. Here we have

$$NS(P_1) = \sum_{i=1}^q \frac{|A_i|}{n} NS(A_i)$$

and

$$NS(P_2) = \sum_{j=1}^r \frac{|B_j|}{n} NS(B_j)$$

We can rewrite $NS(P_1)$ as

$$NS(P_1) = \sum_{j=1}^r \left(\sum_{i \in Fam(j)} \frac{|A_i|}{n} NS(A_i) \right)$$

We further note that if $A_i \subseteq B_j$ then $NS(A_i) \leq NS(B_j)$ and hence

$$NS(P_1) \leq \sum_{j=1}^r \left(\sum_{i \in Fam(j)} \frac{|A_i|}{n} NS(B_j) \right)$$

Since $B_j = \bigcup_{i \in Fam(j)} A_i$ and the A_i are disjoint then

$|B_j| = \sum_{i \in Fam(j)} |A_i|$ hence we get desired result

$$NS(P_1) \leq \sum_{j=1}^r \frac{|B_j|}{n} NS(B_j) \leq NS(P_2)$$

□

Let us look at the manifestation of $NS(P)$ for the two cases of specificity of sets, $NS_Y(A)$ and $NS_K(A)$. For the case of NS_Y we get that

$$NS_Y(P) = \sum_{i=1}^q \frac{a_i}{n} \cdot \frac{a_i - 1}{n - 1}$$

where $a_i = |A_i|$. After some algebraic manipulation we have

$$NS_Y(P) = \frac{1}{n-1} \left(\sum_{i=1}^q \frac{a_i^2}{n} - \sum_{i=1}^q \frac{a_i}{n} \right)$$

Since $\sum_{i=1}^q a_i = n$ then

$$\begin{aligned} NS_Y(P) &= \frac{1}{n-1} \left(\sum_{i=1}^q \frac{a_i^2}{n} - 1 \right) \\ &= \frac{n}{n-1} \sum_{i=1}^q \frac{a_i^2}{n^2} - \frac{1}{n-1} \sum_{i=1}^q w_i^2 - \frac{1}{n-1} \end{aligned}$$

Letting $w_i = \frac{a_i}{n}$, the proportion of elements in A_i , we get

$$\begin{aligned} NS_Y(P) &= \frac{n}{n-1} \sum_{i=1}^q w_i^2 - \frac{1}{n-1} \\ &= \frac{1}{n-1} \left(n \sum_{i=1}^q w_i^2 - 1 \right) \end{aligned}$$

We further observe that if n is large then

$$NS_Y(P) \longrightarrow \sum_{i=1}^q w_i^2$$

This is a kind of variance.

For the case where we use $NS_K(A)$ we have

$$NS_K(P) = \sum_{i=1}^q \frac{a_i}{n} \frac{\log_2(a_i)}{\log_2(n)}$$

Again letting $\frac{a_i}{n} = w_i$ we get

$$\begin{aligned} NS_K(P) &= \sum_{i=1}^q w_i \frac{\log_2(nw_i)}{\log_2(n)} \\ &= \frac{1}{\log_2(n)} \sum_{i=1}^q (w_i \log_2(w_i) + w_i \log_2(n)) \end{aligned}$$

Since $\sum_i w_i = 1$ we see

$$NS_K(P) = 1 + \frac{1}{\log_2(n)} \sum_{i=1}^q w_i \log_2(w_i)$$

We see this as a kind of entropy.

In passing we note that an alternative although less sensitive possible measure for the non-specificity of a partition is $NS(P) = \text{Max}_j [NS(A_j)]$. It is easy to show that this has similar properties of the preceding although it is less sensitive.

11. An Application to Data Mining

A common task that occurs in data mining is described in the following. Let X be a class of objects and let $P : \langle B_1, B_2, \dots, B_r \rangle$ be a given partition of the objects based upon some feature V . We call this the target feature.

In addition assume we have a concept hierarchy over X based upon some other features, U . We recall that a concept hierarchy consists of a group of partition of increasing generality. Let P_1, \dots, P_r be a group of partitions on X where $P_i : \langle A_{i1}, A_{i2}, \dots, A_{iq_i} \rangle$. The requirement for these partitions to form a concept hierarchy is that $P_k \subset P_j$ for $k \neq j$. We refer to P_i as the i^{th} level partition. In this case the generality is increasing as the level increases. Concept hierarchies play a fundamental role in the way human's conceptualize the world. A typical example of a concept hierarchy is the following. Let U be the attribute corresponding to an objects home address. A concept hierarchy would be the following

- P_1 is a partition by the *addresses*,
- P_2 is a partition by *neighborhood*,
- P_3 is a partition by *City*,
- P_4 is a partition by *state*,
- P_5 is a partition by *Region*

Our objective here is to build rules predicting the class in P, B_j , to which an object belongs based on its value for the feature U . In order to make these rules human comprehensible we want to use the categories (concepts) associated with the concept hierarchy.

In order to accomplish this task we can consider each level of the hierarchy, P_i . Using the methods described earlier we can find for each level the function g_i that makes the best assignment between the classes in the partitions P_i and the target partition P . Let us denote the value associated with each of the as $\text{Val}(g_i)$. The larger this the better the association. From our preceding discussion we showed that if $P_i \subset P_j$ then the respective assigning functions g_i and g_j are such that $\text{Val}(g_i) \geq \text{Val}(g_j)$. Thus here then the best rules are obtained using the categories in the lowest level of the concept hierarchy P_1 .

However as we noted an important objective in formulating the rules relating U and V is human comprehension and understanding. As reported by Miller in his classic paper¹ human information processing capacity deteriorates if one has to manage to many classes. Thus in order achieve our objective we have a preference for rules using fewer more general concepts for the categories used to describe the values of the variable U . Thus here then we prefer partitions that use more general concepts. Parenthetically we note that an additional benefit of using more general classes for U in formulating our rules is that we can allow less precision in our determination of the values U when using the derived rules. The need for less precision is usually more economical on resources.

As we previously noted the generality of a partitioning can be measured by the its non-specificity. In addition we showed that if P_i and P_j are two partitions such that $P_i \subset P_j$ then $\text{Sp}(P_i) \leq \text{Sp}(P_j)$. In particular we see that generality is increased as we go up the concept hierarchy.

We now see the fundamental conflict associated with our task of trying to build rules to help predict an objects class in the target partition P based upon an observation of its U value. Our desire of attaining correct prediction benefits by using as our partitioning for U one down the bottom of the concept hierarchy while our desire for generalization and human comprehension benefits by going up the concept hierarchy. While we shall not pursue this further here we believe that the measures we have introduced, $\text{Sp}(P_i)$ and $\text{Prog}(P_i, P)$, can play a role in helping

adjudicating this conflict. We also note that in²⁶ we looked at issues closely related to this problem.

12. Weighted Categories

Consider again the task of using the categories in the partition $P_1 : \langle A_1, \dots, A_q \rangle$ to predict an objects class in a target partition $P : \langle B_1, \dots, B_r \rangle$. In the preceding we assumed that there are no costs in predicting an object is in category B_k when it is really in category B_j . In the following we shall briefly investigate the process of determination the best assigning function in the case when there are different payoffs associated with assigning an object to one class when it is in another. In discussing this we shall make considerable use of the matrix C below.

$$\begin{array}{c} B_1 \\ B_k \\ B_r \end{array} \left[\begin{array}{ccc} B_1 & B_j & B_r \\ & C_{k/j} & \\ & & \end{array} \right]$$

In this matrix B_1, \dots, B_r are the classes of the target partition P and $C_{k/j}$ is the payoff for assigning an object to class B_k when it belongs in class B_j . In this situation we assume that positive values are benefits and negative values are costs. We also assume that for each column B_j the payoffs $C_{k/j}$ takes its maximum value for $k = j$, the correct identification.

In addition to this matrix we also have as in the preceding the matrix N where n_{ij} is the number of objects in both A_i and B_j , $n_{ij} = |A_i \cap B_j|$.

Let h be a mapping that assigns each A_i to a class in P , $B_{h(i)}$. The value of this mapping is

$$\text{Value}(h) = \sum_{i=1}^q \left(\sum_{j=1}^r n_{ij} C_{h(i)/j} \right)$$

Our object is now to find the mapping h which gives the maximal value for $\text{Value}(h)$. As before we denote this optimal mapping as g . Calculation of $\text{Value}(h)$ can be decomposed and the determination of $g(i)$ can be independently obtained for each i . In

particular for any i , $g(i) = \mathbf{k}_i^*$ where

$$\sum_{j=1}^r n_{ij} C_{\mathbf{k}_i^*/j} = \mathbf{Max}_{k=1 \text{ to } r} \left[\sum_{j=1}^r n_{ij} C_{k/j} \right]$$

To solve this we shall find the following notation useful. Let $R \cdot A_i$ be a the i^{th} row of N . It is an $1 \times r$ dimension vector. We shall let $R \cdot B_k$ be the k^{th} row of the matrix C . It also is an $1 \times r$ dimension vector. We see that in vector notation $\sum_{j=1}^r n_{ij} C_{k/j} = (R \cdot B_k) \times (R \cdot A_i)^T$. Thus to find $g(i)$ we fix i , $R \cdot A_i$, and go through all $R \cdot B_k$, $k = 1$ to r , to find the one that gives us the biggest inner product with $R \cdot A_i$.

Let us look at a special case of $C_{k/j}$. Consider the case where $C_{k/j} = 1$ for $j = k$ and $C_{k/j} = 0$ for $j \neq k$. In this case for any k we have $\sum_{j=1}^r n_{ij} C_{k/j} = n_{ik}$. Thus here we want $g(i) = k^*$ such that $n_{ik^*} = \mathbf{Max}_k [n_{ik}]$, this was our earlier solution.

13. Conclusion

We investigated a number of measures relating partitions. One class of measures we considered were congruence measures. These measures are used to calculate the similarity between two partitionings. We discussed a number of examples of this type of measure. Another class of measures we investigated were prognostication measures. This type of measure, closely related to a concept of containment between partitions, is useful in indicating how well knowledge of an objects class in one partitioning indicates its class in a second partitioning. We applied our measures to some data mining applications using the structure of the concept hierarchy. We also introduced a measure of the non-specificity of a partition. This measures a feature of a partition related to the granularity or size of the constituent classes of the partition.

References

1. G.A. Miller, "The magical number seven, plus or minus two: Some limitations on our capacity for processing information", *Psychological Review*, **63**, 81–97 (1956).
2. G.A. Miller, "The organization of lexical memory", in: G.A. Talland & N.C. Waugh (Eds.), *The Pathology of Memory*, New York: New York, 1969.
3. A. Kandel, M. Last, and H. Bunke, *Data Mining and Computational Intelligence*, Springer-Verlag: Heidelberg, 2001.
4. O. Kaynak, L.A. Zadeh, B. Turksen, and I.J. Rudas, *Computational Intelligence: Soft Computing and Fuzzy-Neuro Integration Applications*, Springer-Verlag, Berlin, 1998.
5. A.P. Engelbrecht, *Computational Intelligence: An Introduction*, John Wiley & Sons, New York, 2003.
6. M.J.A. Berry and G. Linoff, *Data Mining Techniques*, John Wiley & Sons, New York, 1997.
7. S. Mitra and T. Acharya, *Data Mining: Multimedia. Soft Computing and Bioinformatics*, New York: Wiley, 2003.
8. D. Ruan, G. Chen, E. Kerre, and G. Wets, *Intelligent Data Mining: Techniques and Applications*, Springer-Verlag, Heidelberg, 2005.
9. T.Y. Lin, "Granular computing: From Rough sets and neighborhood systems to information granularization and computing with words", *European Congress on Intelligent Techniques and Soft Computing*, 1602–1606 (1997).
10. T.S. Lin, Y.Y. Yao, and L.A. Zadeh, *Data Mining, Rough Sets and Granular Computing*, Physica-Verlag, Heidelberg, 2002.
11. A. Bargiela and W. Pedrycz, *Granular Computing: An Introduction*, Kluwer Academic Publishers, Amsterdam, 2003.
12. J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, Morgan Kaufmann, San Francisco, 2001.
13. D. Hand, H. Mannila, and P. Smyth, *Principles of Data Mining*, The MIT Press, Cambridge, MA, 2001.
14. L.R.S. da Silva, F. Gomide, and R.R. Yager, "Fuzzy clustering with participatory learning and applications", in: J.V. Oliveira and W. Pedrycz (Eds.) *Advances in Fuzzy Clustering and its Applications*, John Wiley & Sons, Hoboken, NJ, 139–153 (2007).
15. R.R. Yager and F. Petry, "A multi-criteria approach to data summarization using concept ontologies", *IEEE Transactions on Fuzzy Systems*, **14**, 767–780 (2006).
16. Z. Pawlak, *Rough Sets - Theoretical Aspects of Reasoning About Data*, Kluwer, Hingham, MA, 1991.
17. L.A. Zadeh, "Similarity relations and fuzzy orderings", *Information Sciences*, **3**, 177–200 (1971).
18. W.M. Rand, "Objective criteria for the evaluation of clustering methods", *J. Amer. Statis. Ass.*, **66**, 846–850 (1971).
19. L. Hubert, "Comparing partitions", *Journal of Classification*, **2**, 193–218 (1985).
20. F.S. Hillier and G.J. Lieberman, *Introduction to Operations Research*, McGraw Hill, New York, NY, 2005.
21. D.E. Goldberg, *Genetic Algorithms in Search Op-*

- timization and Machine Learning*, Addison-Wesley, Reading, MA, 1989.
22. R.R. Yager, "On measures of specificity", in: O. Kaynak, L.A. Zadeh, B. Turksen, and I.J. Rudas (Eds.), *Computational Intelligence: Soft Computing and Fuzzy-Neuro Integration with Applications*, Springer-Verlag, Berlin, 94–113, 1998.
 23. G.J. Klir, *Uncertainty and Information*, John Wiley & Sons, New York, 2006.
 24. R.R. Yager, "Entropy and specificity in a mathematical theory of evidence", *Int. J. of General Systems*, **9**, 249–260 (1983).
 25. R.R. Yager, "Specificity measures of possibility distributions", Proceedings of the Tenth NAFIPS Meeting, U. of Missouri, Columbia, MO, 240–241 (1991).
 26. R.R. Yager and F.E. Petry, "Evidence resolution using concept hierarchies", *IEEE Transactions on Fuzzy Systems*, (To Appear).