






## Research Article

# A Formal Concept Analysis Approach to Cooperative Conversational Recommendation

Pablo Cordero<sup>1, </sup>, Manuel Enciso<sup>2, </sup>, Ángel Mora<sup>1, </sup>, Manuel Ojeda-Aciego<sup>1,\*, </sup>, Carlos Rossi<sup>2, </sup>

<sup>1</sup>Department Matemática Aplicada, Universidad de Málaga, ETSI Informática, Blv. Louis Pasteur 35, 29071 Málaga, Spain

<sup>2</sup>Department Lenguajes y Ciencias de la Computación, Universidad de Málaga, ETSI Informática, Blv. Louis Pasteur 35, 29071 Málaga, Spain

## ARTICLE INFO

### Article History

Received 21 Mar 2020

Accepted 04 Aug 2020

### Keywords

Formal concept analysis

Simplification logic

Conversational recommender systems

Group recommender systems

## ABSTRACT

We focus on the development of a method to guide the choice of a set of users in an environment where the number of features describing the items is high and user interaction becomes laborious. Using the framework of formal concept analysis, particularly the notion of implication between attributes, we propose a method strongly based on logic which allows to manage the users' preferences by following a conversational paradigm. Concerning complexity, to build the conversation and provide updated information based on the users' previous actions (choices) the method has polynomial delay.

© 2020 The Authors. Published by Atlantis Press B.V.

This is an open access article distributed under the CC BY-NC 4.0 license (<http://creativecommons.org/licenses/by-nc/4.0/>).

## 1. INTRODUCTION

Current information systems deal with big amounts of data. In these systems two issues have to be addressed: the management of high-dimensional data and the interaction with the users. In this paper we focus on the second one. When users search one or several items from sets with very high cardinality, they can become overwhelmed in many cases. Several techniques have been used to approach this problem: within the general area of Information Retrieval, techniques such as Information Filtering [1] help the user to locate items that meet her/his requirements. In a similar way, recommender systems [2] try to predict which items are more suitable for the user, thus saving her/him having to select the items.

This work focuses on a subproblem that arises when working with datasets of high dimensionality, i.e., datasets with a high number of attributes. This complicates the use of the applications, because the user is forced to choose from a high volume of information (attribute values) in order to complete the search for the items that meet their requirements. For example, if a dataset of diseases has one-hundred attributes (representing symptoms or signals), the user (physician or patient) will have to check dozens of such symptoms to get a diagnosis. In this work, we will approach the problem of dimensionality in the context of cooperative information systems. More specifically, we provide an algorithm for *conversational cooperative filtering* in high-dimensional datasets.

Another typical use case can be the selection of activities in a tourist destination by a group of travelers, or a process of cooperative diagnosis by a group of physicians. This is also the case, for instance, in e-commerce [2,3], or online travel agencies [4] among other areas. We refer the reader to Section 5 where we review the literature about conversational recommendation.

Our contribution uses formal concept analysis (FCA) as the underlying formal framework [5]. The solid mathematical basis of FCA is built on lattice theory and Galois connections, allowing to mathematically formalize the notion of “concept” (a general idea that corresponds to some type of entity and that can be characterized by some essential features of the class). The mechanisms used to extract these concepts from a dataset in FCA also allow us to hierarchically organize these concepts in the so-called “concept lattice.” An important aspect to emphasize is that the concept lattice captures all the implicit knowledge that can be deduced from a context, which can be described alternatively as a set of implications. The notion of implication is also used in many very different areas, for instance, implications are called functional dependencies in database theory, or the well-known if-then rules in logic programming, just to name a few. In essence, an “attribute implication” is an expression of type  $A \rightarrow B$  where  $A$  and  $B$  are sets of attributes/properties, and we say that this is fulfilled if every object that has the attributes of  $A$  has those of  $B$  as well.

In previous works [6–8], several logic-based algorithms have been introduced to efficiently manage implications with different purposes (redundancy removal, transformation of implicational sets, etc.). In this paper, given a dataset, we extract a knowledge-base in

\* Corresponding author. Email: [aciego@uma.es](mailto:aciego@uma.es)

terms of attribute implications and the minimal generators of this knowledge-base are used to drive the recommendation process.

From the procedural point of view, we propose a conversational approach. In this sense, it should be explained that in “standard” (nonconversational) filtering or recommendation processes the user indicates all his preferences at the same time, in a single interaction with the system. On the other hand, in conversational filtering or recommendation processes [9,10], the user’s requirements are elicited through an iterative procedure, so that in each step of the dialogue the user indicates his preferences in relation to one (or several) attributes of the dataset.

This conversational scheme is especially suitable when working with high-dimensionality datasets, since otherwise the user would have to provide in a single step the values of tens or hundreds of attributes, which is very laborious. In addition, if a community of users is involved in the recommendation, the so-called group recommendation, a more complex situation arises: we have to combine the different choices of all the users into a common selection.

This work is strongly based on the results provided in [11], where a logic-based algorithm to infer the minimal generators of an implicational system was introduced. These minimal generators are used to build an iterative dialogue. In each step, the members of the group indicate their desired values for some of the attributes of the items they are searching for. Concerning efficiency, although the computation of all minimal generators is NP-hard, the problem we are dealing with does not need all the minimal generators; our approach makes use of an on-demand computation of minimal generators by means of an algorithm with polynomial delay. The main advantage is that it significantly reduces the number of steps in the dialogue, saving users the need to specify their choices for many of the attributes, what makes it suitable to be applied in interactive systems.

Moreover, the algorithm includes aggregation functions to combine the preferences of each group member. A great variety of such functions exist in the literature; depending on the selected function, the behavior of the algorithm differs, e.g., giving greater or lesser relevance to possible extreme values in the preferences of some member of the group.

This work further develops the approach introduced in [12], in which a technique to decrease dialog length in a conversational filtering system was presented. Two important contributions are introduced in this paper: firstly, our new algorithm is proactive, while the previous one was reactive, namely, the algorithm proposes promising directions in the search, instead of reacting to user’s choices; secondly, the new algorithm devises a cooperative process oriented to a group of users, whereas the previous one implemented just an individual filtering process.

The structure of the paper is the following: Section 2 shows the theoretical background and introduces a logic-based method to enumerate minimal generators, which is a key point in our conversational process. Section 3 introduces the main contribution of the paper, i.e., the cooperative recommender method. In Section 4 its empirical evaluation, based on a collection of both synthetic and real datasets, is presented. In Section 5 a brief summary of related works is presented. We end the paper with conclusions and prospects for future work.

## 2. PRELIMINARIES

We recall below some basic notions of FCA (see [5]) which will be used in the extraction and management of knowledge from a dataset.

Datasets are interpreted in terms of *formal contexts*, i.e., triplets  $\mathbb{K} = \langle G, M, I \rangle$  where  $G$  and  $M$  are nonempty finite sets and  $I \subseteq G \times M$  is a binary relation between  $G$  and  $M$ . The elements in  $G$  and  $M$  are called objects and attributes, respectively.

Each formal context  $\mathbb{K}$  defines two derivation operators, both denoted with a prime ‘, namely

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

For each  $A \subseteq G$ ,  $A'$  is the set of attributes common to the objects in  $A$  and, if  $B$  is an attribute set,  $B'$  is the set of objects which have all the attributes in  $B$ .

The pair of derivation operators form a Galois connection between  $\langle 2^G, \subseteq \rangle$  and  $\langle 2^M, \subseteq \rangle$  and, as a result, any of their compositions turns out to be a closure operator, and their fixpoints (termed *formal concepts*) form a complete lattice, known as the *concept lattice* associated with the context. The number of elements in this lattice can be exponential with respect to the size of the context.

The main goal of FCA is to extract knowledge from the context allowing to reason about them. One of the ways to represent knowledge is by means of the concept lattice. Another equivalent alternative knowledge representation, more suitable to define reasoning methods, is given in terms of attribute implications.

**Definition 1.** [*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  $A' \subseteq B'$  (or equivalently,  $B \subseteq A''$ ).

The sets  $A$  and  $B$  will be called, respectively, the *left-hand side* (LHS) and the *right-hand side* (RHS) of the implication  $A \rightarrow B$ .

The most reasonable way to handle attribute implications is in terms of logic, and we will work with the Simplification Logic  $SL$ , which we briefly recall below introducing its syntax, semantics, and an axiom system (further details and proofs can be found in [7]).

**Definition 2.** [*Syntax and Semantics of  $SL$* ]

- Given a non-empty finite alphabet  $S \cup \{\rightarrow\}$ , the language of  $SL$  is  $\mathcal{L}_S = \{A \rightarrow B \mid A, B \subseteq S\}$ . Elements in  $S$  are called attributes.
- A context  $\mathbb{K} = \langle G, M, I \rangle$  is an *interpretation* for  $\mathcal{L}_S$  when  $M = S$ . A *model* for  $A \rightarrow B \in \mathcal{L}_S$  is an interpretation  $\mathbb{K}$  for  $\mathcal{L}_S$  such that  $A' \subseteq B'$ . It is denoted by  $\mathbb{K} \models A \rightarrow B$ .

As usual, for a context  $\mathbb{K}$  and an implicational system  $\Sigma$ ,  $\mathbb{K} \models \Sigma$  means that  $\mathbb{K}$  is a model for every implication in  $\Sigma$  and  $\Sigma \models A \rightarrow B$  denotes that every model for  $\Sigma$  is also a model for  $A \rightarrow B$ . In

addition, two implicational systems  $\Sigma_1$  and  $\Sigma_2$  are said to be equivalent, denoted by  $\Sigma_1 \equiv \Sigma_2$ , when the models of both implicational systems coincide (i.e.,  $\mathbb{K} \models \Sigma_1$  iff  $\mathbb{K} \models \Sigma_2$ ).

We introduce the axiom system of *SL* in which, in order to distinguish between language and metalanguage inside implications,  $AB$  means  $A \cup B$  and  $A \setminus B$  denotes the set difference.

**Definition 3.** [Axiom system] **SL** considers the *reflexivity* axiom  $\vdash A \rightarrow A$ ; together with the following inference rules, which are called *fragmentation*, *composition*, and *simplification* respectively.

- $\{A \rightarrow BC\} \vdash A \rightarrow B$ ;
- $\{A \rightarrow B, C \rightarrow D\} \vdash AC \rightarrow BD$ ;
- $\{A \rightarrow B, C \rightarrow D\} \vdash (C \setminus B) \rightarrow D$  when  $A \subseteq C$  and  $A \cap B = \emptyset$ .

We use the standard notation  $\Sigma \vdash A \rightarrow B$  to denote that  $A \rightarrow B$  can be derived or inferred from  $\Sigma$  by using the axiom system.

The axiom system given above is sound and complete [7], i.e., for all  $A \rightarrow B \in \mathcal{L}_S$  and  $\Sigma \subseteq \mathcal{L}_S$ ,

$$\Sigma \models A \rightarrow B \text{ iff } \Sigma \vdash A \rightarrow B.$$

Knowledge is represented by a set of implications  $\Sigma$  which entail all those that hold in  $\mathbb{K}$  (a set of implications  $\Sigma$  that characterises those that hold in  $\mathbb{K}$ , see [13]). Formally,  $\Sigma$  is a *basis* for  $\mathbb{K}$  if the following condition holds: for all  $A \rightarrow B \in \mathcal{L}_S$ ,

$$\mathbb{K} \models A \rightarrow B \text{ iff } \Sigma \vdash A \rightarrow B.$$

The inference rules of *SL* are enough to compute all the models and their main advantage is that they can be seen as logical equivalence rules (see Proposition 2), providing a transformation stronger than that of the usual inference rules. These equivalences have opened the door to the development of several automated methods to manage implications by using Logic, for instance: to compute the syntactic attribute closure [7], to reduce the specification of implications [6], to compute several kinds of basis of implications [8,14] and, the one that has inspired this work, to compute all minimal generators and closed sets [15].

In a previous work [11], we devised a method to enumerate all minimal generators. Their inherent order structure allows for managing information similar to that provided by a concept lattice, but in a more succinct form. Concept lattices hierarchically present how objects are grouped together, characterized by a set of common attributes, corresponding with a closed set of attributes. These closed sets can also be identified in the ordered structure of minimal generators, allowing a small representation to be used as the core of our method.

We close this section by formally introducing the notion of closure of a set of attributes which, by the soundness and completeness theorem, is strongly related with the closure in the concept lattice.

**Definition 4.** Given  $\Sigma \subseteq \mathcal{L}_S$  and  $A \subseteq S$ , the (syntactic) closure of  $A$  with respect to  $\Sigma$  is the largest subset of  $S$ , denoted  $A_\Sigma^+$ , such that  $\Sigma \vdash A \rightarrow A_\Sigma^+$ .

The mapping  $(-)_\Sigma^+ : 2^S \rightarrow 2^S$  is a closure operator on  $\langle 2^S, \subseteq \rangle$ . Moreover, due to the soundness and completeness results, if  $\Sigma$  is a basis for  $\mathbb{K}$ , for all  $A \subseteq M$  we have that  $A'' = A_\Sigma^+$ .

Hereinafter, when no confusion arises, we omit the subscript (i.e., we write  $A^+$  instead of  $A_\Sigma^+$ ).

Our contribution is strongly based on the following theorem and proposition.

**Theorem 1.** [Deduction Theorem] [7] Let  $A \rightarrow B \in \mathcal{L}_S$  and  $\Sigma \subseteq \mathcal{L}_S$ . Then,

$$\begin{aligned} \Sigma \vdash A \rightarrow B & \text{ iff } B \subseteq A_\Sigma^+ \\ & \text{ iff } \{\emptyset \rightarrow A\} \cup \Sigma \vdash \{\emptyset \rightarrow B\} \end{aligned}$$

**Proposition 2.** [7] For all  $A, B, C \in M$ , the following equivalences hold:

- **Eq. I:**  $\{\emptyset \rightarrow A, B \rightarrow C\} \equiv \{\emptyset \rightarrow AC\}$  when  $B \subseteq A$ .
- **Eq. II:**  $\{\emptyset \rightarrow A, B \rightarrow C\} \equiv \{\emptyset \rightarrow A\}$  when  $C \subseteq A$ .
- **Eq. III:**  $\{\emptyset \rightarrow A, B \rightarrow C\} \equiv \{\emptyset \rightarrow A, B \setminus A \rightarrow C \setminus A\}$  otherwise.

A linear algorithm for computing the closure  $A_\Sigma^+$  which, in addition, provides a reduced set of implications that can be seen as the knowledge in the system beyond this closure was given in [7]. Later, in [16], we presented how *SL* can be used as a basis to enumerate all closed sets and all minimal generators ( $X$  is a minimal generator if  $Y \subsetneq X$  implies  $Y^+ \subsetneq X^+$ ), and to manage social network information. The function *NextMinGen* introduced in this section corresponds to a slight modification of the closure algorithm mentioned above which computes and outputs the minimals of the LHSs, instead of giving the closure. In [17] it was proved that any minimal generator contains at least a minimal LHS. Concerning complexity, in [7] it was proved that the algorithm for the closure using *SL* (the repeat loop in *NextMinGen*) is  $\mathcal{O}(n)$ , where  $n = \sum_{A \rightarrow B \in \Sigma} (|A| + |B|)$ . On the other hand, the cost of computing the minimal LHSs of  $\Gamma$  is  $\mathcal{O}(wm)$ , where  $m = |\Gamma|$  and  $w = |\text{Next}|$ . Note that  $w \leq m \leq \frac{n}{2}$ .

The following example illustrates the procedure to obtain closures and the corresponding reduced set of implications using the algorithm from [7].

**Example 1.** Given  $\Sigma = \{a \rightarrow c, bc \rightarrow d, c \rightarrow ae, d \rightarrow e\}$  and  $A = \{c, e\}$ . The closure algorithm returns  $A^+ = \{a, c, e\}$  together with the reduced set of implications  $\{b \rightarrow d\}$ . Figure 1 summarizes its trace, showing the closed set in the left column and the remaining set of implications in the right one. Each row corresponds to an iteration in the closure algorithm, traversing the set of implications. The attributes that are removed in each iteration are crossed out.

### 3. OUR PROPOSAL

As stated in the introduction, we introduce a group conversational recommender system following a cooperative paradigm. The usage scenario is that a group of users would like to get a common recommendation, for instance, a group of tourists would like to find a restaurant where gathering together for dinner. Our approach is based on a conversational paradigm, which allows to reach the recommended item by means of an interaction with the users who dialog with the system in order to get a recommendation. In the conversation, preference elicitation is carried out by asking directly to the users.

**Function** NextMinGen( $X, \Sigma$ )

---

**input** :  $X$ , a set of attributes selected by the user;  
 $\Sigma$ , a set of implications describing the knowledge in the system;  
**output**:  $Next$ , a set of attribute sets which are the choices of the next step in the guided search;  
 $\Gamma$ , a reduced set of implications;  
**repeat**  
   $\Gamma := \Sigma$ ;  $\Sigma := \emptyset$   
  **foreach**  $A \rightarrow B \in \Gamma$  **do**  
    **if**  $A \subseteq X$  **then**  $X := X \cup B$   
    **else if**  $B \not\subseteq X$  **then**  $\Sigma := \Sigma \cup \{A \setminus X \rightarrow B \setminus X\}$   
**until**  $\Gamma = \Sigma$   
 $Next := \text{Minimal}s\{A \subseteq M \mid A \rightarrow B \in \Gamma \text{ for some } B \subseteq M\}$   
**return** ( $Next, \Gamma$ )

---

Guide	$\Sigma$			
$\emptyset \rightarrow ce$	$a \rightarrow c$	$bc \rightarrow d$	$c \rightarrow ae$	$d \rightarrow e$
$\emptyset \rightarrow ce$	$a \rightarrow \phi$	$b\phi \rightarrow d$	$\phi \rightarrow a\phi$	$d \rightarrow \phi$
$\emptyset \rightarrow ace$	$b \rightarrow d$			

**Figure 1** | Example of the simplification closure method.

In this situation some problems arise: how to combine the different user preferences, how to guide the conversation, how to deal with a joint dialogue among the system and the users, how to manage the (possible) information overwhelming of users when offered to choose among a large number of attributes, etc.

To begin with, the information is structured in a binary table where items to be recommended correspond to rows (objects in FCA), and attributes of these items correspond to columns. FCA provides a set of efficient methods to extract a set of rules which represents the knowledge in a suitable way.

The general overview of our approach is the following: the system uses the implicit knowledge in the set of rules stated above, and applies an iterative process in which each iteration can be described, at a high level, as follows:

- The system shows just a subset of attributes to avoid information overwhelming. For example, to decide a restaurant for having dinner the users are given a selection of attributes to choose from. These attributes are those which might better discriminate other attributes and shorten the conversation; for this, we focus on the attributes in the LHSs of the rules and, specifically, consider those that belong to some minimal LHS.
- Each person in the group chooses a set of attributes that suits him best, adding a degree of preference to the selected items. Following with the previous example, if the items were restaurants, the attributes could be vegetarian cuisine, disabled access, parking, etc.
- The preferences of all group members are aggregated taking into account the knowledge represented in the whole set of rules. The aggregation function can be independently customized for each problem.

- The set of rules is refined, by using  $SL$  and the previous aggregated preferences, in order to provide the subset of attributes to be shown in the next iteration.
- The resulting items or objects (restaurants in our example) are shown. It is obvious that in each iteration of the dialog, the number of items is decreased.
- If the number of the resulting items is acceptable for the users, they can decide to stop the process.

It is worth mentioning that we achieve two important benefits with this approach: on the one hand, we avoid users to express their preferences on large sets of attributes and, on the other hand, the length of the dialogue (number of steps of conversational recommendation process) is shortened with respect to the approach by [15]. These benefits are confirmed by the results obtained in the evaluation of our method (see Section 4).

In order to provide the specific details of the proposed method, the following key points need to be taken into account. The notion of *minimal generator* is the crux of our contribution in this paper, since they are used to obtain the selection of attributes to be shown to users in each iteration. The method uses Simplification Logic  $SL$ , and is implemented in terms of the function NextMinGen.

As stated above, the input of the method is a binary dataset with the information of the items to be recommended. We consider a set of items  $G$  (objects) having a set of features  $M$  (attributes), represented as a formal context  $\mathbb{K}$ . As stated before, our cooperative environment allows simultaneous participation of a group of users, each one choosing a set of features that suits him best.

The workflow of the algorithm is described in Figure 2. It starts by inferring the implicational system  $\Sigma$  from the formal context  $\mathbb{K}$  by using the ARules package introduced in [18] (*pre-processing stage*). Then each user selects a set of attributes representing the initial needs. Given the choices of all the users, called  $A \subseteq M$ , the algorithm adds the formula  $\emptyset \rightarrow A$  to the implicational set  $\Sigma$  and the function NextMinGen provides, in each step, several sets of attributes, each one corresponding to a promising way to reach a solution for the previous search. The users can decide to stop the algorithm when the set of objects that have been reached is manageable; i.e., when the query to the data fulfilling all the selected

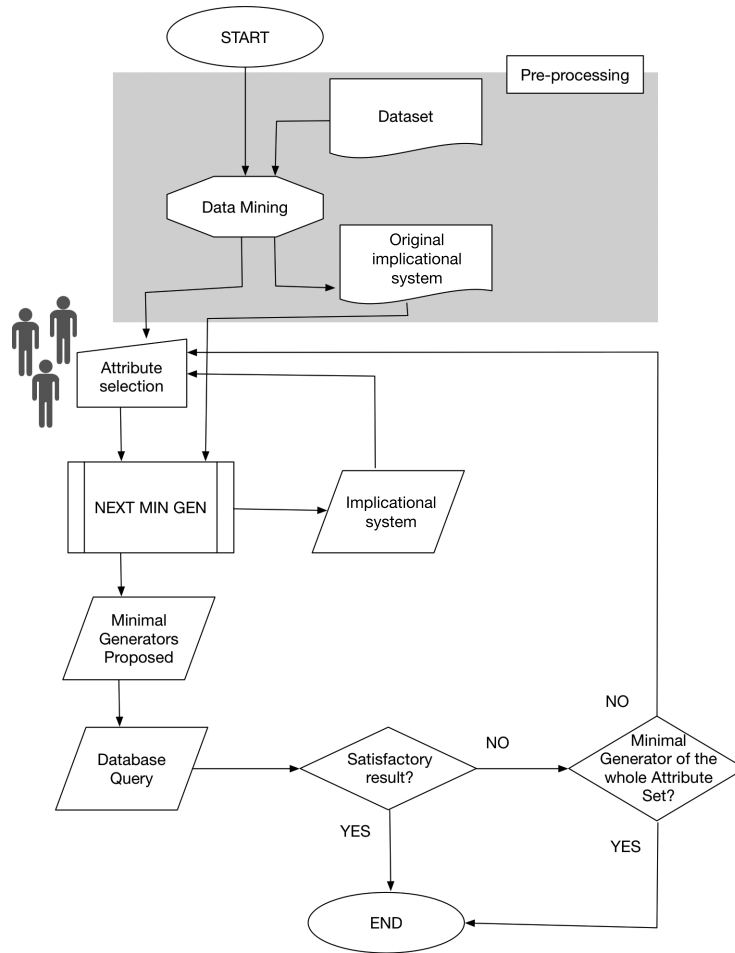


Figure 2 | Workflow of the algorithm.

attributes returns a number of items suitable for their needs. Otherwise, the algorithm requires the users to make a new choice and the process continues iteratively.

Figure 3 shows the activity diagram (using UML notation) of the implementation of the recommender system based on our algorithm, which is described below:

**Step #1: attribute filtering.**

The method filters the attributes to be presented to the user by selecting just the minimal LHSs (with respect to set inclusion) of the basis  $\Sigma$  of the set of rules, denoted  $MiLHS(\Sigma)$ . The results presented in [15] ensure that the elements in  $MiLHS(\Sigma)$  are the smallest sets of attributes providing the minimal generators of the next level, on which we can build the next step of the conversation. Then, each user is asked to assign an independent degree to each attribute from  $MiLHS(\Sigma)$ , (s)he is interested in.

We write  $\Xi(\Sigma)$  to refer to the union of the sets in  $MiLHS(\Sigma)$ . Let the finite set  $D = \{d_i \mid i \in \Lambda\}$  be the set of degrees and let  $K$  be a set of user index.

Assume that each user  $k \in K$  provides his preferences with the set  $P(U_k) = \{(x, d_x^k) \mid x \in \Xi(\Sigma), d_x^k \in D\}$ . We consider a *neutral*<sup>1</sup> degree for those attributes not selected by any user.

<sup>1</sup>The term “neutral” indicates the neutral element for the aggregation function we will use in the following steps.

**Step #2: grouping user preferences.**

In this step, all the personal preferences provided by the users ( $P(U_k), k \in K$ ) have to be unified. Our main goal is to obtain a set  $\{(x, d_x) \mid x \in \Xi(\Sigma), d_x \in D\}$  in which, for each attribute,  $d_x$  represents a suitable aggregation of the preferences of all the users. The specific aggregation function to be used is problem dependant and could be, for instance, any OWA operator chosen according to some expected behavior: maximum user satisfaction, minimal impact on extreme user preferences, etc. If the aggregator used is  $O_1$ , we obtain the degree associated with each attribute by  $d_x = O_1(d_x^k \mid k \in K)$ .

**Step #3: top-m minimal generators.**

For all  $X \in MiLHS(\Sigma)$ , its degree is defined as  $D_X = O_2(d_x \mid x \in X)$  where  $O_2$  is an aggregator, not necessarily the same than  $O_1$ . Note that  $x$  denotes an attribute whereas  $X$  denotes a set of attributes (corresponding with a LHS). This step ends with the selection of the top-m minimal LHS with highest  $D_X$  degrees.

**Step #4: top-m minimal generators improved.**

These top-m minimal generators could be a good solution to guide the search, but we further improve them by considering the closure of each minimal generator. The goal of this step is to increase the degree  $D_X$  of those minimal LHSs  $X$  whose closure contains the attributes selected in Step#1 by some users (these new attributes can belong to another LHS). The degrees  $D_X$  are then updated using the

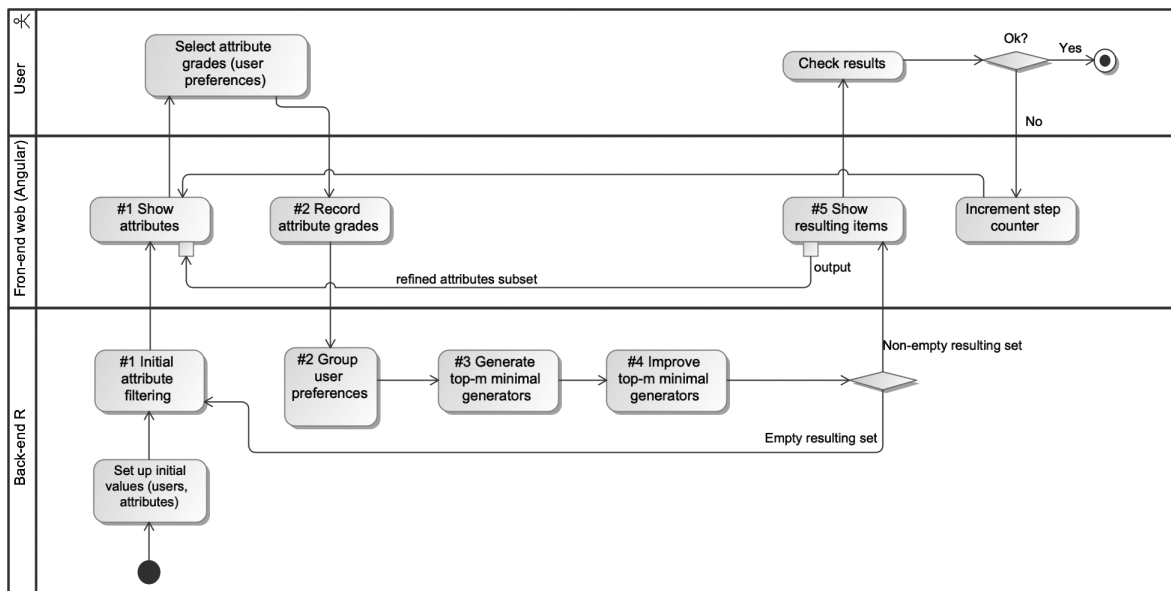


Figure 3 | Activity diagram of the recommender system.

information in the closure; formally, for all  $X \in \text{MilHS}(\Sigma)$  we consider  $D_{X^+} = O_2(d_x \mid x \in X^+)$ .

Now, the method considers the set of attributes  $X^+$  having the highest degree  $D_{X^+}$  provided that the set of objects holding all the attributes in  $X$  (the extent  $X'$  in terms of FCA) is nonempty. Otherwise, the output of the method is the set of objects of the previous iteration.

#### Step #5: Resulting items.

If the users consider that the cardinality of  $X'$  is manageable to make a decision, then they can stop the process.

Otherwise, the recommender system initiates another iteration from Step#1, in which NextMinGen reduces the possible choice of attributes to  $M \setminus X^+$  and provides a simplified implicational set at no extra cost.

#### Complexity considerations.

Steps#4 and #5 make use of the function NextMinGen which, as mentioned above, is linear with respect to the size of the set of implications. The rest of steps have all linear complexity.

We finish the section with an illustrative toy example:

**Example 2.** Let  $U = \{U_1, U_2\}$  be a set of users,  $M = \{a_1, \dots, a_6\}$  a set of attributes and  $\Sigma = \{a_1 \rightarrow a_3, a_2 a_4 \rightarrow a_3, a_2 \rightarrow a_1 a_3, a_4 \rightarrow a_6, a_5 \rightarrow a_6\}$  a set of implications. The set of degrees is the interval  $[0, 1]$ , the parameter  $m$  in top- $m$  is 2, and the aggregation functions, for simplicity, are the arithmetic mean.

The set  $\Xi(\Sigma) = \{a_1, a_2, a_4, a_5\}$  is computed and the users are asked to assign preferences to these attributes (Step#1) providing, for instance:  $P(U_1) = \{(a_1, 0.6), (a_3, 0.1)\}$  and  $P(U_2) = \{(a_2, 0.4), (a_3, 0.2)\}$ .

The set of unified degrees (Step#2) is  $\{(a_1, 0.6), (a_2, 0.4), (a_3, 0.15), (a_4, 0), (a_5, 0)\}$ , ordering  $\text{MinLHS}(\Sigma)$  in (Step#3) as follows:

$[\{(a_1\}, 0.6), \{(a_2\}, 0.4), \{(a_2, a_4\}, 0.2), \{(a_4\}, 0), \{(a_5\}, 0)]$ .

Since  $m = 2$  the method provides the minimal generators  $\{a_1\}$  and  $\{a_2\}$ . Note that since  $a_3 \notin \text{MinLHS}(\Sigma)$  this attribute is not relevant even though it had been selected by the users.

The closure of the two first minimal generators is computed in Step#4:  $a_1^+ = \{a_1, a_3\}$  and  $a_2^+ = \{a_2, a_3, a_1\}$ . The new order in the closed sets is  $(a_2^+, 0.38), (a_1^+, 0.375)$ .

The stage for the new iteration (Step#5) is  $M = a_4, a_5, a_6$  and  $\Sigma = a_4 \rightarrow a_6, a_5 \rightarrow a_6$ . Then, suppose that the users evaluate as follows:  $P(U_1) = \{(a_4, 0.7)\}$ ,  $P(U_2) = \{(a_4, 0.7)\}$ , which directly produces (Step#2) the following order in the elements of  $\text{MinLHS}(\Sigma)$ :  $(a_4^+, 0.7), (a_5^+, 0), (a_6^+, 0)$ .

In (Step#4) we obtain the closure  $a_4^+ = \{a_4, a_6\}$  and the stage for the new iteration is  $M = \{a_5\}$  and  $\Sigma = \emptyset$ , stopping the method.

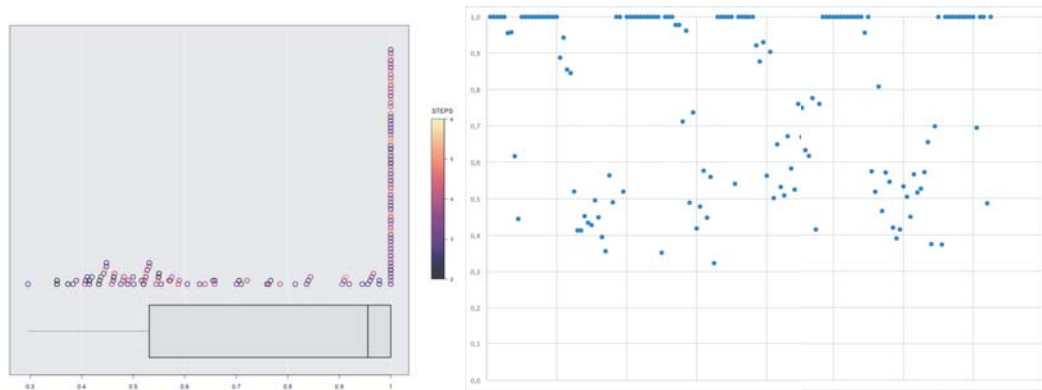
Therefore, the output of the recommender is the set of objects holding the attributes  $a_2$  and  $a_4$ , i.e.,  $\{a_2, a_4\}'$ .

## 4. EVALUATION

The recommender introduced in this paper has been named *Cooperative Filtering*.<sup>2</sup> This section describes the experiments designed to show the advantages of our proposal and evaluate the obtained results. The recommender system has been tested in two different contexts, considering synthetic and real datasets.

The synthetic datasets have been randomly generated in order to show how the method deals with datasets with different sizes. Thus, we have run 329 experiments, the number of attributes ranging from 30 to 80 with an average of 47.69. The number of implications extracted from the datasets ranges from 30 to 200 with an average of 100.

<sup>2</sup> The code is available in <https://github.com/amorabonilla/CooperativeFiltering>. In this repository, the file *readme.md* explains how to use this recommender (and how to install the R packages).



**Figure 4** | Accuracy with respect to the number of steps and average percentage of satisfied preferences per user.

The interaction with the users has been implemented as follows: in each execution, we simulate that three users can choose up to three attributes. Moreover the users select how important these attributes are to them using a 4 point Likert Scale and, using this information about their preferences, the recommender prunes the set of implications and computes which are the attributes to be offered for choice in the next iteration.

The results for the synthetic datasets are very promising: the average length of the dialogue with the users is 3.86 steps. It is remarkable that in the 90% of cases, the length of the dialogue is less than or equal to 4 and the maximum obtained is 7 (see Figure 4).

We have also obtained that the length of the conversation is independent of several parameters related to dimension or size: number of objects, number of attributes, and number of implications. Specifically, the values obtained for Spearman's rank correlation are 0.03, 0.05, and 0.07 respectively. As a result, it seems that the number of steps in the dialogue is mostly related to the structure of the implications, as usual in this type of problems.

In addition to the length of the conversation, another important measure is the accuracy of the results provided by the recommender system. When the system provides a set of objects at the end of the process, we can check how far the characteristics of these objects fit the individual preferences given by the users.

As a general conclusion, the accuracy results are very successful. Figure 4 (right) shows the average of the percentages of satisfaction of preferences per user (the horizontal axis represents just the different experiments). The global average percentage of satisfaction of preferences is 0.7785 which, in our opinion, is a very good value, especially taking into account that it is for a group. The standard deviation is 0.2565 and the values determining the quartiles are the following: 0.219 (min), 0.5117 (25%), 0.8947 (50%), 1 (75%), and 1 (max).

It is worth noting that we have also analyzed the results of accuracy with respect to other dimensions, and we have found that there is no correlation between these values and other parameters of the experiment. Figure 4 (left) shows the relation between level of accuracy and number of steps in the dialog; although, visually, we could think of certain relation between these two dimensions, once again we found that there is no significative correlation between them.

As a second approach, we have also conducted experiments with real datasets.<sup>3</sup> the following three public datasets suitable for treatment with our proposal (see Figure 6 for a summary of the features of each dataset):

- Celeb.<sup>4</sup> The *CelebFaces Attributes Dataset* (containing information about over 200k images of celebrities with 40 binary attribute annotations). This dataset is often used for training and testing models for recognizing facial attributes. The exact size of the dataset is 202599 rows (persons) and 40 columns (attributes of a person). The number of implications extracted is 63533.
- DSK.<sup>5</sup> The dataset *Disease-Symptom Knowledge Database* contains information about diseases and symptoms, with 398 rows (diseases) and 40 columns (symptoms). The number of implications extracted is 10013234.
- HPO.<sup>6</sup> Extracted from the *Human Phenotype Ontology Consortium* dataset and used in [12], also contains information about diseases and symptoms. Its size is 446 rows (diseases) and 985 columns (corresponding to symptoms). The number of implications extracted is 3678.<sup>7</sup>

The obtained results are summarized as follows: concerning the length of the dialogue, our recommender system, always finishes after a small number of steps independently from the characteristics of the datasets (either having many objects or many attributes, different sparsity in the dataset and different number of extracted implications). Figure 7 shows the number of experiments that finalized with a given length of dialogue. Summarizing, in most of the cases the length of the dialogue is two, 85.3%, 98.7%, and 54.4% for the Celeb, DSK, and HPO datasets, respectively. The cases of three interactions in the dialogue is significant just in the HPO dataset

<sup>3</sup>The code and some reproducible examples with the corresponding outputs are available as Rmd and Html files in <https://github.com/amorabonilla/CooperativeFiltering>. More specifically,

<sup>4</sup><https://www.kaggle.com/jessicali9530/Celeb-dataset>

<sup>5</sup><http://people.dbmi.columbia.edu/~friedma/Projects/DiseaseSymptomKB/index.html>

<sup>6</sup><http://www.human-phenotype-ontology.org>

<sup>7</sup>The number of implications depends of the sparsity of the dataset.

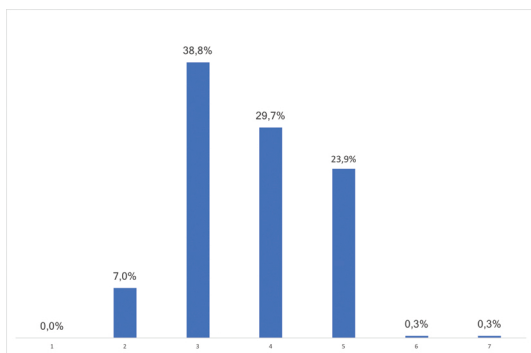
(43.7%) whereas in the other two datasets the percentage is 13.1% and 1.3% respectively. The dialogues four interactions long are just the 1% of the total number (in the three datasets) and the longest interaction was five steps, just in 0,17% of the cases, all of them in the Celeb dataset.

Another interesting conclusion inferred from these experiments is the evolution of the number of attributes, objects, implications, and closure sizes while the conversation is being developed. Figure 8 shows that the first three items decrease and the fourth one increase for the Celeb, DSK, and HPO datasets. *Cooperative Filtering* significantly reduces the number of objects which are filtered and shown to the user. It is also worth mentioning that the method scales adequately in high-dimensions (one thousand attributes for a recommender system). Figure 8 shows how the number of attributes suggested to the users considerably decreases for the three datasets.

To finish with, a recommender system was introduced in [12] where the HPO were also used to experiment with; however, that approach does not consider the collaborative paradigm. The table of Figure 9 shows a comparative with the results obtained there, just measuring the number of steps, the only one considered in that work. *Cooperative Filtering* reduces the number of steps obtained in this previous work: it does not need more than 3 steps in any execution.

## 5. LITERATURE REVIEW

Similar approaches to this work have been adopted in the so-called *conversational* recommender systems [19–23], closely related to *critiquing* recommender systems [24,25]. In these systems, recommendations are incrementally generated by means of a dialogue with the user. In [22], the authors propose a dialogue system similar to ours, although based on different foundations; specifically, they train the recommender with the dialogue state, user



**Figure 5** | Distributions of the number of iterations of the dialogue in synthetic experiments.

	celebA	DSK	HPO
Columns	41	1017	986
Rows	202599	398	446
Sparsity	0.76	0.99	0.99
Implications	4933	11844	814
Attr.Left	40	1016	985

**Figure 6** | Information about the three real datasets.

information and item information and use the factorization machine (FM) methods.

The goal of decreasing the number of steps of a conversational filtering process also appears in works by Trabelsi *et al.* [23], using two models (quantitative and qualitative) of attribute dominance, whereas we use FCA and logic instead. With respect to previous works using FCA, we emphasize the approach of [26] using fuzzy information obtained from a locate-based social network and making recommendations according to the users interests. However, their approach is based on the concept lattice obtained from the dataset and neither does it use any implications or logic.

Jannach *et al.* [27] defend the adequacy of the knowledge-based approach in the conversational process, as we propose as well. Moreover, they use constraint-based reasoning techniques to design a query tightening strategy, similar to our proposal.

Our work is also related to feature selection, an area that has been widely studied in the literature [28]. As it is well-known, its main objective is to identify whether the attributes are meaningful or not. The techniques in this area constitute a good approach to tackle high-dimensionality datasets and they are widely being used to improve the performance of machine learning classification techniques. Different tools have been used in this area, such as mathematical techniques (principal component analysis, linear discriminant analysis, independent component analysis), association rules or evolutionary computing, among others.

Most of these tools follow some noninteractive processing schemes. However, in line with our proposal, other works define an iterative feature selection process. For instance, in [29] we can find a tree-based feature selection process strongly based on fuzzy modeling and fuzzy rules. They obtain good results, but with a significant number of required user interactions when applied to high-dimensional datasets. A similar work is [30], that uses support vector machine as the classification technique. An interactive process is also proposed in [31], where features are processed one by one. Their underlying framework is information theory (mutual information, conditional mutual information, and entropy).

## 6. CONCLUSIONS AND FUTURE WORKS

In this work, we propose a new framework to manage datasets with high dimensionality. Our approach is strongly based on formal techniques such as FCA and implication logic providing a solid basis and, also, a good performance.

We design a cooperative tool to collect the preferences of a set of users, providing a conversational process to reach a group recommendation. Our goal is to generate a conversation with a small number of dialogue steps. The semantic information stored in the minimal generators is used to drive the recommendation process and some aggregation functions are introduced to join the different preferences of the users.

We propose two lines as future works. On the one hand, some FCA extensions can be used to deal with complex information. As inspirational pieces, the work [32] generates recommendations based on association rules drawn from the concept lattice and [33] defines a collaborative recommendation process using fuzzy annotations in the lattice.



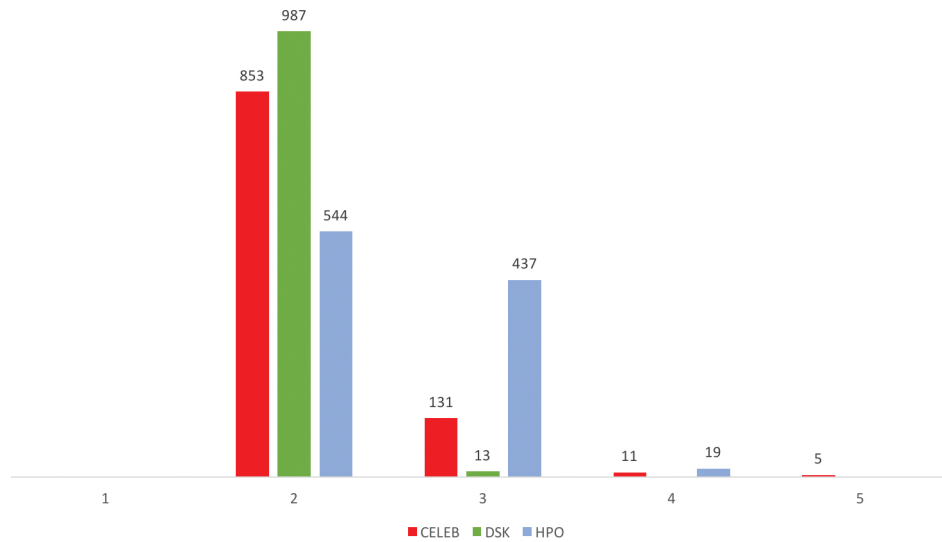


Figure 7 | Distributions of the number of steps of the dialogue in real datasets.

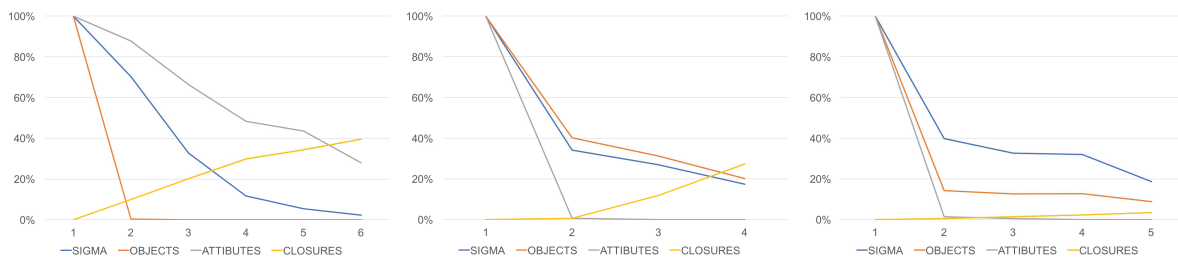


Figure 8 | Evolution of the number of attributes, objects, implications, and closure sizes in the Celeb, DSK, and HPO datasets.

	1 Step	2 Steps	3 Steps	4 Steps	5 Steps
Benito2018	0	68	26	5	1
Cooper.Filt.	0	54.4	43.7	0	0

Figure 9 | Comparison with a previous work.

Last but not least, we intend to characterize a set of OWAs that can be used [34] to set up this method for different environments, providing in this way an adaptive tool for different group idiosyncrasies.

## ACKNOWLEDGMENTS

Partially supported by projects TIN2017-89023-P, CSO2017-82592-R, and PGC2018-095869-B-I00 of the Ministry of Science and Innovation of Spain, and projects UMA2018-FEDERJA-001 and UMA18-FEDERJA-158 of the Junta de Andalucía, all co-funded by the European Regional Development (ERDF).

## CONFLICT OF INTEREST

The authors certify that there is no actual or potential conflict of interest in relation to this article.

## AUTHORS' CONTRIBUTIONS

All authors contributed equally to the publication.

## REFERENCES

- [1] H.K. Azad, A. Deepak, Query expansion techniques for information retrieval: a survey, *Inf. Process. Manag.* 56 (2019), 1698–1735.
- [2] J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez, Recommender systems survey, *Knowl. Based Syst.* 46 (2013), 109–132.
- [3] M. Zhou, Z. Ding, J. Tang, D. Yin, Micro behaviors: a new perspective in e-commerce recommender systems, in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining (WSDM'18)*, ACM, New York, NY, USA, 2018, pp. 727–735.
- [4] I. Garcia, L. Sebastia, E. Onaindia, On the design of individual and group recommender systems for tourism, *Expert Syst. Appl.* 38 (2011), 7683–7692.
- [5] B. Ganter, R. Wille, *Formal Concept Analysis: Mathematical Foundations*, Springer, Berlin, Heidelberg, Germany, 1999.
- [6] P. Cordero, M. Enciso, A. Mora, J.M. Rodríguez-Jiménez, Computing non-redundant sets of functional dependencies via simplification, in *IEEE Symposium on Foundations of Computational Intelligence (FOCI)*, IEEE, Singapore, 2013, pp. 9–14.
- [7] A. Mora, P. Cordero, M. Enciso, I. Fortes, G. Aguilera, Closure via functional dependence simplification, *Int. J. Comput. Math.* 89 (2012), 510–526.
- [8] E. Rodríguez-Lorenzo, K. Bertet, P. Cordero, M. Enciso, A. Mora, Direct-optimal basis computation by means of the fusion of simplification rules, *Discrete Appl. Math.* 249 (2018), 106–119.

- [9] A. Jameson, M. Willemsen, A. Felfernig, M. Gemmis, P. Lops, G. Semeraro, L. Chen, Human decision making and recommender systems, in: F. Ricci, L. Rokach, B. Shapira (Eds.), *Recommender Systems Handbook*, second ed., Springer, Boston, MA, USA, 2015, pp. 611–648.
- [10] N. Tintarev, J. Masthoff, Explaining recommendations: design and evaluation, in: F. Ricci, L. Rokach, B. Shapira (Eds.), *Recommender Systems Handbook*, second ed., Springer, Boston, MA, USA, pp. 353–382.
- [11] P. Cordero, M. Enciso, A. Mora, M. Ojeda-Aciego, Computing minimal generators from implications: a logic-guided approach, in *Proceeding of 9th International Conference on Concept Lattices and Their Applications (CLA)*, CEUR Workshop Proceedings 972, Málaga, Spain, 2012, pp. 187–198. <http://ceur-ws.org/Vol-972/paper16.pdf>
- [12] F. Benito-Picazo, M. Enciso, C. Rossi, A. Guevara, Enhancing the conversational process by using a logical closure operator in phenotypes implications, *Math. Methods Appl. Sci.* 41 (2018), 1089–1100.
- [13] E. Rodríguez-Lorenzo, P. Cordero, M. Enciso, A.M. Bonilla, Canonical dichotomous direct bases, *Inf. Sci.* 376 (2017), 39–53.
- [14] E. Rodríguez-Lorenzo, K.V. Adaricheva, P. Cordero, M. Enciso, A. Mora, Formation of the d-basis from implicational systems using simplification logic, *Int. J. Gen. Syst.* 46 (2017), 547–568.
- [15] F.B. Picazo, P. Cordero, M. Enciso, A. Mora, Minimal generators, an affordable approach by means of massive computation, *J. Supercomput.* 75 (2019), 1350–1367.
- [16] P. Cordero, M. Enciso, A. Mora, M. Ojeda-Aciego, C. Rossi, Knowledge discovery in social networks by using a logic-based treatment of implications, *Knowl. Based Syst.* 87 (2015), 16–25.
- [17] P. Cordero, M. Enciso, A. Mora, I.P. de Guzmán, A tableaux-like method to infer all minimal keys, *Log. J. IGPL.* 22 (2014), 1019–1044.
- [18] M. Hahsler, B. Grun, K. Hornik, ARules - a computational environment for mining association rules and frequent item sets, *J. Stat. Softw.* 14 (2005), 1–25.
- [19] K. Christakopoulou, F. Radlinski, K. Hofmann, Towards conversational recommender systems, in *ACM SIGKDD 2016 Conference on Knowledge Discovery & Data Mining*, San Francisco California USA, 2016, pp. 815–824.
- [20] S.E. Guerrero, M. Salamo, Increasing retrieval quality in conversational recommenders, *IEEE Trans. Knowl. Data Eng.* 24 (2012), 1876–1888.
- [21] D. Jannach, G. Kreutler, Rapid development of knowledge-based conversational recommender applications with ADVISOR suite, *J. Web Eng.* 6 (2007), 165–192. [http://www.configworks.com/dj/Journal\\_WebEngineering\\_2007.pdf](http://www.configworks.com/dj/Journal_WebEngineering_2007.pdf)
- [22] Y. Sun, Y. Zhang, Conversational recommender system, in *The 41st Intl ACM SIGIR Conference on Research & Development in Information Retrieval*, Ann Arbor, MI, USA, 2018, pp. 235–244.
- [23] W. Trabelsi, N. Wilson, D. Bridge, F. Ricci, Preference dominance reasoning for conversational recommender systems: a comparison between a comparative preferences and a sum of weights approach, *Int. J. Artif. Intell. Tools.* 20 (2011), 591–616.
- [24] K. Mccarthy, J. Reilly, L. Mcginty, B. Smyth, Thinking positively - explanatory feedback for conversational recommender systems, in *Proceedings of the ECCBR 2004 Workshops*, Madrid, Spain, 2004. <https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.100.2628>
- [25] F. Ricci, L. Rokach, B. Shapira (Eds.), *Recommender Systems Handbook*, Springer, Boston, MA, USA, 2015.
- [26] J. Medina, K. Pakhomova, E. Ramírez-Poussa, Recommendation Solution for a locate-based social network via formal concept analysis, in: M. Cornejo, L. Kóczy, J. Medina, A. De Barros Ruano (Eds.), *Trends in Mathematics and Computational Intelligence, Studies in Computational Intelligence*, vol. 796, Springer, Cham, Switzerland, 2019, pp. 113–121.
- [27] D. Jannach, M. Zanker, M. Fuchs, Constraint-based recommendation in tourism: a multiperspective case study, *Inf. Technol. Tour.* 11 (2009), 139–155.
- [28] G. Chandrashekar, F. Sahin, A survey on feature selection methods, *Comput. Electr. Eng.* 40 (2014), 16–28.
- [29] A.S. Fialho, F. Cismondi, S.M. Vieira, S.R. Reti, J.M.C. Sousa, S.N. Finkelstein, Data mining using clinical physiology at discharge to predict ICU readmissions, *Expert Syst. Appl.* 39 (2012), 13158–13165.
- [30] S. Shilaskar, A. Ghatol, Feature selection for medical diagnosis: evaluation for cardiovascular diseases, *Expert Syst. Appl.* 40 (2013), 4146–4153.
- [31] K. Yu, X. Wu, W. Ding, J. Pei, Towards scalable and accurate online feature selection for big data, in *2014 IEEE International Conference on Data Mining*, Shenzhen, China, 2014, pp. 660–669.
- [32] C. Zou, D. Zhang, J. Wan, M.M. Hassan, J. Lloret, Using concept lattice for personalized recommendation system design, *IEEE Syst. J.* 11 (2017), 305–314.
- [33] S. Senatore, G. Pasi, Lattice navigation for collaborative filtering by means of (fuzzy) formal concept analysis, in *Proceedings of the 28th Annual ACM Symposium on Applied Computing (SAC'13)*, ACM, New York, NY, USA, 2013, pp. 920–926.
- [34] R. Fuller, On obtaining OWA operator weights: a sort survey of recent developments, in *2007 IEEE International Conference on Computational Cybernetics*, Gammarth, Tunisia, 2007, pp. 241–244.