Explanation of Recommenders Using Formal Concept Analysis

Belen Diaz-Agudo(✉), Marta Caro-Martinez, Juan A. Recio-Garcia, Jose Jorro-Aragoneses, and Guillermo Jimenez-Diaz

Department of Software Engineering and Artificial Intelligence, Universidad Complutense de Madrid, Madrid, Spain
{belend,martcaro,jareciog,jljorro,gjimenez}@ucm.es

Abstract. Formal Concept Analysis is a mathematical approach which enables formalisation of concepts as basic units of human thinking and analysing data in the object-attribute form. In this paper, we propose the use of FCA as a general resource for explanations and apply it to explain the results of recommender systems. Our method is reusable and applicable to different domains. We define different types of explanations by travelling the lattice structure and analyse how the lattice metrics can be used to characterise the different types of user profiles.

Keywords: Explanations · Explainable artificial intelligence · Formal concept analysis · Recommender systems

1 Introduction

Explainable artificial intelligence and case based explanations have become active areas of research in the last few years [13,21,26]. In recommender systems, explanations are essential to improve user trust and persuasion [12,19] and there are different approaches that have been reviewed elsewhere [6,25].

The term *explanation* can be interpreted in two different ways: in AI in general and in recommender systems in particular [1,4,24]. First interpretation refers to *transparency* and deals with explanations as part of the reasoning process itself and with the goal of understanding how the reasoning process works. The other interpretation deals with *justification* or attempting to make a certain reasoning process, or its result, understandable to the user. Recommendations resulting from *content-based strategies* are more comprehensible for users, because they are based on the explicit user profiles. The *content filtering* approach creates a profile for each user or product to characterise its contents and recommends a similar product that matches the user profile. Most of the content-based recommenders typically generate case-based explanations presenting the

Supported by the UCM (Research Group 921330), the Spanish Committee of Economy and Competitiveness (TIN2017-87330-R) and the fundings provided by Banco Santander in UCM (CT17/17-CT17/18) and (CT42/18-CT43/18).

K. Bach and C. Marling (Eds.): ICCBR 2019, LNAI 11680, pp. 33–48, 2019.
https://doi.org/10.1007/978-3-030-29249-2_3

items that are most similar to a new user profile, and the similarity and dissimilarity knowledge between the user profile preferences (query) and the item set and the user's experiences (likes or dislikes) [10,16,18,20,28]. This is related to the view of CBR systems as self explainable systems because having cases as precedents of similar problem solving experiences are by themselves useful pieces of knowledge to explain the system outputs [15]. Most of the approaches using explanatory cases do not explain how the system has reached its solution. However, the usefulness of explanatory cases as a support or justification of the results has been demonstrated [24]. Note also that many authors [9,22] agree that displaying the best case is not always sufficient explanation. That is especially true when the goal of the explanation is to provide transparency and when the solution is not a simple reuse of a similar experience but it emerges from complex retrieval and adaptation processes. In this situation specific explanation knowledge, apart from the cases, is required [2,17,22], and explanations may make explicit details about the features that the query has in common with the retrieved cases, why some similarities and differences are more relevant than others for the solution and why the system performs a certain adaptation. When recommender systems use the *collaborative filtering* approach, the system does not use an explicit user profile because the recommendation algorithm relies only on user ratings. Collaborative filtering identifies new user-item associations and predicts users preferences as a linear, weighted combination of other user preferences. Collaborative filtering is more flexible and generally more accurate than content-based techniques [3]. However, it suffers from the self explanatory capability as it lacks from explicit profiles. Other related algorithms like matrix factorization also suffers the same problem.

In this paper, we propose a general approach for explanations in recommender systems using Formal Concept Analysis (FCA). We study how the use of FCA helps in finding the knowledge structure of a recommender system and how this knowledge is useful as the explanation knowledge in the system. Note that the sense of the term explanation here refers to *justification*, as it attempts to make the result of a recommender systems understandable to the user. We propose different approaches that vary in the way we apply FCA and travel the lattice. We propose building the *user profile lattice* with the user personal best rated items. This lattice can be used itself to explain the user profile, and the diversity of her preferences, and let her refine her ratings or understand why a certain item has been recommended. Besides, we also explore how the dependencies between attributes and the maximal groups of items are useful as explanation knowledge in different ways: *item-style*, *property-style* and *dependency style*. We first review the related work in Sect. 2. Section 3 introduces the basics of FCA. Section 4 describes the explanation algorithm and Sect. 5 evaluates the structural properties of profile lattices. Section 6 concludes the paper and outlines the future work.

2 Related Work

Case based explanations have become an active area of research in the last few years [21]. Even if most of the approaches use explanatory cases that do not explain how the AI system has reached its solution (i.e. transparency), the usefulness of explanatory cases as a support or justification of the results of a twin black-box AI system has been demonstrated [24]. Applying explanations in recommendation systems is an important area of research in this type of systems. The main problem with recommendation systems is that users do not know why a product has been recommended to them. Recommender systems that use explanations improve user confidence in those recommendations [25]. In addition, users consume more products that are the result of the recommendations that are explained to them [12].

In previous work [8] we have proposed the use of FCA to help knowledge acquisition and refinement and to help the CBR processes. We studied how the use of FCA can support the task of discovering knowledge embedded in a case base. FCA application provides an internal sight of the case base conceptual structure and allows finding regularity patterns among the cases. Moreover, FCA lattice supports classification based retrieval processes and extracts dependence rules between the attributes describing the cases, that is useful to guide the query formulation process. Given a query, the concept lattice allows accessing all the cases that share properties with the query at the same time so that they are grouped under the same concept. In [7] we used FCA to elicit knowledge from the case based including dependencies between attributes. In the proposed general explanation framework we also explore these dependencies and the maximal groups of items are useful as explanation knowledge. Since its origin in early 80s [27] FCA has became a popular human-centred tool for knowledge representation, data analysis and knowledge discovery with numerous applications. Ontology engineering and big data and their analysis attracted the attention of some researchers using FCA to find the pattern structure and its visualisation [23]. We are not aware of any work using FCA to generate explanations.

3 Formal Concept Analysis

FCA is a mathematical approach to data analysis based on the lattice theory of Birkhoff [5]. It provides a way to identify maximal groupings of objects with shared properties, and enables formalisation of concepts as basic units of human thinking and analysing data in the object-attribute form. This is a clear characteristic of recommender systems where there are items described by properties. Even for collaborative filtering approaches based on collecting ratings there are object-attribute knowledge about the items.

FCA application provides with a conceptual hierarchy, because it extracts the *formal concepts* and the hierarchical relations among them, where related items are clustered according to their shared properties. The lower in the graph, the more characteristics can be said about the items; i.e. the more general concepts are higher up than the more specific ones. In this paper, we propose using

Table 1. Sample set G with 6 movies

Movie Id.	Movie title	Director	Genre	Actors	Year
223	Clerks	Kevin Smith	Comedy	Jason Mewes Jeff Anderson	1994
231	Dumb & Dumber	Peter Farrelly	Comedy	Lauren Holly Teri Garr	1994
235	Ed Wood	Tim Burton	Biography, Comedy Drama	Johnny Deep Martin Landau	1994
110	Braveheart	Mel Gibson	Biography, Drama, History, War	Mhairi Calvey James Robinson	1995
151	Rob Roy	Michael Caton-Jones	Adventure, Biography	Liam Neeson Eric Stoltz	1995
1	Toy Story	John Lasseter	Adventure, Animation, Comedy, Family, Fantasy	Tom Hanks Jim Varney	1995

this conceptual structure as the knowledge base of an explanation framework for recommender systems. In the proposed general explanation framework we also explore dependencies between attributes and metrics on the lattices as explanation knowledge (see Sect. 4.2).

We first briefly review the basics of the FCA technique. See [8] for a description of our previous work on FCA to elicit knowledge from CBR systems. We refer the interested reader to [11,23] for a complete description on FCA and its applications.

A *formal context* is defined as a triple $\langle G, M, I\rangle$ where there are two sets G (of objects) and M (of attributes), and a binary (incidence) relation $I \subseteq G\text{x}M$, expressing which attributes describe each object (or which objects are described using an attribute), i.e., $(g, m) \in I$ if the object g carries the attribute m, or m is a descriptor of the object g. With a general perspective, a concept represents a group of objects and is described by using *attributes* (its intent) and *objects* (its extent). The extent covers all objects belonging to the concept while the intent comprises all attributes (properties) shared by all those objects. With $A \subseteq G$ and $B \subseteq M$ the following operator (*prime*) is defined as:

$$A\prime = \{m \in M | (\forall g \in A)(g, m) \in I\} \qquad B\prime = \{g \in G | (\forall m \in B)(g, m) \in I\}$$

A pair (A,B) where $A \subseteq G$ and $B \subseteq M$, is said to be a *formal concept* of the context $\langle G, M, I\rangle$ if $A' = B$ and $B' = A$. A and B are called the *extent* and the *intent* of the concept, respectively.

It can also be observed that, for a concept (A, B), $A'' = A$ and $B'' = B$, which means that *all objects of the extent of a formal concept, have all the attributes of the intent of the concept, and that there is no other object in the set G having all the attributes of (the intent of) the concept.*

	DIRECTOR						GENRE									ACTORS												YEAR	
id	John Lasseter	Kevin Smith	Mel Gibson	Michael Caton-Jones	Peter Farrelly	Tim Burton	Adventure	Animation	Biography	Comedy	Drama	Family	Fantasy	History	War	Eric Stoltz	James Robinson	Jason Mewes	Jeff Anderson	Jim Varney	Johnny Depp	Lauren Holly	Liam Neeson	Martin Landau	Mhairi Calvey	Teri Garr	Tom Hanks	1994	1995
223		X								X								X	X									X	
231					X					X												X				X		X	
235						X			X	X	X										X			X				X	
110			X						X		X			X	X		X								X				X
151				X			X		X							X							X						X
1	X						X	X		X		X	X							X							X		X

Fig. 1. Example of applying FCA: cross table

Example. We illustrate how to apply FCA to a set G of objects containing 6 movies. The set is described in Table 1, where the selected attributes are the columns in the table. The binary (incidence) relation $I \subseteq G\text{x}M$ is represented by the cross table in Fig. 1. Figure 2 shows the Hasse diagram of the concept lattice resulting of the FCA application[1]. Each node represents a *formal concept* of the context, and the ascending paths of line segments represent the subconcept-superconcept relationship. The lattice contains exactly the same information that the cross table (Fig. 1), so the incidence relation I can always be reconstructed from the lattice.

In Fig. 2 the attributes from the intent are inside grey box labels and the objects from the extent are inside white box labels. A lattice node is labelled with the attribute $m \in M$ if it is the upper node having m in its intent; and a lattice node is labelled with the object $g \in G$ if it is the lower node having g in its extent. Using this reduced labelling, each label (attribute or object name) is used exactly once in the diagram. If a node C is labelled by the attribute m and the object g then all the concepts more general than C (above C in the graph) have the object g in their extents, and all the concepts more specific than C (below C in the graph) have the attribute m in their intents. This way, the intent of a concept in a Hasse diagram in Fig. 2 can be obtained as the union of the attributes in its grey-boxed label and attributes in the grey-boxed labels of the concepts above it in the lattice. Conversely, the extent of a concept is

[1] *Conexp tool* (https://sourceforge.net/projects/conexp/).

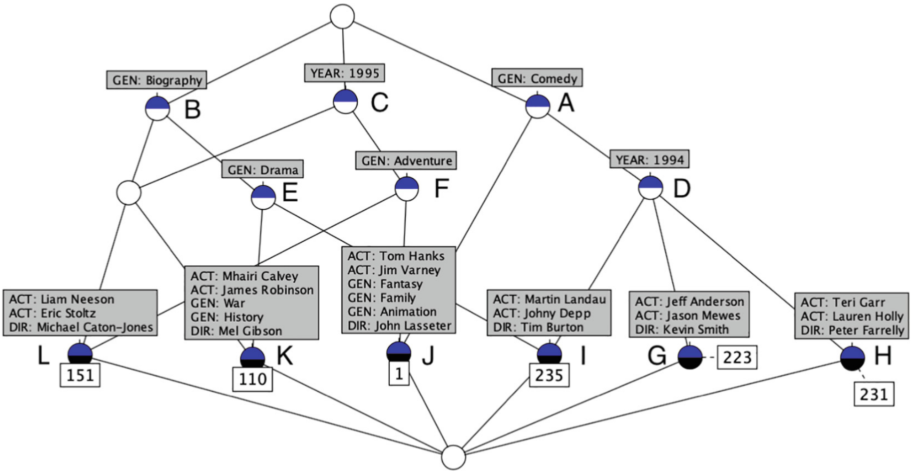

Fig. 2. Example of applying FCA: Hasse diagram of the Lattice.

Table 2. Dependency rules examples extracted from lattice in Fig. 2

GEN: Drama → *GEN: Biography*
GEN: Adventure → *YEAR: 1995*
YEAR: 1994 → *GEN: Comedy*
ACT: Liam Neeson, *ACT: Eric Stoltz*, *DIR: Michael CatonJones* → *GEN: Biography*, *YEAR: 1995*
ACT: Mhairi Calvey, *ACT: James Robinson*, *GEN: War*, *GEN: History*, *DIR: Mel Gibson* → *GEN: Drama*, *YEAR: 1995*
ACT: Martin Landau, *ACT: Johny Depp*, *DIR: Tim Burton* → *GEN: Drama*, *YEAR: 1994*

obtained as the union of the objects in its white-boxed label and objects in the white-boxed labels of the concepts below it in the lattice.

Besides the hierarchical conceptual clustering of the objects, FCA captures knowledge about the co-appearance or associations between attributes. A *dependence rule* [8] between two attribute sets (written $M1 \rightarrow M2$, where $M1, M2 \subseteq M$) means that any object having all attributes in $M1$ has also all attributes in $M2$. We can read the dependence rules in the graph as follows:

– Each line between nodes labelled with attributes means a dependence rule between the attributes from the lower node to the upper one.
– When there are several attributes in the same label it means that there is a co-occurrence of all these attributes for all the objects in the sample and we can infer co-dependence rules.

For example, in the lattice of Fig. 2 we find a dependence between nodes E and B meaning that, in the set of items used to build the lattice, all the Drama movies are also Biographies (*GEN: Drama* → *GEN: Biography*). Table 2 shows some of the dependency rules.

The operation $classify(i)$ returns the formal concept C in the lattice that recognises *all* the attributes that describe i. For example, given a movie m

with known attributes [*GEN: Comedy*, *ACT: Tom Hanks*], m is classified in the concept A because it is the lower concept where all the given attributes are fulfilled. The concept J does not recognise the individual as an instance because, even if it has the attribute [*ACT: Tom Hanks*], there are other required attributes [*GEN: Fantasy*, *GEN: Family*, *GEN: Animation*,..] that m should fulfil.

4 FCA-Based Explanation Algorithm

A collaborative filtering approach recommends items based on users' past behaviour and ratings. However, it lacks from an explicit aggregated model of the user preferences and the capability of explaining their results. We propose building the *user profile lattice* with the user personal best rated items. This lattice can be used as a model to explain the user profile and the diversity of her preferences. This explanation allows the self-comprehension of the user profile based on her ratings (Sect. 4.1) or understanding why a certain movie has been recommended (Sect. 4.2). The explanation lattice is computed for each user and it is reused for different recommendations.

Our approach to generate explanations is general and applicable to different recommendation domains. It uses a set of the user ratings and the item properties to generate the FCA lattice based on the best rated items, and generates personalised explanations for each particular user profile. The *FCA-based explanation algorithm* (see Algorithm 1) allows us to organise the knowledge on the user preferences and obtain the vocabulary to explain the user profile and, according to this profile, why an item has been recommended using either the *item-style* explanation, which includes the similar items rated by the user; the *property-style* explanation, which describes the properties from the formal concepts; and the *dependency-style* explanation, that includes the description of the association rules elicited by the FCA. The general process runs as follow:

Step 1 Selection of the M_u (attributes) and G_u (items) sets used to build the lattice (details of selection strategies sel_g and sel_m in Sect. 5.2).
Step 2 Apply FCA and evaluate and refine the resulting lattice.
Step 3 Choose between explaining the user profile (details in Sect. 4.1) or explain a specific recommendation (rec), so we classify rec to generate more specific explanations (details in Sect. 4.2).
Step 4 Explanations are generated from textual templates filled with the corresponding elements obtained while travelling the lattice *item-style, property-style, dependency-style*.

The different styles of explanations can be combined. In the *property-style* explanation we show the intent of the formal concepts that explains the properties that make these items be grouped together. In the *item-style* explanation we show the extent of the formal concepts. It shows items that are somehow similar according to the maximal groups. Association (or dependency) rules are very interesting pieces of knowledge, difficult to see at first sight from the items, and very useful for explanations.

Algorithm 1. Travelling the lattice to build Explanations

Input: G_u, M_u, I_u, rec, sel_g, sel_m
Output: *Expl-item, Expl-property, Expl-dependency*

$G_u{}' = sel_g(G_u)$
$M_u{}' = sel_m(M_u)$
$Ret = FCA(G_u{}', M_u{}', I_u)$
$C_r \leftarrow Ret.classify(rec) \parallel TOP$
$Expl\text{-}item \leftarrow \{traverseLevels(C_r.extent)\ \}$
$Expl\text{-}property \leftarrow \{traverseLevels(C_r.intent)\ )\ \}$
$Expl\text{-}dependency \leftarrow \{\text{obtainRules(Ret)}\ \}$

Table 3. Example of selected best rated items for user u

Movie Id.	Movie title	Director	Genre	Actors	Year	Rating
594	Snow White and the Seven Dwarfs	William Cottrell	Animation, Family, Fantasy, Musical	Adriana Caselotti Lucille La Verne	1937	5.0
596	Pinocchio	Norman Ferguson	Animation, Family, Fantasy, Musical	Mel Blanc Cliff Edwards	1940	4.5
588	Aladdin	Ron Clements	Adventure, Animation, Comedy, Family, Fantasy, Musical, Romance	Robin Williams Scott Weinger	1992	5.0
364	The Lion King	Roger Allers	Adventure, Animation, Drama, Family, Musical	Matthew Broderick Niketa Calame	1994	5.0
317	The Santa Clause	John Pasquin	Comedy, Drama, Family, Fantasy	Judge Reinhold Peter Boyle	1994	3.5
34	Babe	Chris Noonan	Comedy, Drama, Family	Miriam Margolyes Roscoe Lee Browne	1995	4.0
158	Casper	Brad Silberling	Comedy, Family, Fantasy	Eric Idle Cathy Moriarty	1995	3.0
48	Pocahontas	Mike Gabriel	Adventure, Animation, Drama, Family, History, Musical, Romance	Christian BaleIrene Bedard	1995	5.0

We have generated a user profile using the items in Table 3. The corresponding lattice is shown in Fig. 3 and it will be used as running example in the following subsections.

4.1 Explanation of the User Profile

The user profile and the corresponding explanations are personalised for each user because the item set G'_u used to generate the lattice are selected from the user ratings. In the example, Table 3 represents the selected best rated items and attributes for a certain user G'_u and M'_u. We apply FCA and generate the corresponding lattice (Fig. 3). According to the general Algorithm 1 described in Sect. 4 we propose a property-style explanation that travels the lattice, level by level, from top to bottom. Note that the label [*GEN: Family*, *LAN: English*] in the TOP concept means that all the movies in G'_u have these properties

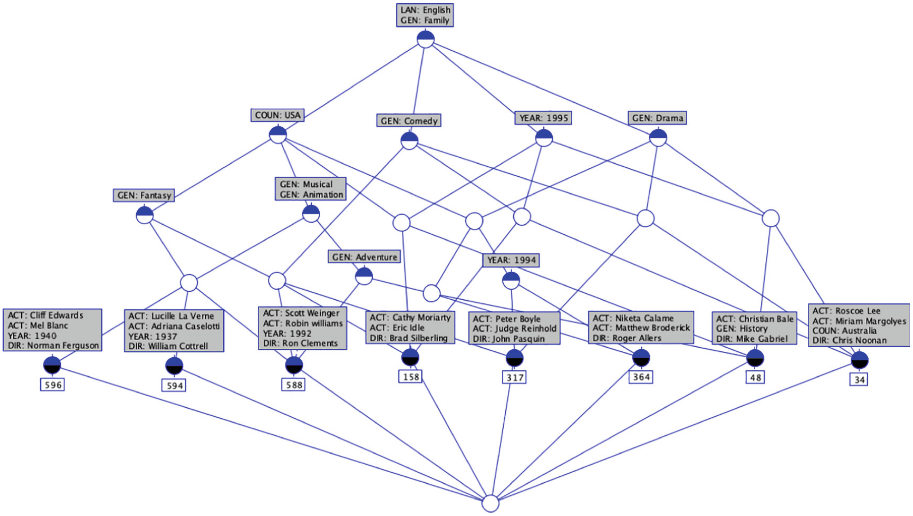

Fig. 3. User profile lattice Ret = FCA(G'_u, M'_u, I_u) with G'_u M'_u and I_u in Table 3

Table 4. Example *Property-style* explanations for the user profile in Fig. 3

Level 0	"*All your high scored movies share GEN: Family LAN: English*"
Level 1	"*Most of your high scored movies are GEN: Drama or GEN: Comedy*"
	"*Would you like to see them?*" (item-style)
	"*Your movies in* GEN: Drama *are:* 590, 364, 317, 34, 48" (Extent of concept)
	"*Your movies in GEN: Comedy are*: 588, 317, 34, 158" (Extent of concept)
	"*Most of your high scored movies are COUN: USA*"
	"*Would you like to see them?*" (item-style)
	"*Most of your high scored movies are YEAR: 1995*"
	"*Would you like to see them?*" (item-style)
Level 2	"*Some of your high scored movies are* GEN: Musical, *GEN: Animation or GEN: Fantasy*"

and, therefore, all the concepts below TOP inherit the property. The *property-style* explanation can be combined with *item-style* explanations by retrieving the extent of a certain concept. In the example, the user chooses when to see the specific items. The textual explanation generated for this user profile will employ different templates for each level in the lattice of Fig. 3, as described in Table 4.

In addition to *property-style* explanations we can also use complementary *dependency-style* explanations, which provide association between attributes in G'_u. Dependency-style explanations are very useful to help the user to understand her profile and refining the scoring if the user disagrees with any of the extracted rules. Note that transitivity could be applied between rules and that some of the explanations are redundant with respect to the *property-style* explanations on the same lattice. Table 5 shows some examples of dependency rules between attributes in the lattice of Fig. 3 and the corresponding textual explanations.

Table 5. *Dependency-style* explanations generated for the user profile in Fig. 3

GEN: Comedy → *GEN: Family*, *LAN: English*
"*All your high scored movies that have GEN: Drama have also GEN: Family and LAN: English*"
GEN: Animation, *GEN: Musical* → *COUN: USA*
"*All your high scored movies that have GEN: Animation and GEN: Musical have also COUN: USA*"
GEN: Adventure → *GEN: Animation*, *GEN: Musical*
"*All your high scored movies that have* GEN: Adventure *have also GEN: Animation and GEN: Musical*"
GEN: Romance → *GEN: Adventure*
"*All your high scored movies that have GEN: Romance have also* GEN: Adventure"
GEN: Drama → *GEN: Family*
"*All your high scored movies that have GEN: Drama have also GEN: Family*"
...

Note that the textual templates for explaining dependency rules fill the gaps using the rule (Left/Right) Hand Sides (L/R)HS): *"All your high scored movies that have"* LHS *"have also"* RHS.

4.2 Explaining a Recommendation

Besides explaining user profiles, the FCA lattice can be used to explain a recommendation, i.e, why a particular item has been recommended over others. Step 3 in Algorithm 1 classifies an item *rec* in the lattice by its properties. Then, we build an *item-style* or *property-style* explanation using the concept C_r that recognises the object *rec*.

As we described in Sect. 3, *rec* is classified in the lattice –$classify(rec)$– if any of lattice concepts recognises all *rec* properties. In [7] we proposed a classification based retrieval method, where a partially defined query is classified in the FCA lattice that organise the case base. We proposed a query completion process based and the use of dependency rules that helps to complete the query towards similar cases. We cannot apply this method here because *rec* is not a partially defined query, but a complete individual with all its properties. Note that the classification process fails if none of the formal concepts recognises all the properties in *rec*.

An easy approach would be rebuilding the FCA lattice using $\{G_u \cup r\}$ and generating the explanations using Algorithm 1. Depending on the size of the formal context and the optimisation of the FCA implementation that could become inefficient. As an alternative, we propose using each property separately, completing with dependency rules when possible and generating partial explanations. For example, if the recommender systems recommends item 223, whose properties are [*GEN: Comedy*, *YEAR: 1995*, (...)], we can generate the explanations detailed in Table 6 using the properties one by one.

5 Evaluation

In order to evaluate the possibilities of the FCA lattice to provide explanations we have analysed several descriptive metrics. These metrics let us conclude the

Table 6. Explanations of a recommendation based on properties of the proposed item.

"*This movie is recommended to you because of the GEN: Comedy property*" (property-style)
"*The recommended movie shares the property GEN: Comedy with* 588, 317, 34, 158" (Extent of concept) (item-style)
"*The recommended movie shares the property YEAR: 1995 with* 34, 48, 158" (Extent of concept) (item-style)

feasibility of the method for a real dataset and discuss the potential quality of the explanations according to them. Concretely, we have combined two popular datasets. The first of them is the MovieLens dataset [14]. This dataset contains 100,000 ratings made by users in the MovieLens recommendation system. The second dataset contains the features of 5000 movies extracted from IMDB[2]. The features of the movies used in the evaluation are: genres, directors, actors, screenwriters and the decade in which they were released. The metrics defined to analyse the lattice are:

Num. Nodes (N) represents the number of nodes in the lattice. It includes *top* and *bottom* nodes. We can measure the number of nodes with respect to the number of items used to generate the lattice. It allows us to measure the proportion between the nodes in the lattice and the number of items used to generate it.

Level Width (LW) is the highest number of nodes in a lattice level. It represents the maximum width of the lattice.

Depth (D) measures the length of the longest path from *bottom* to *top* of the lattice. Width and depth allow to study the distribution and the diversity/homogeneity of the attributes in the items of the user's profile.

Branch Factor (BF) measures the average number of children for the nodes in the lattice.

First, we have analysed the global behaviour of the lattices when varying the number of movies used to create them. Later, we have studied the behaviour of the lattices when using two different approaches for selecting the movies. Next, we discuss both evaluations and their impact in the quality of the explanations.

5.1 Global Behaviour of the FCA Lattices

This first evaluation aims to explore the features of the lattices with respect to the number of movies used to create them. This way we can figure out the optimal number of movies required to create the lattice that provides the explanations and analyse the global behaviour of lattices regarding diversity, homogeneity and complexity. To perform this evaluation we have chosen randomly 100 users and generated their lattices with a fixed number of movies (from 2 to 75). These movies are the best rated by the user. Then for each number of movies we average the results of the 100 users.

[2] https://www.imdb.com/.

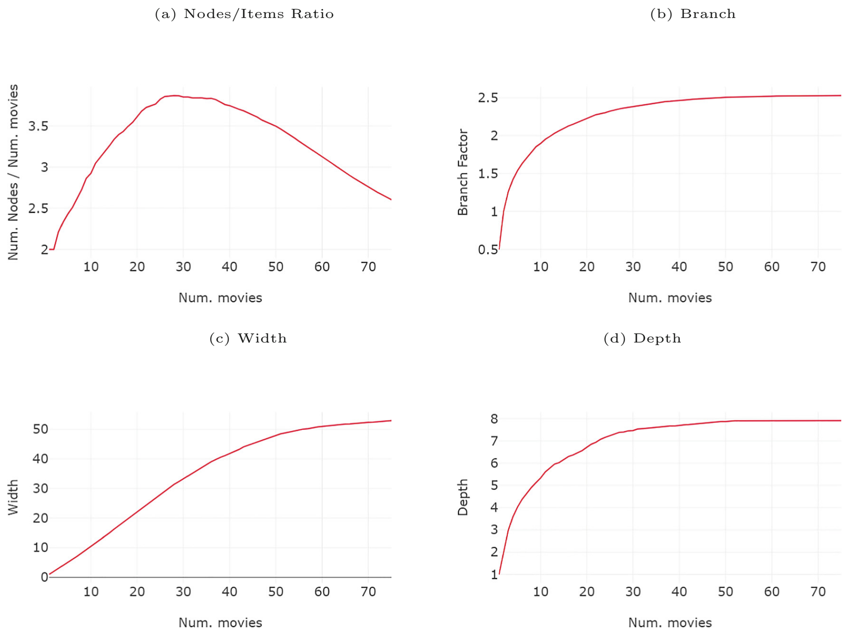

Fig. 4. Lattice properties analysis w.r.t. the number of movies $|G'_u|$ (x-axis)

Figure 4 summarises our findings. The first chart (Fig. 4(a)) shows the relationship between the number of nodes of the lattices and the items used to create them. As expected, the lattice grows as the number of items increases. However, it saturates around $|G'_u| \sim 20$ and starts decreasing. This way, we can conclude that it is the optimal size to compute the user's lattice.

Then, Figs. 4(c) and (d) describe the behaviour of the width and depth metrics. The decrease of the lattice size as the number of items is greater than approximately 20, makes this ratio to slightly slow down after that value in the case of the width metric. The depth metric stabilises completely for large lattices. As this metric is associated to homogeneity of the user preferences, results clearly denote that larger profiles are more heterogeneous because they became wider but depth value is stabilised. If we analyse the width/depth ratio we can clearly obtain that conclusion. For example, lattices built with $|G'_u| = 20$ items are on average 3 times wider.

Finally, Fig. 4(b) shows the branch factor. Here we can a observe a behaviour similar to the depth ratio. Increasing the number of items does not imply a linear raise of the complexity of the lattices, i.e., the number of intermediate nodes representing shared properties does not raise proportionally.

Once we have analysed the global behaviour of the lattices we can evaluate the item selection strategies used to create them.

Table 7. Averaged results of the selection strategies.

Metric	sel_g^{best}	sel_g^{20}
Num. Nodes/Num. Movies	3.711	3.611
Depth/Num. Movies	0.270	0.336
Width/Num. Movies	1.078	1.101
Branch factor/Num. Movies	0.087	0.111

5.2 Item Selection Strategies

Before applying FCA we select the item's attributes and the subset of movies from the user's profile that will be analysed. It is the first step of the general process outlined in Sect. 4 and represented as the sel_m and sel_g functions in Algorithm 1.

The selection of the item's attributes (sel_m) is a simple process that depends on the domain. In our example we can discard the *id* and *title* attributes as they cannot be used to classify the movies in the lattice.

However, there are several alternatives for selecting the movies used to compute the FCA lattice (sel_g strategy). As we have concluded from the previous evaluation, 20 is approximately the optimal number of movies to generate the FCA lattice. Therefore, we could select the 20 movies with the maximum score, select them randomly, use a stratified selection according to the rating values, etc. On the other hand, we could ignore this fixed number of items and implement an alternative with all the items with a high score. In order to evaluate the impact of this selection method we have computed the previous metrics for every user but selecting items according to the following strategies:

$\boldsymbol{sel_g^{best}}$: Select all the items with score >= 4 stars.
$\boldsymbol{sel_g^{20}}$: Select the 20 items with the maximum scores.

From sel_g^{best} and sel_g^{20}, we are able to compute two lattices, respectively, for each user. Our goal to compare the properties of both lattices using the previous metrics. However, we cannot compare them directly as the number of items used to generate the lattices $((|G'_u|))$ is different. Therefore, we will normalise every metric (nodes, wide, depth and branch factor) according to the number of movies. Again, we have randomly chosen 100 users that have watched more than 20 movies. The mean number of movies rated by these 100 users is 40.20.

To summarise the achieved results, we have averaged every metric for the whole set of users. Results are shown in Table 7. Analysing this table we can conclude that both approaches got very similar results. To analyse the non-aggregated results we have generated the graphs in Fig. 5. Here, we compare the difference obtained by every metric, once has been normalised, between the pair of lattices generated for each user. Therefore, every column represents $sel_g^{20} - sel_g^{best}$. Columns have been ordered according to the difference value to be able to compare which is the the winning strategy. Beginning with the number of

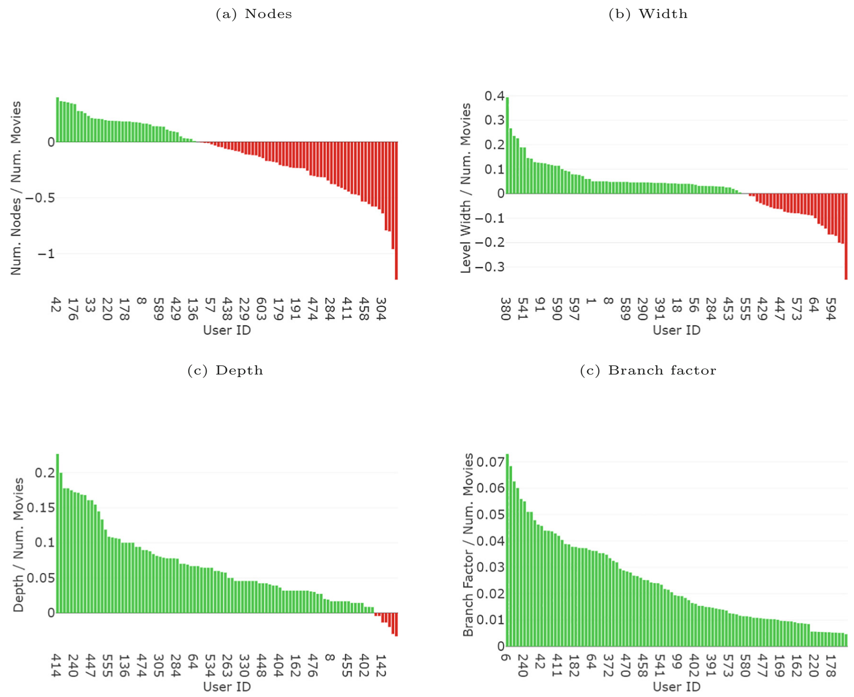

Fig. 5. Differences in the evaluation of every user using both selection strategies ($sel_g^{20} - sel_g^{best}$). Every metric has been previously normalised w.r.t. the number of movies.

nodes shown in Fig. 5(a), it is slightly balanced to the sel_g^{best} method (there are more negative values). It means that this selection strategy provides items that generate larger lattices. Regarding the width of the lattices, Fig. 5(b), it is quite balanced. However, Fig. 5(c) shows that the sel_g^{20} method obtains deeper lattices, associated to homogeneous profiles. It makes sense as this strategies retrieves the best rated items for the user, and they are usually quite similar. Finally, we cannot conclude any significant result regarding the branch factor because the magnitude of the differences is too low.

6 Conclusions and Future Work

In this paper, we have proposed the use of Formal Concept Analysis to help finding the knowledge structure of the user profiles in recommender systems and how to use this knowledge as explanation knowledge for the system. We have proposed a FCA-based explanation approach to organise the knowledge about users' preferences in a lattice that helps to obtain the vocabulary to explain why an item is recommended using other either similar items (*item-style explanations*) that the user has rated or similar properties (*property-style explanations*) with the best rated items for that user. Additionally, the *user profile lattice* itself can be employed as a form of scrutability of the recommender system, where the user understands which profile is inferred by the system according to her ratings.

We have analysed a set of user profile lattices according to their structural properties and the next step is to evaluate the explanations with real users. Besides, the study of lattice metrics and properties opens a line of future work where we plan to study further the different types of user profiles in terms of their rating behaviours, both in quantity and distribution of the rating values. Our hypothesis is that the number of ratings and the distribution of the rating values would affect the structural properties of the user profile lattice. For this reason we should test with different methods to refine the way the set of items (G'_u) employed to generate the user profile are chosen. Similarly, we would like to test different ways to select the set of properties (M'_u) that will be used to generate the lattice and, therefore, the explanations.

References

1. Aamodt, A.: Explanation-driven case-based reasoning. In: Wess, S., Althoff, K.-D., Richter, M.M. (eds.) EWCBR 1993. LNCS, vol. 837, pp. 274–288. Springer, Heidelberg (1994). https://doi.org/10.1007/3-540-58330-0_93
2. Armengol, E., Plaza, E.: Using symbolic descriptions to explain similarity on CBR. In: Proceedings of the 8th International Conference of the ACIA, Alguer, Italy, pp. 239–246 (2005)
3. Bell, R., Koren, Y., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(08), 30–37 (2009)
4. Biran, O., Cotton, C.V.: Explanation and justification in machine learning: a survey. In: IJCAI 2017 Workshop on Explainable AI (XAI) (2017)
5. Birkhoff, G.: Lattice Theory, vol. 25. American Mathematical Society, New York (1940)
6. Caro-Martinez, M., Jimenez-Diaz, G., Recio-Garcia, J.A.: A theoretical model of explanations in recommender systems. In: ICCBR 2018, p. 52 (2018)
7. Díaz-Agudo, B., González-Calero, P.A.: Classification based retrieval using formal concept analysis. In: Aha, D.W., Watson, I. (eds.) ICCBR 2001. LNCS (LNAI), vol. 2080, pp. 173–188. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44593-5_13
8. Díaz-Agudo, B., González-Calero, P.A.: Formal concept analysis as a support technique for CBR. Knowl.-Based Syst. **14**(3–4), 163–171 (2001)
9. Doyle, D., Cunningham, P., Bridge, D.G., Rahman, Y.: Explanation oriented retrieval. In: Funk, P., González Calero, P.A. (eds.) ECCBR 2004. LNCS (LNAI), vol. 3155, pp. 157–168. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-28631-8_13
10. Doyle, D., Tsymbal, A., Cunningham, P.: A review of explanation and explanation in case-based reasoning (2019)
11. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, Heidelberg (2012)
12. Gedikli, F., Jannach, D., Ge, M.: How should I explain? A comparison of different explanation types for recommender systems. Int. J. Hum Comput Stud. **72**(4), 367–382 (2014)

13. Gunning, D.: Darpa's explainable artificial intelligence (XAI) program. In: 24th IUI, Marina del Ray, CA, USA (2019)
14. Harper, F.M., Konstan, J.A.: The movielens datasets: History and context. ACM Trans. Interact. Intell. Syst. **5**(4), 19:1–19:19 (2015)
15. Leake, D.B., McSherry, D.: Introduction to the special issue on explanation in case-based reasoning. Artif. Intell. Rev. **24**(2), 103–108 (2005)
16. Lee, O., Jung, J.J.: Explainable movie recommendation systems by using story-based similarity. In: 23rd ACM IUI, Tokyo, Japan (2018)
17. Maximini, R., Freßmann, A., Schaaf, M.: Explanation service for complex CBR applications. In: Funk, P., González Calero, P.A. (eds.) ECCBR 2004. LNCS (LNAI), vol. 3155, pp. 302–316. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-28631-8_23
18. McSherry, D.: Explanation in recommender systems. Artif. Intell. Rev. **24**(2), 179–197 (2005)
19. Muhammad, K., Lawlor, A., Smyth, B.: A multi-domain analysis of explanation-based recommendation using user-generated reviews. In: 31st FLAIRS, Melbourne, USA, pp. 474–477 (2018)
20. Pu, P., Chen, L.: Trust-inspiring explanation interfaces for recommender systems. Knowl.-Based Syst. **20**(6), 542–556 (2007)
21. Schank, R.C., Kass, A., Riesbeck, C.K.: Inside Case-Based Explanation. Psychology Press (2014)
22. Roth-Berghofer, T.R.: Explanations and case-based reasoning: foundational issues. In: Funk, P., González Calero, P.A. (eds.) ECCBR 2004. LNCS (LNAI), vol. 3155, pp. 389–403. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-28631-8_29
23. Singh, P.K., Aswani Kumar, C., Gani, A.: A comprehensive survey on formal concept analysis, its research trends and applications. Int. J. Appl. Math. Comput. Sci. **26**(2), 495–516 (2016)
24. Sørmo, F., Cassens, J., Aamodt, A.: Explanation in case-based reasoning-perspectives and goals. Artif. Intell. Rev. **24**(2), 109–143 (2005)
25. Tintarev, N., Masthoff, J.: Explaining recommendations: design and evaluation. In: Ricci, F., Rokach, L., Shapira, B. (eds.) Recommender Systems Handbook, pp. 353–382. Springer, Boston, MA (2015). https://doi.org/10.1007/978-1-4899-7637-6_10
26. Weber, R.O., Johs, A.J., Li, J., Huang, K.: Investigating textual case-based XAI. In: Cox, M.T., Funk, P., Begum, S. (eds.) ICCBR 2018. LNCS (LNAI), vol. 11156, pp. 431–447. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01081-2_29
27. Wille, R.: Restructuring lattice theory: an approach based on hierarchies of concepts. In: Rival, I. (ed.) Ordered Sets, pp. 445–470. Springer, Dordrecht (1982). https://doi.org/10.1007/978-94-009-7798-3_15
28. Zanker, M., Ninaus, D.: Knowledgeable explanations for recommender systems, vol. 1, pp. 657–660 (2010)