Electronic Notes in Theoretical Computer Science 40 (2002)
URL: http://www.elsevier.nl/locate/entcs/volume40.html 24 pages

Clusters, Concepts, and Pseudometrics

Michael D. Rice [a] and Michael Siff [b]

[a] *Department of Mathematics & Computer Science*
Wesleyan University
Middletown, CT 06459
Email: mrice@wesleyan.edu

[b] *Faculty of Computer Science*
Sarah Lawrence College
Bronxville, New York 10708
Email: msiff@slc.edu

Abstract

The fields of cluster analysis and concept analysis are both used to identify patterns in data. Concept analysis identifies similarities between sets of objects based on their attributes. Cluster analysis groups objects with related characteristics based on some notion of distance.

In this paper, we investigate connections between these two approaches. In particular, for each binary relation defined on a set of objects $\mathcal{O}$ and attributes $\mathcal{A}$, we define distance functions ρ (on the power set of $\mathcal{O}$) and δ (on $\mathcal{O}$).

We prove that ρ and δ are pseudometrics and use them to

- specify a *clustering algorithm* that computes a subset of the concept lattice
- discover new *interpretations* of basic notions in concept analysis.

In particular, we characterize concepts in terms of ρ and characterize a family of concept lattices based on all subsets with a fixed cardinality bound in terms of δ.

Our clustering algorithm differs from the classical algorithms since, first, the values of ρ, not δ, determine which pairs of sets are combined at each level, and second, the clusters defined at each level in the algorithm are generally anti-chains and may not be partitions. Therefore, the analysis of the algorithm depends on the metric-geometry of ρ and is more involved than the analysis of the classical algorithms.

We have developed a software environment that permits the execution of the algorithm on finite relations and the storage and analysis of the resulting clusters. The algorithm has been run on relations generated from a variety of sources ranging from medical research to sporting events. Our results indicate that the number of

iterations of the clustering algorithm is *linear* in the size of $\mathcal{O}$ and produces a *linear* number of clusters that are concepts. Hence the algorithm can be a useful tool for computing a tractable subset of a very large concept lattice.

1 Introduction

The fields of cluster analysis and concept analysis are both used to identify patterns in data. Concept analysis identifies similarities between sets of objects based on their attributes. Cluster analysis groups objects with related characteristics based on some notion of distance. In this paper, we investigate connections between these two approaches.

The framework for concept analysis is a finite set of objects, a finite set of attributes, and a binary relation between the two sets that describes objects according to their attributes. A concept is a maximal rectangle in the relation. The goal of concept analysis is to compute concepts and to analyze them for significant groupings of objects and attributes. However, concept analysis is not a practical tool for identifying patterns in large data sets. First, it can be computationally expensive to compute all of the concepts. Secondly, the number of concepts may be too large to analyze in a reasonable amount of time.

In contrast to concept analysis, cluster analyses provide ways to group objects according to nearest distances (assuming a collection of objects and the ability to compute the distance between any two objects). Cluster analyses tend to efficiently generate a set of clusters that is often proportional to the number of objects. However, it is frequently unclear what distance function to choose in order to perform a cluster analysis that guarantees that the clusters found will be meaningful in some way independent of distance.

The main contribution of this paper is to provide a bridge between concept analysis and traditional hierarchical clustering. The structure of the paper is the following. Section 2 presents an introduction to concept analysis. Section 3 defines two distance functions ρ_R and δ_R based on a binary relation R, presents a proof that they are both pseudometrics (Theorem 3.3), and presents a metric characterization of extents (Theorem 3.5). Section 4 discusses a family of reduced contexts that generate a special class of lattices and presents a metric characterization of the power set lattice (Theorem 4.1). Section 5 describes a hierarchical clustering algorithm based on ρ_R that computes a family of concepts. We prove that the algorithm always terminates (Theorem 5.3) and present some insight into the behavior of the algorithm (Theorem 5.4). We also show that, under a realistic hypothesis, both the number of iterations used by the algorithm and the number of clusters generated are linear functions of the number of objects (Corollary 5.5). Section 6 presents some conclusions and Section 7 discusses related work. Section 8 contains summaries of statistics obtained by running the algorithm on various types of data.

2 Basic Concept Analysis

2.1 Preliminaries

In this section, we present the basic ideas in concept analysis needed for this paper. Many of the ideas in this section are also presented in [8] using a slightly different notation.

For each set S, $\mathcal{P}(S)$ [respectively $\mathcal{P}_{fin}(S)$] denotes the family of subsets [respectively finite subsets] of S. For each $P, Q \subseteq S$, $P \Delta Q$ denotes the symmetric difference of P and Q and for each $P \in \mathcal{P}_{fin}(S)$, $|P|$ denotes the cardinality of P. In general, $\mathcal{O}$ denotes a non-empty finite set of objects, $\mathcal{A}$ denotes a non-empty finite set of attributes, and $R \subseteq \mathcal{O} \times \mathcal{A}$ denotes a relation that is called a *context*. The opposite relation $R^{op} \subseteq \mathcal{A} \times \mathcal{O}$ is defined by $R^{op} = \{(a, x) \mid (x, a) \in R\}$. Often, we view a context graphically by using a 1 to denote that a row-column pair belongs to the context and a 0 otherwise. For example, the following figure shows a context $R \subseteq \mathcal{O} \times \mathcal{A}$, where $\mathcal{O} = \{1, 2, 3, 4\}$ and $\mathcal{A} = \{a, b, c, d, e, f, g, h\}$.

	a	b	c	d	e	f	g	h
1	0	1	0	0	0	0	1	1
2	0	0	1	1	1	1	0	0
3	0	0	1	0	1	0	1	1
4	1	0	1	0	0	0	0	0

Fig. 1.

Given a context $R \subseteq \mathcal{O} \times \mathcal{A}$, define the mappings $\sigma_R : \mathcal{P}(O) \to \mathcal{P}(A)$ and $\tau_R : \mathcal{P}(\mathcal{A}) \to \mathcal{P}(\mathcal{O})$ as follows: for each $X \subseteq \mathcal{O}$ and $Y \subseteq \mathcal{A}$,

$$\sigma_R(X) = \{a \in \mathcal{A} \mid (\forall x \in X)\,(x, a) \in R\}$$
$$\tau_R(Y) = \{x \in \mathcal{O} \mid (\forall a \in Y)\,(x, a) \in R\}.$$

For each $x \in \mathcal{O}$ and $a \in \mathcal{A}$, we write $\sigma_R(x) = \sigma_R(\{x\})$ and $\tau_R(a) = \tau_R(\{a\})$. For the context R in Figure 1, $\sigma_R(1) = \{b, g, h\}$, $\sigma_R(\{1, 3\}) = \{g, h\}$, $\tau_R(g) = \{1, 3\}$, and $\tau_R(\{c, d, e\}) = \{2\}$.

A context R is *strongly-well-formed* if for each $x \neq y \in \mathcal{O}$, $\sigma_R(x)$ is not a subset of $\sigma_R(y)$. (This is a strengthening of the well-formed notion introduced in [14].) A context R is *reduced* if both R and R^{op} are strongly well-formed. For example, the context in Figure 1 is strongly-well-formed, but it is not reduced since $\tau_R(d) \subsetneq \tau_R(e)$.

The mappings σ_R and τ_R satisfy the following properties: for each $X, X' \subseteq \mathcal{O}$ and $Y, Y' \subseteq \mathcal{A}$,

- $X \subseteq X' \Rightarrow \sigma_R(X') \subseteq \sigma_R(X)$
  $Y \subseteq Y' \Rightarrow \tau_R(Y') \subseteq \tau_R(Y)$

- $X \subseteq \tau_R(\sigma_R(X))$
  $Y \subseteq \sigma_R(\tau_R(Y))$.

These properties are equivalent to the assertion that for each $X \subseteq \mathcal{O}$ and $Y \subseteq \mathcal{A}, Y \subseteq \sigma_R(X)$ if and only if $X \subseteq \tau_R(Y)$. This says that the pair (σ_R, τ_R) is a *Galois connection* [4].

Define the mapping $cl_R : \mathcal{P}(O) \to \mathcal{P}(O)$ as follows: for each $X \subseteq \mathcal{O}, cl_R(X) = \tau_R(\sigma_R(X))$. It follows from the above properties that $\sigma_R \circ \tau_R \circ \sigma_R = \sigma_R$ and the following conditions hold:

- $X \subseteq cl_R(X)$,
- $X \subseteq X' \Rightarrow cl_R(X) \subseteq cl_R(X')$,
- $cl_R(cl_R(X)) = cl_R(X)$.

Therefore, cl_R is a closure operator [4]. For each $x \in \mathcal{O}$, we write $cl_R(x) = cl_R(\{x\})$. A subset $X \subseteq \mathcal{O}$ is an *extent* if $X = cl_R(X)$. The collection of extents of R is denoted by *Extents(R)* and for each $X \subseteq \mathcal{O}$, $cl_R(X) = \bigcap\{C \in$ *Extents(R)* $\mid X \subseteq C\}$ is the smallest extent containing X. If $X \subseteq \mathcal{O}$ is an extent, then $Y = \sigma_R(X)$ is an intent and the pair (X, Y) is called a *concept*. For the context R in Figure 1, $cl_R(\{1,3\}) = \tau_R(\sigma_R(\{1,3\})) = \tau_R(g,h) = \{1,3\}$ and $cl_R(\{2,4\}) = \tau_R(\sigma_R(\{2,4\})) = \tau_R(c) = \{2,3,4\}$. Hence $\{1,3\}$ is an extent and the pair $(\{1,3\},\{g,h\})$ is a concept, but $\{2,4\}$ is not an extent.

The set $\mathcal{O}$ is an extent and the empty set $\emptyset$ is an extent if and only if for each $x \in \mathcal{O}, \sigma_R(x) \neq \mathcal{A}$. The set *Extents(R)* with the inclusion order is a lattice where the meet and join of extents $X, X' \subseteq \mathcal{O}$ is given by

$$X \wedge X' = X \cap X' \qquad X \vee X' = cl_R(X \cup X').$$

The lattice *Extents(R)* is equivalent to the notion of a concept lattice based on the pairs $(X, \sigma_R(X))$, but it is more convenient for the work in this paper. The following figure shows the lattice of extents for the context in Figure 1.

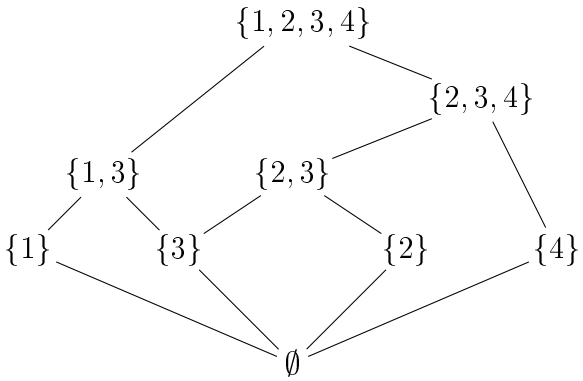

Fig. 2.

By definition, an extent is a set of objects that is maximal in terms of its common attributes. When examining real-world contexts, the family of extents may not include interesting sets of objects that have in common a lack of certain attributes. In order to identify these sets, it is useful to augment a context by adding negative information - namely the complements a' of existing attributes a. (An object has attribute a if and only if it does not have attribute a'.)

Formally, given $R \subseteq \mathcal{O} \times \mathcal{A}$, for each $a \in \mathcal{A}$, let $\mathcal{A}_{ce} = \mathcal{A} \cup \mathcal{A}'$, where $\mathcal{A}' = \{a' \mid a \in \mathcal{A}\}$ and $\mathcal{A} \cap \mathcal{A}' = \emptyset$. Then the context $R_{ce} \subseteq \mathcal{O} \times \mathcal{A}_{ce}$ defined by

$$R_{ce} = R \cup \{(x, a') \in \mathcal{O} \times \mathcal{A}' \mid (x, a) \notin R\}.$$

is called the *complemented extension* of R. (A variant of this idea is found in [14] where enough complemented attributes are added to make the resulting context well-formed.) The complemented extensions of contexts satisfy the following two properties:

- With respect to R_{ce}, the sigma set of a single object contains *one-half* of the attributes in $\mathcal{A}_{ce}$; hence, if R has no pair of identical rows, then R_{ce} is strongly-well-formed.
- *Extents(R)* $\subseteq$ *Extents(R_{ce})* and, in most cases, the latter set is larger.

The following figure shows the complemented extension R_{ce} of the context $R \subseteq \mathcal{O} \times \mathcal{A}$ represented by the first three columns of the figure, where $\mathcal{O} = \{1, 2, 3\}$ and $\mathcal{A} = \{a, b, c\}$.

	a	b	c	a'	b'	c'
1	1	1	0	0	0	1
2	0	0	1	1	1	0
3	0	1	1	1	0	0

Fig. 3.

In this case, *Extents(R)* $= \{\emptyset, \{1\}, \{3\}, \{1, 3\}, \{2, 3\}, \mathcal{O}\}$ and *Extents(R_{ce})* $=$ *Extents(R)* $\cup \{\{2\}\}$.

3 Distance Functions

Given a context R, we would like to define a distance function on *Extents(R)* that captures the notion of two non-empty sets being "close" exactly when they have many attributes in common. More precisely, the distance function should satisfy the following reasonable assumptions:

(1) it does not distinguish a set X from $cl_R(X)$, that is, the distance between these sets is zero.

(2) it assigns the same maximum distance to pairs of sets that have no attributes in common.

For example, the distance defined by $d(X, X') = |\sigma_R(X)\Delta\sigma_R(X')|$ satisfies (1) since $\sigma_R(X) = \sigma_R(cl_R(X))$ but it does not satisfy (2); if $\sigma_R(X) \cap \sigma_R(X') = \emptyset$, then $d(X, X') = |\sigma_R(X)| + |\sigma_R(X')|$ depends on the sizes of the respective sets of attributes. By normalizing the distance d, we can construct an appropriate distance function. For each pair $X, X' \in \mathcal{O}$, define

$$\rho_R(X, X') = \frac{|\sigma_R(X)\Delta\sigma_R(X')|}{|\sigma_R(X) \cup \sigma_R(X')|}$$

if $\sigma_R(X) \cup \sigma_R(X') \neq \emptyset$ and 0 otherwise. Clearly, ρ_R satisfies (1), but it also satisfies (2) since the definition can be rewritten in the form

$$\rho_R(X, X') = 1 - \frac{|\sigma_R(X) \cap \sigma_R(X')|}{|\sigma_R(X) \cup \sigma_R(X')|}$$

if $\sigma_R(X) \cup \sigma_R(X') \neq \emptyset$.

For the context in Figure 1, $\sigma_R(1) = \{b, g, h\}$ and $\sigma_R(3) = \{c, e, g, h\}$, so

$$\rho_R(\{1\}, \{3\}) = \frac{|\sigma_R(1)\Delta\sigma_R(3)|}{|\sigma_R(1) \cup \sigma_R(3)|} = \frac{|\{b, c, e\}|}{|\{b, c, e, g, h\}|} = 3/5.$$

Similarly, since $\sigma_R(\{2, 3\}) = \{c, e\}$, $\rho_R(\{3\}, \{2, 3\}) = \frac{|\{g,h\}|}{|\{c,e,g,h\}|} = 1/2$. On the other hand, $\rho_R(\{1, 4\}, \{1, 2, 3, 4\}) = 0$ since $cl_R(\{1, 4\}) = \{1, 2, 3, 4\}$. This illustrates property (1). Also, $\rho_R(\{1, 3\}, \{2, 3, 4\}) = 1$ since $\sigma_R(\{1, 3\}) = \{g, h\}$ and $\sigma_R(\{2, 3, 4\}) = \{c\}$ have no attributes in common. This illustrates property (2).

The basic properties of ρ_R are summarized in the following result:

Lemma 3.1 *For each context $R \in \mathcal{O} \times \mathcal{A}$ and each pair $X, X' \in \mathcal{O}$,*

(a) $0 \leq \rho_R(X, X') \leq 1$.

(b) $\rho_R(X, X') = 0 \Leftrightarrow \sigma_R(X) = \sigma_R(X') \Leftrightarrow cl_R(X) = cl_R(X')$.

(c) $\rho_R(X, X') = 1 \Leftrightarrow \sigma_R(X) \cap \sigma_R(X') = \emptyset$ *and* $\sigma_R(X) \cup \sigma_R(X') \neq \emptyset$.

Proof. Parts (a) and (c) follow from the definition of ρ_R. To show part (b), suppose $\rho_R(X, X') = 0$. Then by definition, $\sigma_R(X) = \sigma_R(X')$, so $cl_R(X) = \tau_R(\sigma_R(X)) = \tau_R(\sigma_R(X')) = cl_R(X')$. On the other hand, if $cl_R(X) = cl_R(X')$, then $\sigma_R(X) = \sigma_R(cl(X)) = \sigma_R(cl(X')) = \sigma_R(X')$, so $\rho_R(X, X') = 0$. □

To establish that ρ_R and δ_R are credible distance functions (like the Euclidean distance between n-dimensional vectors or the Hamming distance between fixed-length bit strings), we show that ρ_R is a pseudometric on $\mathcal{P}(\mathcal{O})$; namely, it satisfies the following conditions ([12], Chapter 4): for any subsets P, Q and R:

- $\rho_R(P, P) = 0$,

- $\rho_R(P,Q) = \rho_R(Q,P)$
- $\rho_R(P,Q) \leq \rho_R(P,R) + \rho_R(Q,R)$. [triangle inequality]

To establish the triangle inequality, we use the following distance function: for each $P, Q \in \mathcal{P}_{fin}(A)$, define

$$\rho(P,Q) = \begin{cases} \frac{|P \Delta Q|}{|P \cup Q|}, & \text{if } P \cup Q \neq \emptyset, \\ 0, & \text{otherwise.} \end{cases}$$

By definition, $\rho_R(X, X') = \rho(\sigma_R(X), \sigma_R(X'))$.

Lemma 3.2 *The distance function ρ is a metric on $\mathcal{P}_{fin}(A)$* [1].

Proof. By definition, $\rho(P,P) = 0$ and $\rho(P,Q) = \rho(Q,P)$, so we need to establish the triangle inequality

$$\rho(P,Q) \leq \rho(P,R) + \rho(Q,R).$$

It is straightforward to check that the inequality holds if any one of the three sets P, Q or R is empty, so we can assume that P, Q and R are non-empty sets. Define the following quantities:

$$a = |P \cap Q|, \alpha = |P \cup Q|, b = |P \cap R|, \beta = |P \cup R|, c = |Q \cap R|, \gamma = |Q \cup R|$$
$$\theta = |P \cap Q \cap R|, x = |P \setminus (Q \cup R)|, y = |Q \setminus (P \cup R)|, z = |R \setminus (P \cup Q)|.$$

Then the triangle inequality is equivalent to the statement

$$1 - a/\alpha \leq 1 - b/\beta + 1 - c/\gamma,$$

or

$$b/\beta + c/\gamma \leq 1 + a/\alpha. \tag{1}$$

To establish (1), let $k = a - 2\theta$ and $p = b + c + k$. Then $\alpha = x + y + p$, $\beta = x + z + p$, and $\gamma = y + z + p$, so a calculation establishes that

$$b/\beta + c/\gamma - 1 = n/d \tag{2}$$

where

$$n = -(x+z)(y+z) - (b+k)(x+z) - (c+k)(y+z) - kp$$
$$d = (x+z+p)(y+z+p).$$

By definition, $a, b, c \geq q$, so $a + k, b + k, c + k \geq 0$. Without loss of generality, assume that $p = b+c+k > 0$; otherwise, $0 \leq \theta \leq \max\{a,b,c\} \leq a+b+c-2\theta = p \leq 0$, so $a = b = c = \theta = 0$ and (1) holds. We claim that n and d satisfy the following inequalities:

(3) $n \leq ap$

(4) $d \geq p(x+y+p)$.

[1] Most likely, this statement is found in the literature, but we don't have a reference for the result.

To establish (3), note that $n = -(x+z)(y+z) - (b+k)(x+z) - (c+k)(y+z) - kp$, so $n \leq -kp \leq ap$ since $x, y, z \geq 0$, $p > 0$, and $a+k, b+k, c+k \geq 0$.

To establish (4), note that

$$d = (x+z+p)(y+z+p) \geq p(x+y+2z) + p^2 \geq x+y+p$$

since $x, y, z \geq 0$ and $p > 0$. It follows from (2), (3) and (4) that

$$b/\beta + c/\gamma - 1 = n/d \leq a/(x+y+p) = a/\alpha$$

which establishes (1). □

In the sequel, the following restriction δ_R of the distance function ρ_R to the singleton subsets of $\mathcal{O}$ is also used; for each $x, y \in \mathcal{O}$, define $\delta_R(x,y) = \rho_R(\{x\}, \{y\})$.

Theorem 3.3 *The distance functions ρ_R and δ_R are pseudometrics on $\mathcal{P}(\mathcal{O})$ and $\mathcal{O}$, respectively. Therefore, ρ_R is a metric on Extents(R).*

Proof. The first statement follows from Lemma 3.2 and the equality $\rho_R(X, X') = \rho(\sigma_R(X), \sigma_R(X'))$. Hence, the second statement follows from Lemma 3.1(b).□

By definition, for each context $R \in \mathcal{O} \times A$ and $X \in \mathcal{O}, \rho_R(X, cl_R(X)) = 0$. Hence ρ_R is a metric if and only if *Extents(R)* $= \mathcal{P}(\mathcal{O})$. Similarly, δ_R is a metric if and only if each singleton subset of $\mathcal{O}$ is an extent. In fact, this condition is equivalent to saying that R is strongly-well-formed: Suppose each singleton set is an extent and $\sigma_R(x) \subseteq \sigma_R(y)$ for $x, y \in \mathcal{O}$. Then $y \in \tau_R(\sigma_R(y)) \subseteq \tau_R(\sigma_R(x)) = cl_R(x) = \{x\}$, so $x = y$; hence R is strongly-well-formed. Conversely, suppose R is strongly-well-formed and $y \in cl_R(x)$. Then $\sigma_R(x) = \sigma_R(cl_R(x)) \subseteq \sigma_R(y)$, so $x = y$; hence $cl_R(x) = \{x\}$ is an extent.

The pseudometric ρ_R also satisfies the following inequality with respect to unions. This result will be used for analyzing the clustering algorithm in Section 5.

Lemma 3.4 *For each $X, X' \in \mathcal{O}, \rho_R(X, X \cup X') \leq \rho_R(X, X')$ with equality if and only if $\sigma_R(X') \subseteq \sigma_R(X)$.*

Proof. By definition, $\rho_R(X, X') = \rho(\sigma_R(X), \sigma_R(X'))$ and since $\sigma_R(X \cup X') = \sigma_R(X) \cap \sigma_R(X')$,

$$\rho_R(X, X \cup X') = \rho(\sigma_R(X), \sigma_R(X \cup X')) = \rho(\sigma_R(X), \sigma_R(X) \cap \sigma_R(X')),$$

so the inequality is a consequence of the following result:

Claim: *For each $P, Q \in \mathcal{P}_{fin}(A), \rho(P, P \cap Q) \leq \rho(P, Q)$ and equality holds if and only if $Q \subseteq P$.*

To establish this fact, note that by definition,

$$\rho(P, P\cap Q) = \begin{cases} \frac{1-|P\cap(P\cap Q)|}{|P\cup(P\cap Q)|} = \frac{1-|P\cap Q|}{|P|} & \text{if } P \neq \emptyset, \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

and

$$\rho(P, Q) = \begin{cases} \frac{1-|P\cap Q|}{|P\cup Q|} & \text{if } P \cup Q \neq \emptyset \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

Therefore, if $P = Q = \emptyset$, then $\rho(P, P \cap Q) = \rho(P, Q) = 0$. Otherwise, $P \cup Q \neq \emptyset$, so it follows from (5), (6) and the inequality $|P| \leq |P \cup Q|$ that $\rho(P, P\cap Q) \leq \rho(P, Q)$ with equality if and only if $|P| = |P\cup Q|$ or, equivalently, $Q \subseteq P$. □

The final result in this section provides a characterization of extents in terms of ρ_R.

Theorem 3.5 *Suppose $C \subsetneq \mathcal{O}$ and $n = |\sigma_R(C)|$. Then C is an extent if and only if*

$$n > 0 \quad \text{and} \quad (\forall p \notin C)\ \sigma_R(p) = \emptyset \text{ or } \rho_R(p, C) > 1 - \frac{n}{|\sigma_R(p)|}. \tag{7}$$

In this case, for each $x \in C$,

$$\rho_R(x, C) = \begin{cases} 1 - \frac{n}{|\sigma_R(x)|} & \text{if } \sigma_R(x) \neq \emptyset \\ 0 & \text{otherwise.} \end{cases}$$

Proof. We use the following fact to show that every extent satisfies (7):

$$\text{If } A \text{ and } B \text{ are non} - \text{empty finite sets, then } A \subseteq B \Leftrightarrow \frac{|A|}{|B|} \leq \frac{|A\cap B|}{|A\cup B|}. \tag{8}$$

Suppose C is an extent. If $n = 0$, then $C = \tau_R(\sigma_R(C)) = \tau_R(\emptyset) = \mathcal{O}$, which is a contradiction; hence $n > 0$. Suppose $p \in \mathcal{O} \setminus C$. If $\sigma_R(p) \neq \emptyset$, then $A = \sigma_R(C)$ is not a subset of $B = \sigma_R(p)$ [otherwise, $A \subseteq B \Rightarrow p \in \tau_R(B) \subseteq \tau_R(A) = cl_R(C) = C$]. Hence, it follows from (8) that

$$r = \frac{|\sigma_R(C) \cap \sigma_R(p)|}{|\sigma_R(C) \cup \sigma_R(p)|} < \frac{n}{|\sigma_R(p)|}$$

so

$$\rho_R(p, C) = 1 - r > 1 - \frac{n}{|\sigma_R(p)|}.$$

Conversely, suppose (7) holds and let $p \in cl_R(C)$. Then $\sigma_R(cl_R(C)) = \sigma_R(C) \subseteq \sigma_R(p)$ and $\sigma_R(p) \neq \emptyset$ since $n > 0$. Therefore, by definition, $\rho_R(p, C) = 1 - \frac{n}{|\sigma_R(p)|}$, so by (7), $p \in C$. Hence, $cl_R(C) = C$, so C is an extent.

Finally, if $x \in C$, then $\sigma_R(C) \subseteq \sigma_R(x)$, so if $\sigma_R(x) = \emptyset$, then by definition $\rho_R(x, C) = 0$. If $\sigma_R(x) \neq \emptyset$, then $\rho_R(x, C) = 1 - \frac{|\sigma_R(x)\cap\sigma_R(C)|}{|\sigma_R(x)\cup\sigma_R(C)|} = 1 - \frac{n}{|\sigma_R(x)|}$. □

Theorem 3.5 has a particularly nice formulation if R is the complemented extension of a context on $\mathcal{O} \times A$, where $|A| = N$. In this case, $|\sigma_R(x)| = N$

for each $x \in \mathcal{O}$, so the result can be restated in the following form where the value $1 - \frac{n}{N}$ is independent of the points p or x selected.

Suppose $C \subseteq \mathcal{O}$ and $|\sigma_R(C)| = n$. Then C is an extent if and only if

$$n > 0 \quad \text{and} \quad (\forall p \notin C)\ \rho_R(p, C) > 1 - \frac{n}{N}. \tag{9}$$

In this case, for each $x \in C$, $\rho_R(x, C) = 1 - \frac{n}{N}$.

4 Power Set Contexts

This section will investigate the structure of the reduced contexts whose family of extents are the k-n-subsets lattices. The main result of the section (Theorem 4.1) establishes a precise correspondence between a special restriction on the pseudometric δ_R and the *Extents(R)* forming the power-set lattice.

Theorem 4.1 *The following statements are equivalent for a reduced context $R \subseteq \mathcal{O} \times \mathcal{A}$, where $n = |\mathcal{O}| = |\mathcal{A}|$:*

(a) $Extents(R) = \mathcal{P}(\mathcal{O})$.

(b) *For each $x \neq y \in \mathcal{O}$, $\delta_R(x, y) = \frac{2}{n}$.*

(c) *For each $S, T \subseteq \mathcal{O}$,*

$$\rho_R(S,T) = \begin{cases} \frac{|S|+|T|-2|S\cap T|}{n-|S\cap T|} & \text{if } S \cap T \subset \mathcal{O} \\ \\ 0 & \text{otherwise.} \end{cases}$$

As an illustration, for $n = 4$, the following reduced context satisfies condition (b).

	a	b	c	d
1	1	1	1	0
2	1	1	0	1
3	1	0	1	1
4	0	1	1	1

Fig. 4.

Given a finite set of objects $\mathcal{O}$ and $1 \leq k < n = |\mathcal{O}|$, we will call $\mathcal{P}_k(\mathcal{O}) = \{S \subseteq \mathcal{O} \mid |S| \leq k\} \cup \{\mathcal{O}\}$ the $k-n$-subsets lattice. For example, $\mathcal{P}_1(\mathcal{O})$ consists of the singleton subsets, $\emptyset$, and $\mathcal{O}$ and $\mathcal{P}_{n-1}(\mathcal{O})$ is the power-set lattice $\mathcal{P}(\mathcal{O})$.

The proof of Theorem 4.1 uses two auxiliary lemmas. The first lemma says that if a reduced context generates a $k - n$-subsets lattice, then the sizes of the sigma sets are completely determined.

Lemma 4.2 *Suppose R is a reduced context, $1 \leq k < n = |\mathcal{O}|$, and Extents(R) is the $k - n$-subsets lattice. For each $S \subseteq \mathcal{O}$,*

$$|\sigma_R(S)| = \begin{cases} C(n - |S|, k - |S|) & \text{if } |S| \leq k \\ \\ 0 & \text{otherwise.} \end{cases}$$

[Here $C(m, l)$ = number of ways to choose l items from m items].

Proof. We claim that

$$\text{For each } a \in \mathcal{A}, |\tau_R(a)| = k. \tag{10}$$

Since R^{op} is strongly-well-formed, $\tau_R(a) \neq \mathcal{O}$. Since $\tau_R \circ \sigma_R \circ \tau_R = \tau_R$, each $\tau_R(a)$ is an extent, so by the hypothesis $|\tau_R(a)| \leq k$. Choose $T \subseteq \mathcal{O}$ such that $\tau_R(a) \subseteq T$ and $|T| = k$. Since T is a proper extent, $\sigma_R(T) \neq \emptyset$, so there exists $b \in \sigma_R(T)$. Hence, $\tau_R(a) \subseteq T \subseteq \tau_R(b)$, so $a = b$ since R^{op} is strongly-well-formed. Therefore, $T = \tau_R(a)$ and $|\tau_R(a)| = k$.

To prove Lemma 4.2, notice that if $a \in \sigma_R(S)$, then $S \subseteq \tau_R(a)$, so it follows from (10) that $|S| \leq k$. Hence if $|S| > k$, then $|\sigma_R(S)| = 0$. If $|S| \leq k$, define

$$\mathbf{F} = \{T \subseteq \mathcal{O} \mid S \subseteq T \text{ and } |T| = k\}.$$

Choose $T \in \mathbf{F}$. Since T is both a proper subset of $\mathcal{O}$ and an extent, there exists $a_T \in \sigma_R(T)$. Therefore, $T \subseteq \tau_R(a_T)$, and by (10) $|\tau_R(a_T)| = k$, so $T = \tau_R(a_T)$.

Let $A = \{a_T \mid T \in \mathbf{F}\}$. The preceding work shows that $T \mapsto a_T$ is a one-to-one mapping, so $|A| = |\mathbf{F}|$. We claim that

$$\sigma_R(S) = A. \tag{11}$$

By definition, $S \subseteq \cap\mathbf{F} = \cap\{\tau_R(a_T) \mid T \in \mathbf{F}\} = \tau_R(A)$, so $A \subseteq \sigma_R(S)$. On the other hand, if $a \in \sigma_R(S)$, then $S \subseteq \tau_R(a)$, so by (10), $|\tau_R(a)| = k$. By definition, $T = \tau_R(a)$ for some $T \in \mathbf{F}$. Since $T \mapsto a_T$ is a one-to-one mapping, $a = a_T$; hence $a \in A$, so $\sigma_R(S) \subseteq A$. This establishes (11).

Since $n = |\mathcal{O}|$, it follows from (11) that $|\sigma_R(S)| = |A| = |\mathbf{F}| = C(n - |S|, k - |S|)$. □

The second lemma establishes that if δ_R is a metric that assumes only one non-zero value, then every set of objects is an extent.

Lemma 4.3 *Suppose R is a reduced context and $n = |\mathcal{O}| = |\mathcal{A}|$. If $\delta_R(x, y) = \frac{2}{n}$ for each pair of distinct points $x, y \in \mathcal{O}$, then Extents$(R) = \mathcal{P}(\mathcal{O})$.*

Proof. The case $n = 2$ can be checked directly, so without loss of generality, assume that $n > 2$. For each $x \neq y \in \mathcal{O}$, define $s_x = |\sigma_R(x)|$ and $s_{x,y} = |\sigma_R(\{x, y\})|$. By assumption, for each $x \neq y \in \mathcal{O}$,

$$\delta_R(x, y) = 1 - \frac{1}{(F - 1)} = \frac{2}{n}$$

where

$$F = \frac{|\sigma_R(x)| + |\sigma_R(y)|}{|\sigma_R(x) \cap \sigma_R(y)|} = \frac{s_x + s_y}{s_{x,y}}.$$

Rearranging terms, we obtain the following set of equalities:

(12) $$\textit{For each } x \neq y,\ (n-2)(s_x + s_y) = 2(n-1)s_{x,y}.$$

We claim that

(13) $$\textit{For each } x \in \mathcal{O},\ s_x = n-1.$$

Choose $y, z \in \mathcal{O} \setminus \{x\}$ with $y \neq z$. By (12),

(1) $(n-2)(s_x + s_y) = 2(n-1)s_{x,y}$.

(2) $(n-2)(s_x + s_z) = 2(n-1)s_{x,z}$.

(3) $(n-2)(s_y + s_z) = 2(n-1)s_{y,z}$.

Subtracting (3) from (2), we obtain

(4) $(n-2)(s_x - s_y) = 2(n-1)(s_{x,z} - s_{y,z})$.

Adding (1) and (4), we obtain $(n-2)s_x = (n-1)p$, where $p = s_{x,y} + s_{x,z} - s_{y,z}$.

Since $\gcd(n-2, n-1) = 1, (n-2)|p$; hence $s_x = (n-1)q$ for some integer q. Since R is strongly-well-formed, $0 < s_x = (n-1)q < |\mathcal{A}| = n$. Therefore, $q = 1$, so $s_x = n-1$. This establishes (13).

Since R is strongly-well-formed and $n = |\mathcal{O}| = |\mathcal{A}|$, it follows from (13) that

(a) For each $x \in \mathcal{O}$, there exists $a_x \in \mathcal{A}$ such that $\sigma_R(x) = \mathcal{A} \setminus \{a_x\}$.

(b) For each $x \in \mathcal{O}, \tau_R(a_x) = \mathcal{O} \setminus \{x\}$.

(c) $\mathcal{A} = \{a_x \mid x \in \mathcal{O}\}$.

Based on (a) and (c), for each $S \subseteq \mathcal{O}$,

$$\sigma_R(S) = \cap\{\mathcal{A} \setminus \{a_x\} \mid x \in S\} = \mathcal{A} \setminus \{a_x \mid x \in S\} = \{a_x \mid x \notin S\}.$$

Therefore, by (b)

$$\tau_R(\sigma_R(S)) = \cap\{\mathcal{O} \setminus \{x\} \mid x \notin S\} = S,$$

which establishes $\mathcal{P}(\mathcal{O}) = \textit{Extents(R)}$. □

Proof of Theorem 4.1 The implication (b) ⇒ (a) follows from Lemma 4.3 and it is straightforward to verify (c) ⇒ (b) using the fact that $\delta_R(x, y) = \rho_R(\{x\}, \{y\})$. To show (a) ⇒ (c), we use Lemma 4.2 with $k = n-1$. In this case, for each $S \subseteq \mathcal{O}$,

(14) $$|\sigma_R(S)| = C(n - |S|, n - 1 - |S|) = n - |S|.$$

Suppose $S, T \subseteq \mathcal{O}$ and $S \cap T \subsetneq \mathcal{O}$. By definition,

$$\rho_R(S, T) = 1 - \frac{|\sigma_R(S \cup T)|}{|\sigma_R(S) \cup \sigma_R(T)|} = 1 - \frac{1}{(F-1)}$$

where $F = \frac{|\sigma_R(S)|+|\sigma_R(T)|}{|\sigma_R(S\cup T)|}$.

Therefore, by (14),

$$F = \frac{2n - |S| - |T|}{n - |S \cup T|} = \frac{2n - |S| - |T|}{n - |S| - |T| - |S \cap T|},$$

which establishes

$$\rho_R(S,T) = \frac{|S| + |T| - 2|S \cap T|}{n - |S \cap T|}. \qquad \square$$

5 Clustering Algorithm

In this section, we present a hierarchical clustering algorithm based on the set pseudometric ρ_R and analyze some of its characteristics. The clusters generated by the algorithm are a family of subsets of *Extents(R)* that is usually significantly smaller than the entire collection.

Assume that $\mathcal{O}$ and $\mathcal{A}$ are finite non-empty sets and $R \subseteq \mathcal{O} \times \mathcal{A}$ is a context. Define the join operator $\psi: Extents(R) \times Extents(R) \to Extents(R)$ by $\psi(C, C') = cl_R(C \cup C') = C \vee C'$.

The following pseudocode specifies the clustering algorithm:

$clusters \leftarrow \mathbf{C} \leftarrow \{cl_R(x) \mid x \in \mathcal{O}\}$ – 1
`while` $(|\mathbf{C}| > 1)$ `do` – 2
{
  $m \leftarrow \min\{\rho_R(C, C') \mid C \neq C' \in \mathbf{C}\}$ – 3
  $\mathbf{E} \leftarrow \{(C, C') \in \mathbf{C} \times \mathbf{C} \mid \rho_R(C, C') = m\}$ – 4
  $\mathbf{V} \leftarrow \{C \in \mathbf{C} \mid (\exists C' \in \mathbf{C})\ (C, C') \in \mathbf{E}\}$ – 5
  $\mathbf{L} \leftarrow \psi[\mathbf{E}]$ – 6
  $\mathbf{C} \leftarrow (\mathbf{C} \setminus \mathbf{V}) \cup \mathbf{L}$ – 7
  $clusters \leftarrow clusters \cup \mathbf{L}$ – 8
}

The algorithm proceeds from the bottom up using the worklist $\mathbf{C}$. Initially, $\mathbf{C}$ consists of the atoms (the closures of singleton objects) on line 1. The algorithm continues by considering certain pairs of clusters from the current worklist $\mathbf{C}$ and using these pairs to update $\mathbf{C}$ and the overall set of clusters. This is done until the cluster $\mathcal{O}$ is obtained. On each iteration, if $\mathbf{C}$ has more than one member, the following four items are computed on lines 3 – 6:

- the minimum distance m between any pair of distinct clusters in $\mathbf{C}$ (by Theorem 3.3, $(Extents(R), \rho_R)$ is a metric space, so $m > 0$)
- the set $\mathbf{E}$ consisting of all pairs of clusters that are distance m apart (viewed as edges of a graph)
- the set $\mathbf{V}$ of clusters that participate in any pair in $\mathbf{E}$ (vertices induced by the edges of the graph)

– the set $\mathbf{L}$ consisting of the join of each pair in $\mathbf{E}$ (labels on the edges of the graph)

On line 7, the worklist $\mathbf{C}$ is updated by (i) removing any clusters in $\mathbf{C}$ that belong to $\mathbf{V}$ and (ii) adding the members of $\mathbf{L}$. On line 8, the clusters in L are also added to the total clusters. The loop terminates when $\mathbf{C}$ contains exactly one cluster. In that case, by Lemma 5.1 below, the cluster must be $\mathcal{O}$.

The following example illustrates the algorithm. Suppose R is the context in Figure 1. Each singleton set is an extent, so $\mathbf{C}$ is initialized to $\{\{1\},\{2\},\{3\},\{4\}\}$. The minimum distance $m = 0.6$ is obtained with the pair $(\{1\},\{3\})$. Therefore, on the first iteration of the loop, $\mathbf{C}$ is updated to $\{\{2\},\{4\},\{1,3\}\}$. On the second iteration, $m = 0.8$ is obtained with the pair $(\{2\},\{4\})$. Since $cl(\{2,4\}) = \{2,3,4\}$, $\mathbf{C}$ is updated to $\{\{1,3\},\{2,3,4\}\}$. On the next iteration, $m = 1$ and $\mathbf{C} = \{\mathcal{O}\}$ is obtained, so the algorithm terminates with clusters $= \{\{1\},\{2\},\{3\},\{4\},\{1,3\},\{2,3,4\},\{1,2,3,4\}\}$, which represents all non-empty extents with the exception of $\{2,3\}$.

One way in which the algorithm presented here differs from traditional hierarchical clustering algorithms ([1], [10]) is the way in which ties are handled. On each iteration, pairs of clusters that are a minimum distance apart are identified. A tie occurs if two or more pairs of clusters are each a minimum distance apart. Traditionally, ties are broken arbitrarily: one of the closest pairs is considered to be the basis for the new cluster. In our algorithm, we add clusters based on all ties to avoid non-determinism and to increase the chance of finding potential clusters of interest. This decision means that the analysis of the algorithm is significantly more involved than the analysis of classical algorithms.

To analyze the algorithm, we use the graph-theoretic interpretations introduced in the description of the algorithm: $\mathbf{V}$ denotes the vertices in the subgraph induced by the edges $\mathbf{E}$ and $\mathbf{L}$ denotes the labels on the edges. For instance, in the preceding example, on the first iteration, $\mathbf{V} = \{1,3\}, \mathbf{E} = \{(\{1\},\{3\})\}$, and $\mathbf{L} = \{\{1,3\}\}$. For each iteration k in the main loop of the algorithm (lines 2 – 8), let $\mathbf{V}_k, \mathbf{E}_k, \mathbf{L}_k$, and $\mathbf{C}_{k+1}$ denote the sets $\mathbf{V}$, $\mathbf{E}$, $\mathbf{L}$ and $\mathbf{C}$, respectively, and let m_k denote m. The cardinalities of the respective families are denoted by $v_k = |\mathbf{V}_k|, e_k = |\mathbf{E}_k|, l_k = |\mathbf{L}_k|$, and $c_k = |\mathbf{C}_k|$.

Using this notation,

$$
\begin{aligned}
m_k &= \min\{\rho_R(C,C') \mid C \neq C' \in C_k\} \\
\mathbf{E}_k &= \{(C,C') \in \mathbf{C}_k \times \mathbf{C}_k \mid \rho_R(C,C') = m_k\} \\
\mathbf{V}_k &= \{C \in \mathbf{C} \mid \exists C' \in \mathbf{C} \text{ such that } (C,C') \in \mathbf{E}_k\} \\
\mathbf{L}_k &= \psi[Ek] = \{C \vee C' \mid (C,C') \in \mathbf{E}_k\} \\
\mathbf{C}_{k+1} &= (\mathbf{C}_k \setminus \mathbf{V}_k) \cup \mathbf{L}_k.
\end{aligned}
$$

Lemma 5.1 *For each iteration $k, \mathbf{C}_k$ is a cover of the set $\mathcal{O}$. Therefore, if the algorithm terminates on iteration K, then $\mathbf{C}_K = \{\mathcal{O}\}$.*

Proof. We use induction on the iteration k. Since $\{x\} \subseteq cl_R(\{x\})$ for each

$x \in \mathcal{O}$, $\mathbf{C}_1$ is a cover of $\mathcal{O}$. Suppose C_k is a cover of $\mathcal{O}$. Let $x \in \mathcal{O}$ and choose $C \in \mathbf{C}_k$ that contains x. If $C \notin \mathbf{V}_k$, then $C \in \mathbf{C}_{k+}1$, so x belongs to a member of $\mathbf{C}_{k+1}$. If $C \in \mathbf{V}_k$, then by definition there exists $D \in \mathbf{C}_k$ such that $(C, D) \in \mathbf{E}_k$. Hence $x \in C \vee D \in \mathbf{C}_{k+1}$, so $\mathbf{C}_{k+1}$ is a cover of $\mathcal{O}$. □

The next result demonstrates how termination occurs for a wide class of contexts.

Lemma 5.2

(a) *If $m_k = 1$ on iteration k, then the algorithm terminates on iteration $k+1$.*

(b) *If the algorithm terminates on iteration K and $\sigma_R(\mathcal{O}) = \emptyset$, then $m_{K-1} = 1, \mathbf{C}_{K-1} = \mathbf{V}_{K-1}$, and $(\mathbf{V}_{K-1}, \mathbf{E}_{K-1})$ is a complete graph.*

Proof. (a) By Lemma 3.1(a), $\rho_R \leq 1$, so by the assumption $m_k = 1$, $\mathbf{V}_k = \mathbf{C}_k$. Since $\rho_R(C, C') = 1$ for each $(C, C') \in \mathbf{E}_k$, then by Lemma 3.1(c), $\sigma_R(C) \cap \sigma_R(C') = \emptyset$. Hence

$$\psi(C, C') = \tau_R(\sigma_R(C \cup C')) = \tau_R(\sigma_R(C) \cap \sigma_R(C')) = \tau_R(\emptyset) = \mathcal{O}$$

so $\mathbf{L}_k = \{\mathcal{O}\}$. Therefore, $\mathbf{C}_{k+1} = (\mathbf{C}_k \setminus \mathbf{V}_k) \cup \mathbf{L}_k = \{\mathcal{O}\}$.

(b) Choose $(A, B) \in \mathbf{E}_{K-1}$. Since $\mathbf{C}_K = \{\mathcal{O}\}$, $A \vee B = \mathcal{O}$, so $\sigma_R(A) \cap \sigma_R(B) = \sigma_R(A \vee B) = \sigma_R(\mathcal{O}) = \emptyset$. Hence by Lemma 3.1(c), $m_{K-1} = \rho_R(A, B) = 1$, so each distinct pair $C, D \in \mathbf{C}_{K-1}$ satisfies $\rho_R(C, D) = 1$. Therefore, $\mathbf{C}_{K-1} = \mathbf{V}_{K-1}$ and $(\mathbf{V}_{K-1}, \mathbf{E}_{K-1})$ is a complete graph. □

The next result establishes that the algorithm always terminates.

Theorem 5.3

(a) *For each iteration k, $\mathbf{C}_k \setminus \mathbf{C}_{k+1} \neq \emptyset$.*

(b) *For each $m \neq n$, $\mathbf{C}_m \neq \mathbf{C}_n$; hence the algorithm terminates.*

Proof. (a) Choose $D \in \mathbf{V}_k$ such that $|D| = \min\{|C| \mid C \in \mathbf{V}_k\}$. By definition, $D \notin \mathbf{L}_k$, so $D \in \mathbf{C}_k \setminus \mathbf{C}_{k+1}$.

(b) Suppose there exist integers $1 \leq m < n$ such that $\mathbf{C}_m = \mathbf{C}_n$ and define

$$\mathbf{C} = \cup\{\mathbf{C}_k \mid m \leq k \leq n\}$$
$$\mathbf{C}^* = \cap\{\mathbf{C}_k \mid m \leq k \leq n\}.$$

We claim that $\mathbf{C} = \mathbf{C}^*$. If $\mathbf{C} \setminus \mathbf{C}^* \neq \emptyset$, choose $P \in \mathbf{C} \setminus \mathbf{C}^*$ such that

$$|P| = \min\{|P| \mid P \in \mathbf{C} \setminus \mathbf{C}^*\}. \tag{15}$$

Then the following assertion holds:

$$\textit{There exists } m \leq k < n \textit{ such that } P \in \mathbf{C}_{k+1} \setminus \mathbf{C}_k. \tag{16}$$

[Define $p = \min\{m \leq j < n \mid P \in \mathbf{C}_j\}$ and $q = \max\{m \leq j < n \mid P \notin \mathbf{C}_j\}$. If $m < p$, then (16) holds with $k = p - 1$. If $m = p$, then $\mathbf{C}_m = \mathbf{C}_n$ implies that $q < n$, so (16) holds with $k = q$.]

Based on (16), there exist A_0, B_0 such that $(A_0, B_0) \in \mathbf{E}_k$ and $P = A_0 \vee B_0$. Since $A_0 \in \mathbf{C}_k, A_0 \in \mathbf{C}$, and since $P \notin \mathbf{C}_k$, $A_0 \subsetneq P$, so by (15), $A_0 \in \mathbf{C}^*$.

In particular, $A_0 \in \mathbf{C}_{k+1} \cap \mathbf{V}_k \subseteq \mathbf{L}_k$, so there exist A_1 and B_1 such that $(A_1, B_1) \in \mathbf{E}_k$ and $A_0 = A_1 \vee B_1$. Without loss of generality, assume that $A_1 \subsetneq A_0$. Since $A_1 \in \mathbf{C}_k, A_1 \in \mathbf{C}$, so it follows from (15) that $A_1 \in \mathbf{C}^*$; hence $A_1 \in \mathbf{C}_{k+1} \cap \mathbf{V}_k \subseteq \mathbf{L}_k$. Continuing in this manner, we construct an infinite sequence $\ldots \subsetneq A_2 \subsetneq A_1 \subsetneq A_0$, which contradicts the fact that $\mathcal{O}$ is a finite set. Hence, $\mathbf{C} \setminus \mathbf{C}^* = \emptyset$.

Since $\mathbf{C} = \mathbf{C}^*$, $\mathbf{C}_m \subseteq \mathbf{C}_{m+1}$, which contradicts part (a). Therefore, $\mathbf{C}_m \neq \mathbf{C}_n$ for each $m \neq n$, so by Lemma 5.1, $\{\mathbf{C}_m\}$ is a family of distinct covers. Since $\mathcal{O}$ is a finite set, it has only a finite number of covers; hence the algorithm terminates. □

The proof of Theorem 4 5.3 does not provide any reasonable estimate for either the number of iterations or the number of clusters. In particular, the analysis of the complexity of the algorithm is complicated by the fact that the covers $\mathbf{C}_k$ may not be partitions. This is a consequence of how ties are handled: at each iteration, the joins of all nearest pairs are added to the set of clusters.

In certain cases, we can predict the number of iterations used by the algorithm and the set of clusters generated. For example, consider the following reduced contexts that generate the $k-n$-subsets lattices discussed in Section 4.

Suppose $n = |\mathcal{O}|, 1 \leq k < n, \mathcal{A} = \{S \subseteq \mathcal{O} \mid |S| = k\}$, and define the reduced context

$$R = \{(x, S) \in \mathcal{O} \times \mathcal{A} | x \in S\}.$$

Then for each $S \subseteq \mathcal{O}, |\sigma_R(S)| = C(n - |S|, k - |S|)$ if $|S| \leq k$, and is 0 otherwise. Furthermore, the clustering algorithm terminates in $k+1$ iterations with clusters $= Extents(R) \setminus \{\emptyset\} = \mathcal{P}_k(\mathcal{O}) \setminus \{\emptyset\}$ and on iteration $1 \leq h \leq k$,

- $\mathbf{C}_h = \mathbf{V}_h = \{S \subseteq \mathcal{O} \mid |S| = h\}$,
- $\mathbf{E}_h = \{(S, T) \mid |S| = |T| = h \text{ and } |S \cap T| = h - 1\}$,
- $m_h = \frac{2(n-k)}{2n-k-h}$.

Based on appropriate assumptions about either the number of nearest pairs on each iteration or the structure of the covers $\{\mathbf{C}_k\}$, we can establish linear bounds on both the number of iterations and the number of clusters. The key insight is suggested by the following result.

Theorem 5.4

(a) *For each iteration* $k, (\mathbf{C}_k \setminus \mathbf{V}_k) \cap \mathbf{L}_k = \emptyset$*; hence* $c_{k+1} = c_k + l_k - v_k$.

(b) *If the algorithm terminates on iteration* K*, then* $\Sigma\{l_k - v_k \mid 1 \leq k < K\} = 1 - c_1$.

Proof. (a) Suppose $L = A \vee B \in \mathbf{C}_k \setminus \mathbf{V}_k$, where $(A, B) \in \mathbf{E}_k$. Without loss of generality, suppose $A \neq L$. By Lemma 3.4, $\rho_R(A, L) = \rho_R(A, A \vee B) \leq \rho_R(A, B) = m_k$.

Since $\rho_R(A, L) > 0$, it follows that $\rho_R(A, L) = m_k$, contradicting the fact $L \notin \mathbf{V}_k$. Hence $(\mathbf{C}_k \setminus \mathbf{V}_k) \cap \mathbf{L}_k = \emptyset$. Based on the algorithm, $\mathbf{C}_{k+1} =$

$(\mathbf{C}_k \setminus \mathbf{V}_k) \cup \mathbf{L}_k$, so it follows from the above work that $c_{k+1} = c_k + l_k - v_k$.

(b) By assumption, $c_K = 1$, so by repeatedly applying part (a), we obtain

$$\begin{aligned} 1 &= c_K = c_{K-1} + l_{K-1} - v_{K-1} \\ &= c_{K-2} + (l_{K-2} - v_{K-2}) + (l_{K-1} - v_{K-1}) \\ &= \cdots = c_1 + \Sigma\{l_k - v_k \mid 1 \le k < K\} \end{aligned}$$

which establishes the result. □

If a context is strongly-well-formed, each singleton subset is an extent (by the discussion after Theorem 3.3). In this case, $\mathbf{C}_1 = \{\{x\} \mid x \in \mathcal{O}\}$, so $c_1 = |\mathcal{O}|$ and by Theorem 5.4(b), $\Sigma\{l_k - v_k \mid 1 \le k < K\} = 1 - |\mathcal{O}|$.

This equality can be rephrased in terms of average values of l_k and v_k on the first $K - 1$ iterations:

$$\text{average}(l) = \text{average}(v) + \frac{1 - |\mathcal{O}|}{K - 1}. \tag{17}$$

In other words, the average number of labels is always smaller than the average number of vertices. In fact, for the majority of examples that we have considered, on every iteration, the number of labels is less than the number of vertices. Based on this assumption, we can establish the following result.

Corollary 5.5 *Assume that the algorithm terminates on iteration K and $l_k < v_k$ for each $1 \le k < K$. Then $K \le |\mathcal{O}|$ and $|clusters| < |\mathcal{O}|^2$.*

Proof. By the hypothesis and Theorem 5.4(b), $1 - c_1 = \Sigma\{l_k - v_k \mid 1 \le k < K\} \le (-1)(K-1)$, so $K \le c_1 \le |\mathcal{O}|$. By the hypothesis and Theorem 5.4(a), $c_{k+1} = c_k + l_k - v_k < c_k$. Hence $c_k \le c_1$ for each $1 \le k < K$, so by the hypothesis and lines 1, 5, and 8 in the algorithm,

$$\begin{aligned} |clusters| &\le c1 + \Sigma\{l_k \mid 1 \le k < K\} \\ &< c_1 + \Sigma\{v_k \mid 1 \le k < K\} \\ &\le c_1 + \Sigma\{c_k \mid 1 \le k < K\} \le K c_1. \end{aligned}$$

By the first part of the proof, $K \le |\mathcal{O}|$; hence $|clusters| \le |\mathcal{O}|^2$. □

Our experience with both real-world data and randomly generated data indicates that the assumption $l_k < v_k$ holds on almost every iteration. This point is discussed further in Section 6.

The final two results in this section show that the hypothesis in Corollary 5.5 is satisfied if natural restrictions are placed on either the number of nearest pairs or the structure of the sets of clusters. Define the number of ties on iteration k as

$$ties_k = |\mathbf{E}_k| - 1.$$

In particular, if there is a unique nearest pair of sets at iteration k, then there are "no ties", i.e. $ties_k = 0$.

Corollary 5.6 *Assume that the algorithm terminates on iteration K.*

(a) *If $ties_k \le 1$ for each iteration k, then $K \le |\mathcal{O}|$ and $|clusters| < 3|\mathcal{O}|$.*

(b) *If* $ties_k = 0$ *for each iteration* k*, then* $|clusters| < 2|\mathcal{O}|$.

Proof. (a) By hypothesis, $e_k \leq 2$ for each k, so $e_k < v_k$. Since $l_k \leq e_k$, it follows that $l_k < v_k$, so by Corollary 5.5, $K \leq |\mathcal{O}|$. Therefore,

$$|clusters| \leq c_1 + \Sigma\{l_k \mid 1 \leq k < K\} \leq |\mathcal{O}| + 2(K-1) < 3|\mathcal{O}|.$$

(b) By hypothesis, $l_k = e_k = 1$, so by part (a),

$$|clusters| \leq c_1 + \Sigma\{l_k \mid 1 \leq k < K\} \leq |\mathcal{O}| + K - 1 < 2|\mathcal{O}|. \qquad \square$$

Corollary 5.7 *Assume that the algorithm terminates on iteration* K.

(a) *If* $\mathbf{C}_{k+1}$ *is a partition for some* $1 \leq k < K$*, then every edge in a component of the graph* $(\mathbf{V}_k, \mathbf{E}_k)$ *has the same label and* $l_k < v_k$.

(b) *If* $\mathbf{C}_k$ *is a partition for each iteration* k*, then* $K \leq |\mathcal{O}|$ *and* $|clusters| < 2|\mathcal{O}|$.

Proof. (a) Let E denote the set of edges in a component of the graph $(\mathbf{V}_k, \mathbf{E}_k)$. If there exists a pair of edges in E with distinct labels, then there exist adjacent edges (A, B) and (B, C) in E such that $L = A \vee B \neq B \vee C = L^*$. Therefore, $B \subseteq L \cap L^*$ and $\{L, L^*\} \subseteq \mathbf{C}_{k+1}$, which contradicts the fact that $\mathbf{C}_{k+1}$ is a partition. Hence every member of E has the same label.

Assume that $(\mathbf{V}_k, \mathbf{E}_k)$ has p_k components. By the above work, $l_k = p_k$. Since each component contains at least two vertices, $2p_k \leq v_k$, so $l_k < v_k$.

(b) The first estimate in part (b) follows from part (a) and Corollary 5.5. To establish the second estimate, by part (a), $l_k = p_k = 2p_k - p_k \leq v_k - l_k$, so by Theorem 5.4(b),

$$\begin{aligned} |clusters| &\leq c_1 + \Sigma\{l_k \mid 1 \leq k < K\} \\ &\leq c_1 + \Sigma\{v_k - l_k \mid 1 \leq k < K\} \\ &= 2c_1 - 1 < 2|\mathcal{O}|. \end{aligned} \qquad \square$$

6 Discussion

The two principal contributions of the paper are the definition of the distance functions ρ_R and δ_R and the formulation of the clustering algorithm. The distance functions provide a natural way to measure the distance between sets and objects that can be used to characterize extents (Theorem 3.5) and power-set lattices (Theorem 4.1). The clustering algorithm provides a method for computing a tractable number of potentially useful family of extents without excessive computational cost.

The algorithm differs from the classical clustering algorithms described in [10]. First, the values of the set function ρ_R, not δ_R, determine which pairs of sets are combined on each iteration. Second, the clusters defined on each iteration are generally anti-chains and may not be partitions. Third, the joins of all nearest pairs of clusters are included to ensure that no potentially interesting cluster is eliminated. Therefore, the analysis of the algorithm is

more involved than the analysis of classical algorithms. The difficulty with the analysis is inherent in using the join operation to guarantee extents. Even if no ties occur, the covers $\{\mathbf{C}_k\}$ still may not be partitions.

Currently, we do not have general results that provide reasonable bounds on the number of iterations of the algorithm. However, experiments on a variety of contexts confirm that the number of iterations is approximately equal to the number of objects. In fact, Example 8.2 shows that on trials with randomly generated data, the number of labels is less than the number of vertices for over 99% of the total number of iterations, and Example 8.1 shows that for the cancer data, the inequality holds on every iteration. Hence the linear behavior is guaranteed by Corollary 5.5.

Using the joins of all nearest pairs of clusters also increases the worst-case upper bound on the number of clusters generated. The reduced contexts that generate the $k-n$-subsets lattices (discussed before Theorem 5.4) show that, in general, there is no polynomial bound on the number of clusters as a function of the number of objects. However, this exponential blowup has never occurred in running the algorithm on any real-world or randomly generated contexts. In all cases, the number of clusters generated is approximately equal to twice the number of objects. This assertion is supported by the data in Example 8.1 and 8.2. Therefore, the addition of joins of all nearest pairs to the set of clusters enriches the set of overall clusters without significantly increasing the computational cost.

In terms of data analysis, the algorithm provides a useful technique for generating a manageable number of concepts to examine as opposed to computing the entire concept lattice. The approach is especially advantageous as the density of 1's approaches 50% in a context. In this case the concept lattice is usually very large; for example, randomly generated contexts of size 60×60 have, on the average, over $280,000$ concepts (Example 8.3). An additional advantage of the algorithm is that computational cost is not increased significantly by adding extra attributes. Therefore, we can freely use the complemented extension of a context to represent negative information. Also, if the context is derived from discretizing continuous data (as in Example 8.1), there is no significant cost penalty for improving the accuracy of the approximation by adding more attributes. In fact, Example 8.2 shows that as the number of attributes increases, the number of ties decreases, which provides further evidence that the number of iterations and the number of clusters will be linear functions of the number of objects.

For most of our work, we have focused on the join operator used in the algorithm. However, we have also studied the union version of the algorithm where ψ is defined by $\psi(C, C') = C \cup C'$. This version has the apparent disadvantage that it may generate clusters that are not extents. Also, the algorithm needs to handle the special case when the minimum distance m is zero (which can happen for distinct sets that are not extents). However, the union version is useful for assessing the degree to which the data is "well-

structured", that is, most of the clusters obtained are extents.

The clustering algorithm has been implemented in a Windows based system that includes a user interface with a variety of execution and display options, a database facility for storing and analyzing clusters, and a facility for conducting statistical trials on randomly generated contexts. The implementation of the algorithm can be further optimized, but in spite of this fact, relatively large contexts (on the order of 200×400) can be processed in a few minutes on a 200 MHz Pentium machine. In particular, the algorithm has been run on contexts generated from a variety of sources ranging from medical research to sporting events. Example 8.1 shows the results from using a set of cancer data. Overall, the implementation is a useful tool for computing a tractable subset of potentially very large concept lattices.

7 Related Work

Historically, cluster analysis is divided into two main categories: bottom-up and top-down. Our algorithm is bottom-up (also known as hierarchical) so it is related in spirit to classical clustering algorithms such as single-linkage and complete-linkage. There are several well-known books on cluster analysis ([1], [10]) that discuss these topics. More modern approaches to cluster analysis are found in ([6], [7]).

The literature on concept analysis can be divided into three areas: the general theory, efficient generation of concept lattices, and applications of concept analysis. Reference [8] presents the mathematical foundations of the subject and [4] also contains an introduction to closure operators, Galois connections, and concept analysis. The reference [13] describes an implementation of an algorithm to compute concept lattices and [9] describes incremental algorithms for generating concept lattices. There are numerous papers regarding the application of concept analysis, perhaps most notably relating to data mining ([3], [11]) and software engineering ([14], [15]).

To our knowledge this paper is the first to formally investigate the relationship between cluster and concept analyses. The paper [5] presents an informal discussion of the comparative advantages and disadvantages of each approach. Finally, we have not seen the distance function ρ_R discussed in the literature, but there are several indirect references to δ_R. For example, in ([1], p. 117), $1 - \delta_R$ is called the *Jaccard coefficient* and in ([2], p. 409) the same quantity is used. The latter reference states without proof that δ_R is a metric.

8 Examples

Example 8.1 We used a subset of the cancer data [16] consisting of 200 rows and 390 columns of 0's and 1's ($|\mathcal{O}| = 200$, $|\mathcal{A}| = 390$). Each row represents one tumor and various sets consisting of 39 columns represent the discretization of 10 distinct attributes of the tumor such as mass and diameter. The rows

represent equal percentages of benign and malignant tumors. Running the algorithm on this context required 191 iterations and generated 389 extents. The algorithm found one extent of size 90 that is 88% malignant, but it failed to find any cluster of significant size that was over 50% benign. Running the algorithm on the 200×780 complemented extension of the context required 151 iterations and generated 409 extents. The algorithm found one extent of size 81 that is 85% malignant and one extent of size 95 that is 88% benign. The following chart shows the distribution of ties for the complemented extension: on 86% of the iterations, there was no tie, and on 94% of the iterations there was at most one tie.

number of iterations	129	12	3	1	2	1	2
$ties_k$	0	1	2	3	4	6	13
% of iterations	86	8	2	0.67	1.34	0.67	1.34

Fig. 5.

Example 8.2 The following tables are based on running the algorithm on 100 randomly generated contexts where the probability of a 1 in a row-column entry is 0.5. In Tables II(a) and II(b), the algorithm was run on the complemented extension of randomly generated contexts with one-half the number of attributes listed for $|\mathcal{A}|$. For each row in Table I(a) and II(a), the average number of iterations on a given run is less than $|\mathcal{O}|$ and the average number of clusters per trial is approximately $2|\mathcal{O}|$. A comparison of rows 1 and 2, and 2 and 3, respectively, in Table I(a), and rows 2 and 3 in Table II(a), shows that as the number of attributes increases, the average number of ties/iteration on a given run significantly decreases. Table I(b) (Table II(b)) shows that for over 94% (84%) of the iterations on contexts of any size, the number of ties is at most 2. Furthermore, the two tables show that for over 99% of the iterations on contexts of any size, the number of labels is less than the number of vertices in the graph.

			iterations		clusters		ties/iteration	
I(a)	$\|\mathcal{O}\|$	$\|\mathcal{A}\|$	average	max	average	max	average	max
1	25	25	20.09	23	49.97	55	0.56	1.20
2	25	50	21.96	24	48.65	51	0.31	0.79
3	50	50	40.23	47	98.98	105	0.52	0.89
4	50	100	44.02	48	96.92	102	0.35	0.70
5	100	100	81.71	90	196.05	201	0.56	0.88

			% of iterations: ties				% of iterations: $l_k - v_k$	
I(b)	$\|\mathcal{O}\|$	$\|\mathcal{A}\|$	$= 0$	$= 1$	$= 2$	> 2	< 0	$= 0$	> 0
1	25	25	71.50	16.82	6.44	5.24	99.74	0.26	0.00
2	25	50	83.83	10.11	3.44	2.62	100.0	0.00	0.00
3	50	50	78.49	14.70	2.78	4.03	99.95	0.05	0.00
4	50	100	89.40	7.51	0.63	2.46	99.98	0.02	0.00
5	100	100	83.11	12.42	2.46	2.01	100.0	0.00	0.00

Table 1

			iterations		clusters		ties/iteration	
II(a)	$\|\mathcal{O}\|$	$\|\mathcal{A}\|$	average	max	average	max	average	max
1	25	50	18.85	22	53.39	64	0.82	1.69
2	50	50	31.08	37	109.63	128	1.34	2.24
3	50	100	33.76	40	105.16	118	0.99	1.76
4	100	100	57.78	67	215.03	230	1.54	2.09

			% of iterations: ties				% of iterations: $l_k - v_k$	
II(b)	$\|\mathcal{O}\|$	$\|\mathcal{A}\|$	$= 0$	$= 1$	$= 2$	> 2	< 0	$= 0$	> 0
1	25	50	66.27	13.62	8.85	11.26	99.27	0.62	0.11
2	50	50	64.66	14.53	4.89	15.92	99.60	0.30	0.10
3	50	100	71.28	11.05	4.97	12.70	99.69	0.31	0.00
4	100	100	71.12	12.48	4.21	12.19	99.84	0.16	0.00

Table 2

Example 8.3 The following table shows the average number of concepts generated by running the algorithm described in [13] on 50 randomly generated contexts where the probability of a 1 in a row-column entry is 0.5.

$\lvert\mathcal{O}\rvert$	$\lvert\mathcal{A}\rvert$	concepts	lg(concepts)		$\lvert\mathcal{O}\rvert$	$\lvert\mathcal{A}\rvert$	concepts	lg(concepts)
40	40	20390	14.32		80	80	2188215	21.06
60	60	284693	18.12		100	100	11077685	23.40

Table 3

References

[1] M. R. Anderberg, "Cluster Analysis for Applications," Volume 19, Probability and Mathematical Statistics, Academic Press, New York, 1973

[2] D. Beeferman and A. Berger, *Agglomerative clustering of a search engine query log,* Proc. KDD-2000, Boston, MA, August, 2000, pp. 407-415.

[3] R. Cole, P. Eklund, and D. Walker, *Constructing conceptual scales in formal concept analysis,* Pacific-Asia Conference on Knowledge Discovery and Data Mining, 1998, pp. 378-379.

[4] B. A. Davey and H.A. Priestley, "Introduction to Lattices and Order," Cambridge University Press, 1990.

[5] A. van Deursen and T. Kuipers, *Identifying objects using cluster and concept analysis,* Proceedings 21st International Conference on Software Engineering (ICSE'99), ACM, 1999, pp. 246-255.

[6] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay, *Clustering in large graphs and matrices,* ACM Symposium on Discrete Algorithms, 1999.

[7] C. Fraley and A. Raftery, *How many clusters? Which clustering method? Answers via model-based cluster analysis,* Computer Journal **41** (1998), pp. 578-588.

[8] B. Ganter and R. Wille, "Formal Concept Analysis - Mathematical Foundations," Springer-Verlag, New York, 1996.

[9] R. Godin, R. Missaoui, and H. Alaoui, *Incremental concept formation algorithms based on Galois (concept) lattices,* Computational Intelligence **11** (1995), pp. 246-267.

[10] J. Hartigan, "Cluster Analysis," Wiley Series in Probability and Mathematical Statistics, New York, 1975.

[11] K. Hu, Y. Lu, and C. Shi, *Incrementally discovering association rules: a concept lattice approach,* Pacific-Aisa Conferenece on Knowledge Discovery and Data Mining, 1999, pp. 109-113.

[12] J. L. Kelley, "General Topology," Van Nostrand, Princeton, 1955.

[13] C. Lindig, *Fast concept analysis,* 8th International Conference on Conceptual Structures (ICCS 2000), Darmstadt, Germany, August, 2000 `http://www.eecs.harvard.edu/ lindig/papers/fca/index.html`

[14] M. Siff and T. Reps, *Identifying modules via concept analysis,* IEEE Transactions on Software Engineering **25** (1999), pp. 749-768.

[15] G. Snelting and F. Tip, *Reengineering class hierarchies using concept analysis,* Proc. SIGSOFT Symposium on Foundations of Software Engineering, ACM, 1998.

[16] Wisconsin breast cancer data `ftp://ftp.cs.wisc.edu/math-prog/cpo-dataset/machine-learn/cancer1/`.