Clustering and Identification of Core Implications

Domingo López-Rodríguez(✉), Pablo Cordero, Manuel Enciso, and Ángel Mora

Universidad de Málaga, Málaga, Spain
{dominlopez,pcordero,enciso,amora}@uma.es

Abstract. FCA exhaustively uses the notion of cluster by grouping attributes and objects and providing a solid algebraic structure to them through the concept lattice. Our proposal explores how we can cluster implications. This work opens a research line to study the knowledge inside the clusters computed from the Duquenne-Guigues basis. Some alternative measures to induce the clusters are analysed, taking into account the information that directly appears in the *appearance* and the *semantics* of the implications. This work also allows us to show the `fcaR` package, which has the main methods of FCA and the Simplification Logic. The paper ends with a motivation of the potential applications of performing clustering on the implications.

1 Introduction

Formal Concept Analysis (FCA) has established itself at the theoretical level and is increasingly used in real-life problems [6,7,33]. Our community explores how to solve real problems in data science, machine learning, social network analysis, etc. Solving problems from these areas and developing new tools could be a way to open a window to researchers outside FCA.

Since the early eighties, when R. Wille and B. Ganter [16] developed Formal Concept Analysis, the community has been growing. The interest in the use of this well-founded tool has increased considerably. The continuous development of the theoretical foundations and generalisations of the classical framework [3,4,13,24,28,30] and the enthusiasm of how to put in practice this progress [6–8,18,33] have formed a solid community formally linked. However, as U. Priss mentioned in [32], "FCA is widely unknown among information scientists in the USA even though this technology has a significant potential for applications". The community recognises that it is necessary an additional effort and perhaps new tools to make FCA more appealing. Books about machine learning, big data, and data science, in general, have not included anything about FCA, notwithstanding its powerful knowledge and its considerable potential in applications.

Supported by Grants TIN2017-89023-P, UMA2018-FEDERJA-001 and PGC2018-095869-B-I00 of the Junta de Andalucia, and European Social Fund.

© Springer Nature Switzerland AG 2021
A. Braud et al. (Eds.): ICFCA 2021, LNAI 12733, pp. 138–154, 2021.
https://doi.org/10.1007/978-3-030-77867-5_9

Developing new tools and libraries could be a valuable resource to open our community. In this work, we present a library in the R language, named `fcaR`, which implements of the most popular methods and algorithms in FCA. Here we show how to extract interesting knowledge by computing implication clusters from the Duquenne-Guigues basis to illustrate the benefits of such a tool. `fcaR` vertebrates this proposal by introducing some code throughout the paper.

FCA is firmly based on the (bi)clustering of attributes and objects. The Concept Lattice provides a formal structure of these clusters. Nevertheless, the application of clustering to the set of implications has not been explored as far as we know. In the following section, we briefly survey the relationship between Clustering and FCA.

The first step in this paper is to propose several dissimilarity measures between implications to cluster them in different ways. In some of these measures, to compute the distance matrix between implications, we will use Simplification Logic and its attribute closure operator, included in the package `fcaR`. From this distance matrix, clusters of implications arise representing new knowledge. A K-medoid algorithm, specifically the PAM algorithm [23], is used to compute each cluster's central implications and further generate the clusters of implications.

Since our starting point is the Duquenne-Guigues basis, we analyse the different dissimilarity measures taking into account the pseudointents, the right-hand side of the implications, and the closed sets computed from the pseudointents. We end the paper with an experiment result and drawing up some potential applications from the implication clustering.

The rest of the paper is organised as follows: in Sect. 2, we analyse how clustering is used in FCA in the literature. The central notions of FCA and the `fcaR` package are briefly outlined in Sect. 3. Section 4 shows the new research line proposed in this work, along with its formulations and possible developments, also defining the idea of implication dissimilarity in terms of distance functions and how implication clustering is related to cluster pseudointents and their closures. This proposal's promising result is shown in Sect. 5, developing an experiment centred on a dataset well-known in the machine learning community. Finally, Sect. 6 presents some conclusions and future works.

2 Previous Works on FCA and Clustering

FCA carries out clustering of objects by itself. However, it is well-known that the size of the concept lattice is possibly exponential with respect to the size of the formal context, even for a small context. Diatta in [14] ensured that pattern concepts [17] coincide with clusters associated with dissimilarity measures.

Beyond that, techniques based on clustering have been explored to group the closest concepts. For instance, in [29], Melo *et al.* presented a tool to apply visual analytics to cluster concepts using a K-means algorithm [27] to identify clusters. Bocharov *et al.* [5] group the objects by the K-means algorithm and propose modifying the Close-by-One algorithm for consensus clustering to reduce the concept lattice. In [38], the authors compute attribute clusters using similarity and dissimilarity functions to reduce the concept lattice.

Stumme *et al.* [37] proposed Iceberg lattices as a clustering method of the original lattice to reduce computing bases of association rules and their visualisation. Kumar in [26] uses clustering to reduce the formal context and, therefore, the number of association rules extracted from it.

Other authors have used FCA in a variety of approaches to detect objects with similar properties. In [22], triclustering, based on FCA, was developed to detect groups of objects with similar properties under similar conditions and used it to develop recommender systems. Cigarrán *et al.* have some interesting works [7,9] applying FCA in Social Network Analysis to detect topics in Twitter. These authors proved that FCA could solve real problems with better results than classical techniques, generating clusters of topics less subject to cluster granularity changes.

Other works deal with the idea of clustering association rules (in transactional databases) to reduce the number of rules extracted [2,19,36]. These works do not define rule dissimilarity as a function of each rule's terms (items). Instead, they define the dissimilarity between rules in terms of the sets of transactions supporting each rule. Thus, the knowledge present in the rule clustering is explicitly related to the database and cannot be abstracted from it.

3 Background and the `fcaR` package

Over the years, U. Priss has collected a list of the main FCA-based tools on its website https://www.upriss.org.uk/fca/fca.html. We emphasise that the most used for FCA are ConExp, ToscanaJ, Galicia, FcaStone, and some libraries developed in C, Python, etc. In this work, we take the opportunity to present the `fcaR` package[1] as a valuable tool to solve real problems and bring FCA closer to other communities.

In the following, we briefly summarise the main concepts in Formal Concept Analysis (FCA) we need for this work, showing with a running example how the `fcaR` package is used. For more detailed reading about FCA, see [18].

Definition 1 (Formal Context). *A formal context is a triplet* $\mathbb{K} := \langle G, M, I\rangle$ *where* G *and* M *are non-empty finite sets and* $I \subseteq G \times M$ *is a binary relation between* G *and* M*.*

The elements in G and M are named objects and attributes, respectively. In addition, $(g, m) \in I$ is read as the object g has the attribute m.

Example 1. We consider this example appearing in [18] where G is the set of planets and M the set of some properties of these planets (Table 1).

In the R language, we will use the following to introduce this matrix with the name `planets` in the `fc_planets` formal context object. The sets G and

[1] As far as we know, no package using the R language has been developed and published in CRAN repository for FCA, even when the R language together with Python are considered the main languages in data science, machine learning, big data, etc. To this date, `fcaR` has more than 8,000 downloads.

Table 1. Properties of the planets of the solar system.

	Small	Medium	Large	Near	Far	Moon	No_moon
Mercury	×			×			×
Venus	×			×			×
Earth	×			×		×	
Mars	×			×		×	
Jupiter			×		×	×	
Saturn			×		×	×	
Uranus		×			×	×	
Neptune		×			×	×	
Pluto	×				×	×	

M, and also subsequently computed concepts and implications, are stored inside this formal context object[2].

```
> library(fcaR)
> fc_planets <- FormalContext$new(planets)
> fc_planets$attributes
[1] "small"  "medium"  "large"   "near"   "far"     "moon"    "no_moon"
> fc_planets$objects
[1] "Mercury" "Venus" "Earth" "Mars" "Jupiter" "Saturn" "Uranus"
"Neptune" "Pluto"
```

Each formal context $\mathbb{K}$ defines two derivation operators, which form a Galois connection between $\langle 2^G, \subseteq\rangle$ and $\langle 2^M, \subseteq\rangle$. They are the following:

$$(-)'\colon 2^G \to 2^M \text{ where } A' = \{m \in M \mid (g,m) \in I \text{ for all } g \in A\}.$$
$$(-)'\colon 2^M \to 2^G \text{ where } B' = \{g \in G \mid (g,m) \in I \text{ for all } m \in B\}.$$

Example 2. In `fcaR` we use sparse matrices and sparse sets to provide efficiency to the algorithms, then to use an object variable or an attribute variable, first we create a new sparse variable (`SparseSet$new` method), and then we assign the value 1 (`variable$assign` method). To compute `intent` and `extent` in R language, with the planets example, we will do the following:

```
> # The planets  are stored in a vector
> myPlanets <- c("Earth","Mars")
> # A new sparse object variable is created
> mySparsePlanets <- SparseSet$new(attributes = fc_planets$objects)
> # Assigning to myPlanets the value 1 in the variable sparse
> mySparsePlanets$assign(myPlanets,values = 1)
> # The content of the sparse variable is
> mySparsePlanets
```

[2] In this work, we do not use all the methods in the `fcaR` package to manage the formal context, the concept lattice, the concepts, the implications, etc. See https://neuroimaginador.github.io/fcaR/ for more details.

```
{Earth, Mars}
> fc_planets$intent(mySparsePlanets) # Computing the intent
{small, near, moon}

# In a similar way for attributes
> myAttributes <- c("medium", "far","moon")
> mySparseAttributes <- SparseSet$new(attributes = fc_planets$attributes)
> mySparseAttributes$assign(myAttributes,values = 1)
> mySparseAttributes
{medium, far, moon}
> fc_planets$extent(mySparseAttributes)
{Uranus, Neptune}
```

The main aim of this area is to extract knowledge from the context allowing to reason. One of the ways to represent knowledge is utilising the concept lattice. Another equivalent alternative knowledge representation, more suitable to define reasoning methods, is given in terms of attribute implications.

Definition 2 (Attribute Implication). *Given a formal context* $\mathbb{K}$*, an attribute implication is an expression* $A \rightarrow B$ *where* $A, B \subseteq M$ *and we say that* $A \rightarrow B$ *holds in* $\mathbb{K}$ *whenever* $B' \subseteq A'$*.*

That is, $A \rightarrow B$ holds in $\mathbb{K}$ if every object that has all the attributes in A also has all the attributes in B. The closeness of these expressions with propositional logic formulas leads to a logical style way to manage them. Although the most used syntactic inference system is the so-called Armstrong's Axioms, we will use the Simplification Logic, $\mathbb{SL}$, introduced in [10]. This logic allows the design of automated reasoning methods [10–12,31] and it is guided by the idea of simplifying the set of implications by efficiently removing redundant attributes. In [31], the results and proofs about $\mathbb{SL}$ are presented.

Example 3. We use the `fcaR` package to extract the set of implications from the formal context in Example 1, by using the Next_Closure algorithm [16], using the command `fc_planets$find_implications()`. The set of implications is

$$
\begin{aligned}
\Gamma = \{\ &\{\text{no\_moon}\} && \Rightarrow \{\text{small, near}\}\\
&\{\text{far}\} && \Rightarrow \{\text{moon}\}\\
&\{\text{near}\} && \Rightarrow \{\text{small}\}\\
&\{\text{large}\} && \Rightarrow \{\text{far, moon}\}\\
&\{\text{medium}\} && \Rightarrow \{\text{far, moon}\}\\
&\{\text{medium, large, far, moon}\} && \Rightarrow \{\text{small, near, no\_moon}\}\\
&\{\text{small, near, moon, no\_moon}\} && \Rightarrow \{\text{medium, large, far}\}\\
&\{\text{small, near, far, moon}\} && \Rightarrow \{\text{medium, large, no\_moon}\}\\
&\{\text{small, large, far, moon}\} && \Rightarrow \{\text{medium, near, no\_moon}\}\\
&\{\text{small, medium, far, moon}\} && \Rightarrow \{\text{large, near, no\_moon}\}\ \}
\end{aligned}
$$

An interesting argument of the `find_implications()` function, when the number of implications is large, is `parallelize` to take advantage of the cores in the machine. The functions `size, cardinality` can be applied to the `imps` variable to check the number of implications and the size of the attributes on them. The package eases the manipulation of implications using the typical operations of subsetting in R language (`imp[2:3]`, for instance).

To conclude this section, we introduce the outstanding notion of closure of a set of attributes with respect to a set of implications, which is strongly related to the syntactic treatment of implications. Note that the algorithms developed in `fcaR` package to manipulate implications and to compute closures are based on Simplification Logic [31]. For a set of implications, `apply_rules` and `closure` functions can be respectively applied to remove redundancy and to compute the closures of attributes[3]. We make clear that the results in this paper are independent of the closure algorithm used.

Definition 3. *Given $\Gamma \subseteq \mathcal{L}_M$ and $X \subseteq M$, the (syntactic) closure of X with respect to Γ is the largest subset of M, denoted X^+_Γ, such that $\Gamma \vdash X \to X^+_\Gamma$.*

The mapping $(-)^+_\Gamma \colon 2^M \to 2^M$ is a closure operator on $\langle 2^M, \subseteq \rangle$. This notion is the key to designing automatic reasoning methods due to the following equivalence:

$$\Gamma \vdash A \to B \quad \text{iff} \quad \{\varnothing \to A\} \cup \Gamma \vdash \varnothing \to B \quad \text{iff} \quad B \subseteq A^+_\Gamma$$

From now on, we omit the subindex (i.e. we write X^+) when no confusion arises.

Example 4. We will use the following to compute the closure of the attribute named `small` in Example 1 using our `fcaR` package:

```
> S <- SparseSet$new(attributes = fc_planets$attributes)
> S$assign("small"=1)
> imps$closure(S)
{small, far, moon}
```

4 Proposed Research Line

In this section, we propose a new research line accompanied by preliminary results. In this line, we aim to study the potential use and applications of performing (unsupervised) clustering on the Duquenne-Guigues basis of implications. We present this idea using a running example, and we have used the `fcaR` package to help automate the computations and perform experiments.

Given a formal context $\mathbb{K} = (G, M, I)$, and a set of valid implications Γ, we can interpret Γ as a partition (disjoint by definition), i.e., $\Gamma = \Gamma_1 \cup \Gamma_2 \cup \ldots \cup \Gamma_K$, where each set Γ_i is called a *cluster of implications*, and it is defined such that

$$\phi(\Gamma_1, \ldots, \Gamma_K) = \sum_{i=1}^{K} \delta(\Gamma_i)$$

[3] See https://neuroimaginador.github.io/fcaR/articles/implications.html.

is minimum, where $\delta(\Gamma_i)$ represents an internal dissimilarity measure in Γ_i. Thus, our motivation is to group similar implications in the same cluster, building homogeneous groups of implications.

In a similar way to classical clustering techniques, $\delta(\Gamma_i)$ can be defined in terms of the distances between implications in the same cluster Γ_i. Therefore, we propose defining a distance function between implications that can adequately capture and differentiate the essential aspects of their appearance and semantics. Thus, given two implications $P \to Q$ and $R \to T$ from the Duquenne-Guigues basis, we propose to quantify their different appearance by measuring the dissimilarity between P and R and/or between Q and T. Their possibly different semantic information can be quantified by comparing their syntactic attribute closures P^+ and R^+. Our intuition is that the pseudo-intents and the closed sets play an essential role in the clusters, but we want to explore the possibilities.

In order to measure the (dis)similarity between two sets of attributes, we can consider several options. Let us suppose $A, B \subset M$. The following measures are based on well-known distances:

- Hamming (or Manhattan) distance [20]: $d_{\mathrm{M}}(A, B) = |A \triangle B|$ (where $\triangle$ denotes the symmetric set difference operator) measures the amount of attributes that are present in only one of A and B.
- Jaccard index [21]: $d_{\mathrm{J}}(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|}$ measures the proportion of common attributes in A and B.
- Cosine distance: $d_{\cos}(A, B) = 1 - \frac{|A \cap B|}{\sqrt{|A| \cdot |B|}}$.

Thus, the dissimilarity $\mathrm{dis}(P \to Q, R \to T)$ between two implications $P \to Q$ and $R \to T$, following the previous comment, can be defined in terms of $d(P, R)$, $d(Q, T)$ and $d(P^+, R^+)$, where d is any of d_{M}, d_{J} or $d_{\cos}$. The use of one or another of these terms is subject to the interpretation and could partially depend on the problem to solve.

Initially, we aim at studying these different possibilities:

$$
\begin{aligned}
\mathrm{dis}_1(P \to Q, R \to T) &:= d(P, R)\\
\mathrm{dis}_2(P \to Q, R \to T) &:= d(P^+, R^+)\\
\mathrm{dis}_3(P \to Q, R \to T) &:= d(P, R) + d(Q, T)\\
\mathrm{dis}_4(P \to Q, R \to T) &:= d(P, R) + d(P^+, R^+)\\
\mathrm{dis}_5(P \to Q, R \to T) &:= d(P, R) + d(Q, T) + d(P^+, R^+)
\end{aligned}
$$

Remember that $d(P, R)$ is a term that quantifies the difference between the pseudointents forming the left-hand sides of the corresponding implications and that $d(P^+, R^+)$ measures the difference in the closed sets that are produced by using the implications. Pseudointents and closed sets represent two levels in the biclustering of the formal context. Therefore it is reasonable to think about clustering the implications by using those distinctive components.

Once all the pairwise distances are computed, we can use a clustering algorithm to generate the implication clusters. For each cluster, we determine an implication, named *central implication*, $P \to Q$, providing the measure of internal dissimilarity in the cluster as follows:

$$\delta(\Gamma_i) := \frac{1}{|\Gamma_i|} \sum_{R \to T \in \Gamma_i} \operatorname{dis}(P \to Q, R \to T)$$

which can be interpreted as a measure of within Γ_i dispersion.

The clustering algorithm should provide a proper partition of Γ such that $\phi(\Gamma_1, \ldots, \Gamma_K)$ is minimum. Another characteristic of clustering is that each implication $R \to T$ is assigned to a cluster Γ_i if, by definition, its dissimilarity to the central implication of Γ_i (which we will call $P_i \to Q_i$) is lower than its dissimilarity to the central implications of the other clusters, that is, if

$$\operatorname{dis}(P_i \to Q_i, R \to T) \leq \operatorname{dis}(P_j \to Q_j, R \to T) \quad \forall j \neq i$$

Given the definition of dissimilarity above, the proposed clustering aims at building coherent groups of implications that have similar pseudointents or produce similar closed sets.

There are many possible choices of clustering algorithms. In this paper, we propose the use of the PAM (partitioning around medoids) algorithm [23] to compute the clusters and their central implications, which, in this context are called the *medoids* of the clusters, since it is more robust to the presence of noise and isolated components in the data than the K-means algorithm [27], widely used in machine learning.

Example 5. Following our running example, we will find clusters in the implications of Example 3. And, for instance, we consider the dissimilarity function

$$\operatorname{dis}(P \to Q, R \to T) := |P \triangle R| + |P^+ \triangle R^+| \tag{1}$$

The next R code computes the dissimilarity matrix, that is, the matrix $D = (D_{i,j})$ where the entry $D_{i,j} := \operatorname{diss}(P_i \to Q_i, P_j \to Q_j)$ is the dissimilarity between the ith and the jth implications.

```
> diss <- implication_distance(imps)
> D <- as.matrix(diss)
> D
   1  2  3 4 5  6  7 8  9 10
1  0  7  3 8 8  9  7 9  9  9
2  7  0  6 3 3  8 10 8  8  8
3  3  6  0 7 7 10  8 8 10 10
4  8  3  7 0 4  7  9 9  7  9
5  8  3  7 4 0  7  9 9  9  7
6  9  8 10 7 7  0  6 4  2  2
7  7 10  8 9 9  6  0 2  4  4
8  9  8  8 9 9  4  2 0  2  2
9  9  8 10 7 9  2  4 2  0  2
10 9  8 10 9 7  2  4 2  2  0
```

Then, we use the PAM algorithm of the `cluster` R package to compute the clusters using $K = 2$ (two clusters) and their central implications.

```
> cluster <- cluster::pam(diss, k = 2)
> # The following are the central implications
> imps[cluster$id.med]
Implication set with 2 implications.
Rule 1: {far} -> {moon}
Rule 2: {small, near, far, moon} -> {medium, large, no_moon}
```

Therefore, we already have the implications in each cluster:

```
> imps[cluster$clustering == 1]
Implication set with 5 implications.
Rule 1: {no_moon} -> {small, near}
Rule 2: {far} -> {moon}
Rule 3: {near} -> {small}
Rule 4: {large} -> {far, moon}
Rule 5: {medium} -> {far, moon}
> imps[cluster$clustering == 2]
Implication set with 5 implications.
Rule 1: {medium, large, far, moon} -> {small, near, no_moon}
Rule 2: {small, near, moon, no_moon} -> {medium, large, far}
Rule 3: {small, near, far, moon} -> {medium, large, no_moon}
Rule 4: {small, large, far, moon} -> {medium, near, no_moon}
Rule 5: {small, medium, far, moon} -> {large, near, no_moon}
```

Note that the second cluster is formed by implications that present all the attributes. This cluster can be disregarded as uninformative since its implications present combinations of attributes that are not found in any object of the formal context. In terms of association rules, they would be considered as implications with zero-support and not interesting for our proposal. Thus, in what follows, we will consider only implications that do not present all attributes.

```
> # Take the first 5 implications
> imps <- imps[1:5]
> diss <- implication_dist(imps)
> D <- as.matrix(diss)
> rownames(D) <- seq(imps$cardinality())
> D
   1  2  3  4  5
1  0 10  4 12 12
2 10  0  8  4  4
3  4  8  0 10 10
4 12  4 10  0  4
5 12  4 10  4  0
```

Furthermore, for these implications, the computation of the clusters produces the following clusters:

```
> cluster <- cluster::pam(diss, k = 2)
> # The central implications
> imps[cluster$id.med]
```

```
Implication set with 2 implications.
Rule 1: {near} -> {small}
Rule 2: {far} -> {moon}
> # The cluster 1 is:
> imps[cluster$clustering == 1]
Implication set with 2 implications.
Rule 1: {no_moon} -> {small, near}
Rule 2: {near} -> {small}
> # The cluster 2 is:
> imps[cluster$clustering == 2]
Implication set with 3 implications.
Rule 1: {far} -> {moon}
Rule 2: {large} -> {far, moon}
Rule 3: {medium} -> {far, moon}
```

The computed clustering can be viewed as the result of minimising the clustering algorithm's objective function. It will be minimum the mean dissimilarity of each implication in its cluster with respect to its central implication.

As a conclusion, we can observe that the clustering renders a natural result describing the clusters as a set of implications with specific knowledge about:

1. Planets near the Sun which are therefore small.
2. Distant planets that, therefore, have satellites.

Now, we approach to experiment what happens when we change the dissimilarity function to:

$$\mathrm{dis}(P \to Q, R \to T) := |P \triangle R| \tag{2}$$

that is, when we consider only the *difference* in the pseudointents. Similarly, we compute the new dissimilarity matrix and obtain:

```
> D_LHS
  1 2 3 4 5
1 0 2 2 2 2
2 2 0 2 2 2
3 2 2 0 2 2
4 2 2 2 0 2
5 2 2 2 2 0
```

We can observe that any two implications are at distance 2. The consequence is that the clusters will be uninformative. Any possible partition into two clusters Γ_1 and Γ_2, using this dissimilarity matrix, has the same mean dissimilarity; therefore, clusters can be considered as generated by randomness. We can check the central implications:

```
> clusterLHS <- cluster::pam(dissLHS, k = 2)
> # The central implications
> imps[clusterLHS$id.med]
Implication set with 2 implications.
Rule 1: {large} -> {far, moon}
Rule 2: {medium} -> {far, moon}
```

Note the overlap in the closed sets defined by these central implications. In this case, the implications in the first cluster and the unique implication in the second cluster (that is also its central implication) does not provide any further insight, making it clear that, in this case, the clusters are random guesses.

```
# First cluster - Implication set with 4 implications.
Rule 1: {no_moon} -> {small, near}
Rule 2: {far} -> {moon}
Rule 3: {near} -> {small}
Rule 4: {large} -> {far, moon}
# Second cluster - Implication set with 1 implications.
Rule 1: {medium} -> {far, moon}
```

It seems clear that to consider the dissimilarity measure proposed in Eq. (1), representing the difference in the knowledge provided by pseudo-intents and closed sets, is more appropriate than the proposed in Eq. (2), representing only the differences in the pseudointents.

To conclude this section, we explain the line of research we have in mind. The clustering relationship on implications to object and attribute clustering or to concept clustering seems to be interesting. We devise potential future applications in reducing the computational cost of computing closures in specific scenarios or the possible application to FCA's factorisation techniques. Also, it will be of interest to study the different properties of implication clustering when performed on different types of bases (direct-optimal [34] and ordered-direct bases [1] and sets of implications without attribute redundancies, for instance). Last, it will be of interest to extend the study to determine the properties of clustering of association rules with this new proposal, in contrast to what has already been studied [2,19,36].

5 Experimental Results

This section presents results to illustrate how the obtained clustering of implications is consistent with the formal context's observed data.

We apply our proposal to the data from the so-called MONK's problems [15], a well-known set of 3 datasets used in machine learning competitions. Each of the 3 datasets consists of 6 categorical attributes, `a1` to `a6`, taking integer values, and a binary `class` attribute. For this work, all categorical variables have been binarized, making an aggregate of 19 binary attributes, including the two class attributes, `class = 0` and `class = 1`.

For each of these three problems, we have computed the Duquenne-Guigues basis, consisting of 524, 723 and 489 implications, respectively. After removing the implications that incorporate all the attributes, as commented before, the final sets of implications consisted of 505, 704 and 471 implications for problems MONKS-1, MONKS-2 and MONKS-3, respectively.

Then, we apply a dissimilarity function (one of $\mathrm{dis}_1, \ldots, \mathrm{dis}_5$, or any other combination) to obtain a dissimilarity matrix. To determine the optimal number

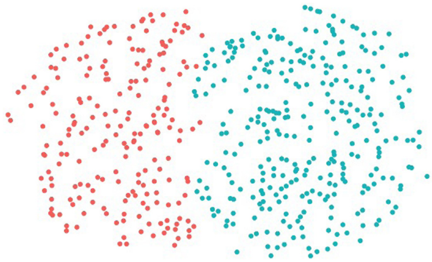

Fig. 1. Bi-dimensional representation of the implication space. Each dot represents an implication. In order to obtain this plot, the multidimensional scaling technique has been used to map implications to $\mathbb{R}^2$ points, preserving their mutual dissimilarities.

of clusters present in the implication set, we use the *silhouette index* [35]. In all these problems, the optimal number of clusters determined by this index was 2.

Thus, to continue with our proposal, for each problem we have applied the method to partition the implication set into two clusters. For instance, if we use the dissimilarity function dis_4, which incorporated the distance between the pseudointents and between the closures, with the Hamming distance, on the MONKS-1 problem, we obtain the following central implications:

$$\{\text{a5} = 1\} \Rightarrow \{\text{class} = 1\}$$
$$\{\text{class} = 0, \text{ a2} = 1, \text{ a5} = 2\} \Rightarrow \{\text{a6} = 2\}$$

The clustering results can be visually inspected by applying an algorithm of multidimensional scaling [25], whose results can be plotted to obtain a graphical bi-dimensional representation of the *implications space*. This plot can also be used to inspect the potential number of clusters present in the implications. The results of the clustering can be checked in Fig. 1.

In that Fig. 1, we can check that 2 clusters seem to be a good proposal. The clustering algorithm has almost correctly identified the two implication groups, confirming the estimated value using the *silhouette index*.

We have explored the consistency of the clustering performed using the different measures of dissimilarity. proposed in Sect. 4 ($\text{dis}_1, \ldots, \text{dis}_5$), based on different distance functions (Hamming, Jaccard and Cosine indexes).

First, we study if the implications leading to the same closed set are grouped in the same cluster. Thus, we introduce the notion of *closure purity*. Let us consider the set of *equivalence classes* in the Duquenne-Guigues basis Γ, as

$$[P \to Q] = \{R \to T \in \Gamma : P^+ = R^+\}$$

Two implications belong to the same equivalence class if the closures of their respective pseudointents are the same. Then, we can define *closure purity* as the proportion of those equivalence classes whose implications are all assigned to the same cluster. The ideal situation is that this index equals 1, meaning that whole equivalence classes form clusters.

Table 2. *Closure purity* for different dissimilarity measures and distance functions.

Problem	Dissimilarity	Hamming	Jaccard	Cosine
MONKS-1	dis_1	0.953	1.000	0.983
	dis_2	1.000	1.000	1.000
	dis_3	0.962	0.953	0.953
	dis_4	1.000	1.000	0.971
	dis_5	1.000	0.988	0.962
MONKS-2	dis_1	0.928	0.966	0.942
	dis_2	1.000	1.000	1.000
	dis_3	0.986	0.966	0.974
	dis_4	0.994	1.000	0.998
	dis_5	0.996	0.954	0.974
MONKS-3	dis_1	0.923	1.000	0.972
	dis_2	1.000	1.000	1.000
	dis_3	0.935	0.985	0.978
	dis_4	1.000	0.997	0.994
	dis_5	1.000	0.994	0.966

The results of this comparison are presented in Table 2. It is evident that dis_2 achieves closure purity equal to 1 since it is defined as the dissimilarity between the closures given by two implications. Thus, any two implications in the same equivalence class have dissimilarity 0 and therefore are assigned to the same cluster. Interestingly, other dissimilarity measures, such as dis_4, taking into account also the difference in the pseudointents, in many occasions achieve closure purity equal to 1, meaning that they can also separate the equivalence classes coherently.

Also, we study if there are common attributes inside the implications in a given cluster. Table 3 shows the attributes that appear in at least 80% of the implications in each cluster. Note that there is always a cluster with no common attributes using both Jaccard and Cosine indexes, indicating greater heterogeneity in that cluster's implications. With the Hamming distance, we obtain that the dissimilarity measures dis_2 (considering only P^+ and R^+) and dis_4 (considering, besides, the difference between pseudointents) always find common attributes in each cluster. Remarkably, the common attributes found are the class attributes mentioned earlier. Hence, the clustering procedure has been able to locate key attributes in a completely unsupervised manner, provided the knowledge present in the implication set. It is evident that if we reduce the threshold to be less than 80%, we will find a greater number of common attributes. We have used a threshold of 80% to retain just representative attributes in each cluster.

Table 3. Sets of common attributes in each of the clusters found for different distance functions and dissimilarity measures. A ∅ symbol indicates that no common attributes are found in the implications of the given cluster.

Problem	Diss.	Hamming		Jaccard		Cosine	
		Cluster 1	Cluster 2	Cluster 1	Cluster 2	Cluster 1	Cluster 2
MONKS-1	dis_1	{class = 1}	∅	{class = 1, a5 = 1}	∅	{class = 1, a5 = 1}	∅
	dis_2	{class = 1}	{class = 0}	{class = 1, a5 = 1}	∅	{class = 1, a5 = 1}	∅
	dis_3	{class = 1}	∅	{class = 1}	∅	{class = 1}	∅
	dis_4	{class = 1}	{class = 0}	{class = 1, a5 = 1}	∅	{class = 1}	∅
	dis_5	{class = 1}	{class = 0}	{class = 1}	∅	{class = 1}	∅
MONKS-2	dis_1	∅	∅	{a5 = 1}	∅	{a4 = 1}	∅
	dis_2	{class = 0}	{class = 1}	{class = 0, a6 = 1}	∅	{class = 0, a5 = 1, a6 = 1}	∅
	dis_3	{class = 0}	∅	{class = 0}	∅	{class = 0}	∅
	dis_4	{class = 0}	{class = 1}	{class = 0, a4 = 1, a6 = 1}	∅	{class = 0}	∅
	dis_5	{class = 0}	∅	{class = 0}	∅	{class = 0}	∅
MONKS-3	dis_1	∅	{class = 1}	{class = 0, a5 = 4}	∅	{class = 0, a5 = 4}	∅
	dis_2	{class = 0}	{class = 1}	{class = 0, a5 = 4}	∅	{class = 0, a5 = 4}	∅
	dis_3	{class = 0}	∅	{class = 0}	∅	{class = 0}	∅
	dis_4	{class = 0}	{class = 1}	{class = 0}	∅	{class = 0}	∅
	dis_5	{class = 0}	{class = 1}	{class = 0}	∅	{class = 0}	∅

This leads us to think that implication clustering could be used with promising results in classification tasks in datasets, and/or, for instance, in recommender systems for medical diagnosis in which some attributes play the role of identifying diseases from symptoms (the rest of the attributes). To finish this section, it seems clear that very compelling results are emerging from the hidden knowledge in the clusters of implications.

6 Conclusions

We have presented the `fcaR` package developed in the R language throughout this work. The package has two objectives. The first one is to provide a tool to the FCA community and make FCA works visible to other areas as machine learning, data science, etc., where the use of the R language is widely extended. Thus, in this line, to promote the so-called reproducible research and the sharing of knowledge, the scripts to replicate the results in this work, as well as the results themselves, are hosted in https://github.com/Malaga-FCA-group/FCA-ImplicationClustering.

From the theoretical point of view, the paper proposes a method to cluster implications, hence extracting interesting knowledge about the central implications, which reveal groups of objects with a special meaning and shared characteristics. This work opens the windows to new interesting research in current areas of interest as Social Network Analysis. The identification of topics could be addressed by our clustering implication method based on logic.

Natural clusters (consistent with the data) seem to emerge from the implication clusters, and this could have potential applications to reduce the concept lattice, the bases of implications, etc. Key attributes arise from the clusters, with potential applications revealing attributes and object clusters and their leaders. It could also be of interest to study the relationship between the concept lattice obtained directly from a formal context and obtained after clustering objects. The study of *closure purity* can reveal interesting properties about closed sets and their features.

References

1. Adaricheva, K., Nation, J., Rand, R.: Ordered direct implicational basis of a finite closure system. Discrete Appl. Math. **161**(6), 707–723 (2013)
2. An, A., Khan, S., Huang, X.: Hierarchical grouping of association rules and its application to a real-world domain. Int. J. Syst. Sci. **37**(13), 867–878 (2006)
3. Belohlavek, R., De Baets, B., Outrata, J., Vychodil, V.: Computing the lattice of all fixpoints of a fuzzy closure operator. IEEE Trans. Fuzzy Syst. **18**(3), 546–557 (2010)
4. Bělohlávek, R., Vychodil, V.: Attribute implications in a fuzzy setting. In: Missaoui, R., Schmidt, J. (eds.) ICFCA 2006. LNCS (LNAI), vol. 3874, pp. 45–60. Springer, Heidelberg (2006). https://doi.org/10.1007/11671404_3
5. Bocharov, A., Gnatyshak, D., Ignatov, D.I., Mirkin, B.G., Shestakov, A.: A lattice-based consensus clustering algorithm. In: International Conference on Concept Lattices and Their Applications, vol. CLA2016, pp. 45–56 (2016)
6. Carbonnel, J., Bertet, K., Huchard, M., Nebut, C.: FCA for software product line representation: mixing configuration and feature relationships in a unique canonical representation. Discrete Appl. Math. **273**, 43–64 (2020)
7. Castellanos, A., Cigarrán, J., García-Serrano, A.: Formal concept analysis for topic detection: a clustering quality experimental analysis. Inf. Syst. **66**, 24–42 (2017)
8. Chemmalar Selvi, G., Lakshmi Priya, G.G., Joseph, R.B.: A FCA-based concept clustering recommender system. In: Vinh, P.C., Rakib, A. (eds.) ICCASA/ICTCC -2019. LNICST, vol. 298, pp. 178–187. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-34365-1_14
9. Cigarrán, J., Castellanos, Á., García-Serrano, A.: A step forward for Topic Detection in Twitter: an FCA-based approach. Expert Syst. Appl. **57**, 21–36 (2016)
10. Cordero, P., Enciso, M., Mora, A., de Guzmán, I.P.: SL_{FD} logic: elimination of data redundancy in knowledge representation. In: Garijo, F.J., Riquelme, J.C., Toro, M. (eds.) IBERAMIA 2002. LNCS (LNAI), vol. 2527, pp. 141–150. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-36131-6_15
11. Cordero, P., Enciso, M., Bonilla, A.M., Ojeda-Aciego, M.: Bases via minimal generators. In: Proceedings of the International Workshop "What can FCA do for Artificial Intelligence?" (FCA4AI at IJCAI 2013), Beijing, China, 5 August 2013, pp. 33–36 (2013)
12. Cordero, P., Enciso, M., Mora, Á., Ojeda-Aciego, M.: Computing minimal generators from implications: a logic-guided approach. In: Proceedings of Concept Lattices and Applications, CLA 2012. pp. 187–198 (2012)
13. Demko, C., Bertet, K., Faucher, C., Viaud, J.F., Kuznetsov, S.O.: NextPriorityConcept: a new and generic algorithm computing concepts from complex and heterogeneous data. Theor. Comput. Sci. **845**, 1–20 (2020)
14. Diatta, J.: A relation between the theory of formal concepts and multiway clustering. Pattern Recogn. Lett. **25**(10), 1183–1189 (2004)

15. Dua, D., Graff, C.: UCI machine learning repository (2017). http://archive.ics.uci.edu/ml
16. Ganter, B., Wille, R.: Formal Concept Analysis. Springer, Heidelberg (1999). https://doi.org/10.1007/978-3-642-59830-2
17. Delugach, H.S., Stumme, G. (eds.): ICCS-ConceptStruct 2001. LNCS (LNAI), vol. 2120. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44583-8
18. Ganter, B., Rudolph, S., Stumme, G.: Explaining data with formal concept analysis. In: Krötzsch, M., Stepanova, D. (eds.) Reasoning Web. Explainable Artificial Intelligence. LNCS, vol. 11810, pp. 153–195. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-31423-1_5
19. Hahsler, M.: Grouping association rules using lift. In: Proceedings of 11th INFORMS Workshop on Data Mining and Decision Analytics (DMDA 2016) (2016)
20. Hamming, R.W.: Error detecting and error correcting codes. Bell Syst. Tech. J. **29**(2), 147–160 (1950)
21. Jaccard, P.: The distribution of the flora in the alpine zone. New Phytol. **11**(2), 37–50 (1912)
22. Kashnitsky, Y., Ignatov, D.I.: Can FCA-based recommender system suggest a proper classifier? In: Proceedings of the International Workshop "What can FCA do for Artificial Intelligence?" (FCA4AI at IJCAI 2014), p. 17 (2014)
23. Kaufman, L., Rousseeuw, P.J.: Partitioning around medoids (program PAM). In: Finding Groups in Data: An Introduction to Cluster Analysis, vol. 344, pp. 68–125 (1990)
24. Konecny, J.: Attribute implications in L-concept analysis with positive and negative attributes: validity and properties of models. Int. J. Approx. Reason. **120**, 203–215 (2020)
25. Kruskal, J.B.: Multidimensional Scaling, no. 11. Sage, Los Angeles (1978)
26. Kumar, C.A.: Fuzzy clustering-based formal concept analysis for association rules mining. Appl. Artif. Intell. **26**(3), 274–301 (2012)
27. MacQueen, J., et al.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Oakland, CA, USA, vol. 1, pp. 281–297 (1967)
28. Medina, J., Ojeda-Aciego, M., Ruiz-Calviño, J.: Formal concept analysis via multi-adjoint concept lattices. Fuzzy Sets Syst. **160**(2), 130–144 (2009)
29. Melo, C., Mikheev, A., Le Grand, B., Aufaure, M.A.: Cubix: a visual analytics tool for conceptual and semantic data. In: Proceedings - 12th IEEE International Conference on Data Mining Workshops, ICDMW 2012, pp. 894–897 (2012)
30. Missaoui, R., Ruas, P.H.B., Kwuida, L., Song, M.A.J.: Pattern discovery in triadic contexts. In: Alam, M., Braun, T., Yun, B. (eds.) ICCS 2020. LNCS (LNAI), vol. 12277, pp. 117–131. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-57855-8_9
31. Mora, Á., Cordero, P., Enciso, M., Fortes, I., Aguilera, G.: Closure via functional dependence simplification. Int. J. Comput. Math. **89**(4), 510–526 (2012)
32. Priss, U.: Formal concept analysis in information science. Ann. Rev. Inf. Sci. Technol. **40**(1), 521–543 (2006)
33. Ravi, K., Ravi, V., Prasad, P.S.R.K.: Fuzzy formal concept analysis based opinion mining for CRM in financial services. Appl. Soft Comput. J. **60**, 786–807 (2017)
34. Rodríguez-Lorenzo, E., Bertet, K., Cordero, P., Enciso, M., Mora, A., Ojeda-Aciego, M.: From implicational systems to direct-optimal bases: a logic-based approach. Appl. Math. Inf. Sci. **2**, 305–317 (2015)

35. Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J. Comput. Appl. Math. **20**, 53–65 (1987)
36. Strehl, A., Gupta, G.K., Ghosh, J.: Distance based clustering of association rules. In: Proceedings ANNIE, vol. 9, pp. 759–764 (1999)
37. Stumme, G., Maedche, A.: FCA-MERGE: bottom-up merging of ontologies. In: IJCAI International Joint Conference on Artificial Intelligence, pp. 225–230 (2001)
38. Sumangali, K., Aswani Kumar, Ch.: Concept lattice simplification in formal concept analysis using attribute clustering. J. Ambient Intell. Humaniz. Comput. **10**(6), 2327–2343 (2018). https://doi.org/10.1007/s12652-018-0831-2