Time, Place and Environment: Can Conceptual Modelling improve Context-Aware Recommendation?

Ángel Castellanos
ETSI Informática UNED Madrid, Spain
acastellanos@lsi.uned.es

Ernesto William De Luca
Potsdam Univ. of Applied Sciences
Potsdam, Germany
deluca@fh-potsdam.de

Juan Cigarrán
ETSI Informática UNED Madrid, Spain
juanci@lsi.uned.es

Ana García-Serrano
ETSI Informática UNED Madrid, Spain
agarcia@lsi.uned.es

ABSTRACT

Recommendation refers to the automatic process of discovering and suggesting new but relevant items to users, according to the preferences inferred from their previous activity. But, not only the items or their content are related to the user preferences but also the context in which the user have consumed the items. In this regard, the so-called context-aware recommenders refer to the systems integrating such contextual information in the recommendation process. However, sometimes it is not clear what context information is the best in order to improve the recommendation process or how it should be included. In this sense, we present a novel approach for context-aware recommendation based on a conceptual modelling for the user-item-context matrix. The modelling is conducted by means of the application of Formal Concept Analysis (FCA) to infer and organize the user preferences derived from their previous contextualized activity. We have experimented with different ways to integrate the contextual information by splitting it in the different dimensions that it addresses (e.g. location, time, company…). To that end, we have proposed different recommendation algorithms (content-based, collaborative filtering and hybrid recommenders) and also different ways to manage the user's ratings. The obtained results prove the suitability of FCA for context-aware recommendation, outperforming other state-of-the-art proposals.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Algorithms, Experimentation, Theory.

Keywords

Formal Concept Analysis, Context-Aware Recommendation, Recommender Systems.

1. INTRODUCTION

The main goal of Recommender systems (RS) is to infer user preferences in order to suggest new contents satisfying these preferences. However, in this context appears the problem of user's "situated action" [21]: a content relevant for a user in a given context, might be irrelevant in a different one. Context-Aware RS try to cope with this problem by integrating contextual information (e.g. location, time, mood…) as a new dimension in the recommendation process. Traditional recommendation environments propose a two-dimension space (User-Items), but the context-aware recommenders operate in a three-dimension space (User-Items-Context). User ratings are therefore a function of the interest for an item in a given context.

In order to model this three-dimensional environment, we propose a recommender system based on Formal Concept Analysis (FCA). FCA is a theory with an already demonstrated performance for content modelling and organization. By applying FCA, it is possible to organize the information in the recommendation environment by grouping together similar users or contents and organizing these groups in a hierarchical structure, based on the subjacent relationships between them. Such organization appears as a valuable help in the recommendation process. Some previous works have intended to exploit this organization performance both for collaborative filtering [14, 27] and content-based recommendation [7, 12, 20]. Nevertheless, these previous proposals barely delve in the advantages offered by FCA for the recommendation process, being mostly focused on testing the viability of the FCA application.

In this regard, we do believe FCA may be very useful to integrate contextual information in the recommendation process, identifying latent relationships or patterns based on this contextual information. To our view, contextual information covers different dimension and it cannot be considered as a single source. For instance, the location of a user (work or home) might have much more influence in the user preferences than the weather. Thus, to have a best insight of this point, we propose an experimental work for studying the influence of different types of contextual information in different recommendation scenarios (content-based, collaborative-filtering-based and hybrid-based).

In the following, we review the related work in the section 2; we present our recommendation framework including a brief description of FCA in the section 3; we describe our experiments and the obtained results in the section 4; and, finally, we expose our conclusions based on the obtained results and we propose the main future lines derived from this work.

2. RELATED WORK

2.1 FCA Recommendation

The FCA application in recommender systems is based on the organization of a set of users according to a set of items, either by the items themselves (collaborative filtering approaches) or by their content (content-based approaches). Mathematically expressed, a recommender system can be interpreted as a bipartite graph partitioned into users (U) and items, or their content, (I). The edges in this graph, $\rho = r(u,i)$, establish the interest of the user u by and item i weighted with a rating r. Following the FCA theory (see more in the section 3.2), the triple (U, I, ρ) can be seen as a FCA *formal context*. In this scenario, FCA provides a powerful modelling technique, creating hierarchical representation based on the latent structure of the information reflected in the *formal context*. This representation allows the inferring of valuable relationships between users and items for the recommendation process.

Trying to take advantage of this modelling performance, in [27] the authors apply the FCA basis to obtain subsets of users sharing the same purchases. Then, recommendation is done by calculating the entropy of each subset in order to find the most suitable ones for a specific item. In [24] the authors also propose a collaborative filtering approach that intends to take advantage of the FCA-based organization to find user similarities according to the items they have interacted with. For that purpose, two methods based on the entry level concept are proposed: one based on the entry level of an attribute and another one on the entry level of a user. Other example of collaborative filtering approach based on FCA is proposed in [26]. The particularity of this approach is the application of FCA on fuzzy data (i.e., instead using binary values, values are continuous in a [0-1] interval). To perform the recommendation process the authors propose a basic collaborative filtering algorithm, recommending the items already seen by the users that share some item with the target user.

But not only collaborative filtering recommendation has been addressed by means of FCA, it has been also applied to content based recommendation. If a set of items is grouped (in clusters, classes, concepts) according to their shared content (e.g. features, text, keywords), it will be helpful to the computation of the recommendation algorithm. Following this idea, in [13] FCA is applied in a crowdsourcing platform to represent users according to the content (mainly keywords) of the projects in which they are interested. Their proposal takes into non binary attributes by using multi-valued formal concepts. Other content-based example is presented in [19]. In this work FCA is used to model item profiles by using their metadata. The resultant lattice is used to infer relations between user and item FCA-based descriptions, recommending the items related to these descriptions. Finally, the work in [20] proposes a FCA-based recommendation approach in an e-learning environment. More concretely, the authors apply Fuzzy Formal Concept Analysis (FFCA) to model contents in RSS-feeds in a Fuzzy Lattice. In this scenario, given the *learning context* of a target user and the aforementioned Fuzzy Lattice, the most similar concepts in the Fuzzy Lattice (according to Wu and Palmer similarity) are recommended to the user. Other interesting approaches of FCA-based recommendation in e-learning scenarios are detailed in [10] and in [18].

In this work, we address both methodologies (content-based and collaborative filtering) as well as their hybridization. We have proposed three recommendation algorithms to test the performance of contextual information in the different recommendation scenarios.

2.2 Context-aware Recommendation

In the beginning the RS were based on simple user models, only including raw information about the users. However, this kind of models do not have the ability to capture the knowledge associated to the information about the users [2]. To cope with this problem, a common approach has been the use of the available context information to create context-dependent sub-profiles. This process may be done either by using context information as a pre-filter, as post-filter or directly in the modelling process (as we propose in this work) [15]. Pre-filter approaches are mainly based on splitting the user or item set according to the contextual information. Following this methodology the authors of [25] apply time and location data to generate different user sub-profiles in a movie recommendation system. On the other hand, the item splitting approach is addressed in the works presented in [3] or in [5]. Post-filtering approaches apply the contextual information after the recommendation process to adjust the results [8]. More information about the different ways to integrate the contextual data and also a comparative analysis about how to integrate these data in different recommendation tasks can be consulted in [24].

To have a clear idea about the field, some other interesting surveys are: the state of the art review in [1] or the one presented in [2] which includes a discussion of the trends and future lines. From this latter work, some conclusions about contextual information can be drawn: 1) there is no dominant technique in terms of overall performance; 2) the accuracy is worse when the context information has a finer granularity and; 3) the approaches using contextual information in the modelling technique (as the one presented here) are, in general, the best-performing in terms of accuracy. These and other issues are addressed in the survey in [4] that reviews the main context-based user modelling approaches from the initial works in the area, based on keywords, to the most novel techniques like Object Role Modelling (ORM) or ontology based models.

As it was said before, the use of fine-grained context information decreases the accuracy of the systems. However, by including a wider context, user modelling becomes more complex due to new issues have to be taken into account: data heterogeneity or relationships between context information among others. In this regard, the FCA-based modelling is able to deal with this granularity problem. The lattice resultant from the FCA application represents the information in a hierarchical structure, covering all the granularity degrees from the most generic one (in the top of the lattice) to the most specific one (in the bottom of the lattice).

Granularity of the context information is not the only issue to take into account. In general, the identification of the most valuable context information (i.e. the one that most influences the user preferences) is still an open problem. To address this point, in this work we propose an experimental study to test the performance achieved by the different types of contextual information.

3. EXPERIMENTAL FRAMEWORK

The experimental framework is based on the application of FCA for content modelling based on the contextual information and an own-developed recommendation algorithm trying to take advantage of the organization provided by FCA. In the following it is presented the collection used for the experimentation, the FCA basis and its application for content modelling and, finally, the developed recommendation algorithm.

3.1 Test Bed: the LDOSCoMoDa

For the experimentation we have used the LDOS-CoMoDa dataset. It is a context-rich movie dataset, containing real user-item interactions and twelve different types of contextual information related to the interactions [17]. More in detail, the dataset includes about 1600 ratings made by about 90 users over a set of 950 items. The context variables cover 12 different types: time, daytype, season, location, weather, social (information about whether the user watched the movie alone or in company; for instance, Alone, My partner, Friends, Colleagues), endEmo (emotion at the end of the movie), dominantEmo (main user emotion during the movie), mood, physical, decision (i.e the user decided which movie to watch or the user was given a movie) and interaction (first or n-th interaction of the user with the movie). In order to conduct our later experimentation we have classified the contextual information in 6 types: user_information (dominantEmo, endEmo, mood, physical), time (daytype, season, time), social (social), location (location, weather), decision (decision), and all together.

3.2 FCA Basics

Formal Concept Analysis (FCA) is a mathematical theory of concept formation [11, 28] derived from lattice and ordered set theories that provides a theoretical model to organize *formal contexts*. A *formal context* is defined as a set structure $\mathbb{K} := (G, M, I)$, where G is a set of (formal) *objects*, M a set of (formal) *attributes* and I a *binary relation* between G and M, i.e. ($I \subseteq G \times M$), denoted by gIm, which is read as: the object g has the attribute m.

The main construct of the theory is the *formal concept*, a pair (A, B) where $A \subseteq G$ is a set of objects (the *extent* of the formal concept) and $B \subseteq M$ is a set of attributes (the *intent* of the formal concept). To construct the *formal concepts* in a *concept lattice*, two interesting kinds of *formal concepts* are *object concepts* and *attribute concepts*. The *object concept*, denoted as γg, associated with an object g is the most specific concept including g in its extent. Conversely, the *attribute concept*, denoted as γm, associated with the attribute m is the most generic concept including m in its intent.

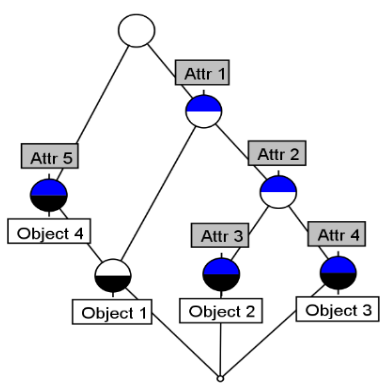

Figure 1- Example of Concept Lattice Representation

Formal concepts can be formally ordered in a *subconcept-superconcept-relation* according to their extents:

$$(A, \mathrm{B}) \leq (A', \mathrm{B}') :\Leftrightarrow (\mathrm{A}, \mathrm{B}) \subseteq (A', \mathrm{B}') \Leftrightarrow \mathrm{A} \subseteq \mathrm{A}'$$

where (A', B') is called a *super-concept* of (A, B) and, conversely, (A, B) is a *sub-concept* of (A', B') (i.e., (A, B) is more specific than (A', B')). The order that results can be proven to be a *lattice*, which is called the *concept lattice*, denoted as $\mathcal{B}\,(G, M, I)$, associated to the *formal context*. Since *concept lattices* are ordered sets, they can be naturally displayed in terms of *Hasse diagrams* [11]. In a *Hasse diagram*: 1) there is exactly one node for each formal concept; 2) if $\mathrm{C} \subseteq \mathrm{C}'$, then C' is placed above C (C is a *sub-concept* of C' or C' is a *super-concept* of C) and, 3) if $\mathrm{C} \subseteq \mathrm{C}'$ but there is no other intermediary concept C' such as $\mathrm{C} \subseteq \mathrm{C}'' \subseteq \mathrm{C}'$, there is a line joining C and C'. In Figure 1 it can be viewed an example of the *concept lattice*:

3.3 Modelling proposal.

As it was explained in the previous section, FCA is useful to organize items according to their attributes. In this work we proposed the application of FCA to model users and items according to their features and the contextual information. More in detail we propose the following FCA-based models to solve the research questions posed in this work: item-based (group together similar items according to the contextual information), user-based (grouping together similar users according to the contextual information) and user-item-based (modelling users according to the consumed items).

3.3.1 Item-based Modelling

This proposal is based on the grouping of similar items according to their contextual features. If the system is able to identify some similarities between the items, it will improve in the recommendation process by offering similar items to the ones already consumed. The *formal context* associated to this modelling includes the items to be modelled as the *objects* and the features related to these items as the *attributes*. After the FCA computation, the resultant lattice will group together similar items (i.e. those sharing the same feature set) and it will hierarchically organize them in the lattice structure.

To find the contextual features related to the items this approach takes the user-item interactions in the dataset. To this intent, the system takes the contextual information relate to each interaction (e.g. this movie has been consumed at home, alone, etc…) and adds to the *formal context* the item as *object* and the contextual information as the *attributes*. It might happen (i.e., in fact it is expected to happen) that the contextual data reflect contradictory information for the same item (e.g., for some interaction the mood of a user may be positive, while for other interaction with the same item the mood of another user may be negative). This problem is addressed by taking only the most frequent value for each contextual feature (e.g. if for the feature "mood" there are 10 interactions with a "positive" value, 2 with a "neutral" value and 3 with a "negative" value, only the positive value will be taken). Although some information can be lost, our idea is to capture the general value for each content feature (if such a values even exist). Following this approach the system will be able to detect that some item is related to some specific context. For instance, this film is likely to be consumed at home, alone, in a cloudy day, at night, with a sad emotion, in a negative mood; or this other one is usually consumed at a friend's house, in a sunny day, with my friends, with a happy emotion and in a positive mood.

User's ratings need to be processed in order to adapt the non-binary ratings (1-5) to the binary values required by FCA. This issue has been widely address in the FCA and recommendation research. Some approaches proposed to this end are: 1) consider each rating as a relation [9], 2) apply fuzzy-FCA techniques [20] or, 3) create multi-valued *formal contexts* [13]. Nevertheless, the common way to address this problem has been based on the discretization of the user ratings. In this paper we follow this latter methodology, proposing four different ways to process the user's ratings:

- **Raw:** All ratings will be considered as a relation between the user and the item. This is a "dummy" baseline approach; it does not consider the differences between the ratings: "I hate this movie" and "I love this movie" will be considered as the same interaction.
- **Like:** This approach consider only the likes: ratings greater or equal to 3. A user-item interaction with a lower rating won't

be considered. It groups items that have been liked in the same context (e.g. I love to watch comedy movies with my friends).

- **Dislike:** The same than the previous approach but using the dislikes: ratings lower than 3. It groups items disliked in the same context (I hate horror movies when I am alone).
- **Like and Dislike:** It takes into account likes and dislikes but differentiating them. This approach will group items liked, disliked or even items liked and disliked in the same context (i.e. I do not like to watch romantic comedies with my friends but I would like to watch them with my girlfriend).

In total we propose 24 item-based models (combining the 6 context types with the 4 ways to process the item ratings). The notation of each model denotes the context and processing used. For instance the model *itemLikeContextLocation* refers to the model applying the Item-based modelling to the "location" context information and using only the "like" interactions.

3.3.2 User-based Modelling

This modelling intends to find relationships between users according to the contextual features related to them. In order to collect the contextual features related to the items, the process is similar to the one followed in the Item-based Modelling. It takes the contextual features related to each user-item interaction and in case of conflicting information, the most frequent value for each feature will be taken. By applying this modelling, the system will be able to infer that some users are related to a given context. In the same way than in the Item-based modelling, we proposed 24 different User-based models combining the 6 context types and the 4 ways to process the item ratings.

3.3.3 User-item-based Modelling

This last modelling tries to combine both previous approaches. More in detail, this approach pursues to find the relationships between users in a similar way than in the User-based Modelling, but complementing the contextual information with the consumed items. The rationale is that the user modelling can be improved by including the user-item interactions. This modelling will allow the inferring of relations such as: this users set has consumed this film set but only in this given context. The same configuration is followed in order to collect the contextual information related to the user-item interactions and to create the 24 different models. The notation is similar to the one already proposed: a model called *userItemDislikeContexSocial* refers to a model applying the User-item-based modelling to the "social" context information and using only the "dislike" interactions.

3.4 Recommendation Proposal

The recommendation process is conducted by going through the lattice structure and using the contents included in it to be recommended, for each modelling proposal a different recommendation algorithm has been developed.

3.4.1 Item-based Recommendation

This approach make use of the Item-based modelling, wherein items are grouped according to their shared features. The recommendation is therefore based on suggesting similar items to those already consumed.

More detailed, the system go through each of the items consumed by a user, looks for the *object concept* related to the item (see section 3.2), takes their children (the concepts linked in the level below of a given one) and sibling (the children concepts of the concepts linked above of a given one, except the concept itself) concepts and recommends the items contained in them. A formal definition of the recommendation algorithm is showed in the **Figure 2**.

Algorithm 1 Item-based Recommendation

Require: UserItemSet: $I_{User} = \{item_1, \ldots, item_n\}$ not empty.
for $item_i \in I_{User}$ **do**
  Get *object concept* of $item_i$: (γi)
  $targetConceptsList := \gamma i.$
  for level = 0 to N **do**
    for $targetConcept \in targetConceptsList$ **do**
      $newTargetConceptsList := \emptyset$
      Get *children_concepts*$([C_1, \ldots, C_n])$ of $targetConcept$
      for $C_i \in [C_1, \ldots, C_n]$ **do**
        if $i \in C_i ==$ false **then**
          Add C_i to the $recoCandidateFCList$
          Add C_i to the $newTargetConceptsList$
      $targetConceptsList := newTargetConceptsList$

  $targetConceptsList := \gamma i$
  for level = 0 to N **do**
    for $targetConcept \in targetConceptsList$ **do**
      $newTargetConceptsList := \emptyset$
      Get *sibling_concepts*$([S_1, \ldots, S_n])$ of $targetConcept$
      for $S_i \in [S_1, \ldots, S_n]$ **do**
        if $i \in S_i ==$ false **then**
          Add S_i to the $recoCandidateFCList$
          Add S_i to the $newTargetConceptsList$
      $targetConceptsList := newTargetConceptsList$

for Formal Concept $FC = (A, B) \in recoCandidateFCList$ **do**
  for $item_i \in A$ **do**
    Add $item_i$ to the $RecommendationList$
return $RecommendationList := \{item_1, \ldots, item_n\}$

Figure 2 – Item-based Recommendation Algorithm. Being N the number of levels to look for recommendations.

The *object concept* is the most specific formal concept in which an item is included. Thus, the formal concepts (and consequently the items contained in them) closer to this one in the lattice structure will be the ones with the most specific relationships. It is expected that it leads to more accurate recommendations. For instance, if two items have been consumed in the user's home at night with all their friends and when the user is in a good mood, they will be more closely related than two items that have been consumed in a sunny day.

3.4.2 User-based Recommendation

This method is based on the User-based modelling which relates users according to their shared contextual features. Broadly speaking, this approach recommends items consumed by similar users (and not yet consumed by the target user). To that end, given a target user, the algorithm looks for the *object concept* of the user, gets the children and sibling concepts (as in section 3.4.1), takes the users in this object concept and recommends the items consumed by this user set. This algorithm applies a collaborative filtering methodology, the set of related users in the children and sibling concept can be seen as the *target user neighbourhood*. In the **Figure 3** the definition of the algorithm is shown.

Algorithm 2 User-based Recommendation

```
Require: UserSet: User_Set = {user_1, ..., user_n} not empty.
  for user_i ∈ User_Set do
    Get object concept of user_i: (γi)
    targetConceptsList := γi.
    for level = 0 to N do
      for targetConcept ∈ targetConceptsList do
        newTargetConceptsList := ∅
        Get children_concepts([C_1, ..., C_n]) of targetConcept
        for C_i ∈ [C_1, ..., C_n] do
          if i ∈ C_i == false then
            Add C_i to the userRelatedFCList
            Add C_i to the newTargetConceptList
        targetConceptsList := newTargetConceptList

    targetConceptsList := γi
    for level = 0 to N do
      for targetConcept ∈ targetConceptsList do
        newTargetConceptsList := ∅
        Get sibling_concepts([S_1, ..., S_n]) of targetConcept
        for S_i ∈ [S_1, ..., S_n] do
          if i ∈ S_i == false then
            Add S_i to the userRelatedFCList
            Add S_i to the newTargetConceptsList
        targetConceptsList := newTargetConceptsList

  for Formal Concept FC = (A, B) ∈ userRelatedFCList do
    for user_i ∈ A do
      Get UserItem_List = {item_1 ... item_j} of user_i
      for each item_i ∈ UserItem_List do
        Add item_i to the RecommendationList
  return RecommendationList := {item_1, ..., item_n}
```

Figure 3 – User-based Recommendation Algorithm.

Algorithm 3 User-item-based Recommendation

```
Require: UserSet: User_Set = {user_1, ..., user_n} not empty.
  for user_i ∈ User_Set do
    Get object concept of user_i: (γi)
    targetConceptsList := γi.
    for level = 0 to N do
      for targetConcept ∈ targetConceptsList do
        newTargetConceptsList := ∅
        Get children_concepts([C_1, ..., C_n]) of targetConcept
        for C_i ∈ [C_1, ..., C_n] do
          if i ∈ C_i == false then
            Add C_i to the recoCandidateFCList
            Add C_i to the newTargetConceptsList
        targetConceptsList := newTargetConceptsList

    targetConceptsList := γi
    for level = 0 to N do
      for targetConcept ∈ targetConceptsList do
        newTargetConceptsList := ∅
        Get sibling_concepts([S_1, ..., S_n]) of targetConcept
        for S_i ∈ [S_1, ..., S_n] do
          if i ∈ S_i == false then
            Add S_i to the recoCandidateFCList
            Add S_i to the newTargetConceptsList
        targetConceptsList := newTargetConceptsList

  for Formal Concept FC = (A, B) ∈ recoCandidateFCList do
    for item_i ∈ B do
      Add item_i to the RecommendationList
  return RecommendationList := {item_1, ..., item_n}
```

Figure 4 – User-item-based Recommendation Algorithm.

3.4.3 User-Item-based Recommendation

This last approach uses the User-item-based Modelling. This modelling, as the User-based modelling, intends to group similar users; however in this case the user-item interactions are included and not only the contextual information. Consequently, the recommendation algorithms is similar to the User-based Recommendation: it looks for similar users to a target one to recommend the items consumed by them. For that, the algorithm looks for the *object concept* of the target user, gets the children and sibling concepts and recommends the items in these concepts. In the **Figure 4**, the algorithm operation is detailed.

4. RESULTS

In this section we analyse the results obtained for each modelling, expressed in terms of Mean Absolute Error (MAE) and Root-Mean-Square Error (RMSE) [6]. In order to have a comparison point, we use the results obtained by the creators of the collection who applied a state-of-the-art recommender systems. In the Figure 5, extracted from [23], the absolute results obtained by their approach are exposed in terms of RMSE. Other works have made use of this dataset; for instance, the authors of [16] present a comparison of several recommender algorithms (Context-Aware Matrix Factorization, Similar Trends Identifying) or the authors of [29] wherein the splitting of users and items is studied. However, these works and other ones [22] produce similar (or worse) results to those presented in [23].

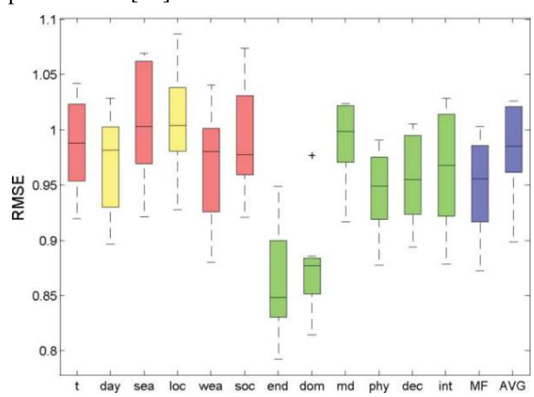

Figure 5 – Results for the LDOS-CoMoDa dataset [23]

4.1 Item-based Results

Table 1 details the results obtained by the Item-based recommendation (see algorithm in **Figure 2**) in terms of MAE and RMSE. Note that both, MAE and RMSE, are error-based measured (i.e., they measure the incorrect recommendations done by the system), so the lower the value, the better the system performance.

The clearest remark that can be extracted from these results is the top performance of the *LikeAndDislike* rating processing for all the context variables. This rating processing takes into account positive and negative ratings by separate in the recommendation process and, as we hypothesized before, it is the best way to manage the ratings. It does make sense; it is this approach the one that takes into account all the information (in contrast to the approaches using only likes and dislikes) but managed in a better way that the "Raw" approach that does not differentiate between likes and dislikes.

It is also remarkable that the negative ratings (Dislike) outperform the positive ones; in fact, positive interactions offer similar results (not significantly better) than the raw interactions. That is, according to this finding, two users who do not like to watch movies at home are more closely related than two users who do.

Focusing in the context types, none of them obtains a clearly better performance than the other ones, not even the aggregation of all of them. Only the "Decision" type seems to clearly offer a worse performance than the other ones. It can be seen in two ways, either all the different kinds of contextual information offers the same performance (i.e., they are able to avoid the same incorrect recommendations) so their aggregation is not able to avoid new erroneous recommendations, or the contextual information does not really affect the recommendation process and we are only seen the baseline recommender system performance in all of the approaches.

This point can be answered by taking a look to the baseline approach (i.e., the one without contextual information). All the context-aware recommendations achieve a significantly improvement in comparison to the baseline. Thus, we can conclude that contextual information does influence the recommendation process, improving it.

Table 1 – Item-based Results

APPROACH	MAE	RMSE
itemInfo	0.7878	1.3449
itemRawContextAll	0,4968	0,9385
itemRawContextDecision	0,5016	1,0649
itemRawContextLocation	0,4653	0,8969
itemRawContextSocial	0,4666	0,9077
itemRawContextTime	0,4728	0,9229
itemRawContextUser	0,4769	0,9232
itemLikeContextAll	0,4993	0,9628
itemLikeContextDecision	0,502	1,0656
itemLikeContextLocation	0,4636	0,8981
itemLikeContextSocial	0,4617	0,8973
itemLikeContextTime	0,4665	0,9069
itemLikeContextUser	0,4706	0,9136
itemDislikeContextAll	0,456	0,886
itemDislikeContextDecision	0,5431	1,0562
itemDislikeContextLocation	0,4266	0,8299
itemDislikeContextSocial	0,4163	0,8109
itemDislikeContextTime	0,408	0,7966
itemDislikeContextUser	0,3990	0,7797
itemLikeAndDislikeContextAll	0,4054	0,7895
itemLikeAndDislikeContextDecision	0,4024	0,7864
itemLikeAndDislikeContextLocation	0,4067	0,7946
itemLikeAndDislikeContextSocial	0,4120	0,8036
itemLikeAndDislikeContextTime	0,4145	0,8075
itemLikeAndDislikeContextUser	0,4187	0,8145

4.2 User-based Results

Table 2 recompiles the MAE and RMSE results for the User-based recommender. Table 2 – User-based Results

APPROACH	MAE	RMSE
userInfo	1,1738	2,004
userRawContextAll	1,0986	1,8633
userRawContextDecision	1,0893	1,8143
userRawContextLocation	0,9255	1,5157
userRawContextSocial	1,0638	1,7319
userRawContextTime	1,1296	1,8485
userRawContextUser	1,2007	1,9656
userLikeContextAll	1,228	2,006
userLikeContextDecision	1,2066	1,9667
userLikeContextLocation	1,153	1,8704
userLikeContextSocial	1,1741	1,9050
userLikeContextTime	1,1932	1,9435
userLikeContextUser	1,2202	1,99
userDislikeContextAll	0,6072	0,9899
userDislikeContextDecision	0,6723	1,0692
userDislikeContextLocation	1,1425	1,8651
userDislikeContextSocial	1,1298	1,8472
userDislikeContextTime	1,1142	1,8252
userDislikeContextUser	0,9256	1,5091
userLikeAndDislikeContextAll	0,9378	1,397
userLikeAndDislikeContextDecision	1,0059	1,5801
userLikeAndDislikeContextLocation	1,0554	1,7329
userLikeAndDislikeContextSocial	1,0672	1,7498
userLikeAndDislikeContextTime	1,0798	1,7724
userLikeAndDislikeContextUser	1,0939	1,798

As a general conclusion it can be said that these results confirm the already obtained in the previous approach.

As in the Item-based recommendation, the best results are obtained by applying the *LikeAndDislike* rating processing; however, in this case the differences are not as clear as in the previous case. Regarding the negative interactions, they seem to be again more appropriate to infer recommendations.

It is also important to remark, as it happened in the item-based results, how contextual information is able to outperform the baseline recommendation (i.e., the one without contextual information) and, regarding the different types of contextual information, how none of them stands out above the other ones.

4.3 User-item-based Results

Finally, the User-item recommendation results are detailed in the Table 3, in terms of MAE and RMSE. These results confirm again the conclusions pointed out by the two previous approaches. That is, the performance of negative interactions is better than the one of the positive interactions, the best results are obtained by applying the *LikeAndDislike* rating processing, and the similar performance of all the context types, but improving the baseline recommendation.

Table 3 – User-item-based Results

APPROACH	MAE	RMSE
userInfo	1,1738	2,004
userItemRawContextAll	0,4782	0,6036
userItemRawContextDecision	0,4181	0,7942
userItemRawContextLocation	0,4068	0,7668
userItemRawContextSocial	0,4301	0,8189
userItemRawContextTime	0,4249	0,8163
userItemRawContextuserItem	0,3957	0,7499
userItemLikeContextAll	0,4191	0,7996
userItemLikeContextDecision	0,4225	0,8134
userItemLikeContextLocation	0,4161	0,7984
userItemLikeContextSocial	0,4205	0,8074
userItemLikeContextTime	0,4151	0,7993
userItemLikeContextuserItem	0,4017	0,7694
userItemDislikeContextAll	0,3818	0,5987
userItemDislikeContextDecision	0,3831	0,6222
userItemDislikeContextLocation	0,3703	0,6013
userItemDislikeContextSocial	0,3692	0,7125
userItemDislikeContextTime	0,3588	0,5825
userItemDislikeContextuserItem	0,3387	0,5494
userItemLikeAndDislikeContextAll	0,3036	0,4949
userItemLikeAndDislikeContextDecision	0,3221	0,5214
userItemLikeAndDislikeContextLocation	0,3091	0,4888
userItemLikeAndDislikeContextSocial	0,3161	0,4998
userItemLikeAndDislikeContextTime	0,3091	0,4888
userItemLikeAndDislikeContextuserItem	0,3121	0,4809

One important point to note about this approach is that it achieves the best result of all of the three recommendation approaches. This approach includes the user and item information in a sort of hybrid recommendation. Consequently, it is reasonable that the best results are obtained by the approach including as much information as possible about the user-item interactions. If we take the results in [23], or in the aforementioned works using this collection, as baseline, it can be seen as all the User-item approaches outperforms the ones obtained in [23], especially when *LikeAndDislike*-based results are taken into account.

5. CONCLUSIONS

In this work we proposed an experimental study in the field of context-aware recommender systems. Many works have been previously done in this field; however, the novelty of the work herein proposed is the application of a concept-based modelling approach (Formal Concept Analysis). We intended to take advantage of its high-performance in content organization to model user-item interactions, integrating a new dimension (the contextual information) in the modelling process. Besides the application of FCA for Context-aware Recommendation, in the experimental setup two other aspects were addressed: 1) the differentiation and processing of negative and positive ratings, and 2) the splitting of contextual information in different types. Our hypothesis was that both aspects play an important role in the recommendation process, so its managing should lead to an improvement in the overall recommendation performance.

From the experimental results, different conclusions can be extracted. Regarding the overall performance of our proposal, comparing the FCA-based results to the state-of-the-art approaches, it was proven as a suitable technique for context-aware recommendation, outperforming the ones in the literature. Focusing on the two aspects we want to experiment with, the managing of different types of ratings (negative and positive) has been proven as an important aspect in the recommendation process. While the approaches which do not take into account this differentiation offer the worst performance, the *LikeAndDislike* approaches which differentiate and manage both types of interactions, obtain the best performance. An important remark here is that the negative interactions are the ones more related to the final performance: the users are more closely related by what they dislike than by what they like.

By taking into account the different types of contextual information, the results refutes our initial hypothesis. To include the contextual information by taking each type by separate does not seems to affect to the final system performance. In other words, no contextual information type is more useful to infer user preferences than other one. However, the inclusion of contextual information does represent a helpful information for the recommendation process, improving the baseline algorithm performance (i.e. the one that does not include contextual information).

Finally, some future work lines can be drawn. The first one is related to the managing of user's ratings. A general threshold has been applied; that is, a rating lower than 3 was considered as a negative one and a rating higher as a positive one. However, in the state of the art of recommendation some ideas have been proposed in order to refine this process, adapting this threshold to each user based on these previous rating distribution. Since this work proved this parameter as an important one in the recommendation process, its refinement shall lead to a better recommendation process. Focusing on the contextual information, the proposed processing seemed to have a low impact in the final performance. However, maybe a more fine-grained context classification, even taking each context variable by separate, or a different way to integrate this information might have more impact in the recommendation process.

6. ACKNOWLEDGMENTS

This work has been partially supported by the Spanish Government under the project VOXPOPULI (TIN2013-47090-C3-1-P).

7. REFERENCES

[1] ADOMAVICIUS, G., MOBASHER, B., RICCI, F., AND TUZHILIN, A. Context-aware recommender systems. *AI Magazine 32(3)* (2011), 67–80.

[2] ADOMAVICIUS, G., AND TUZHILIN, A. Context-aware recommender systems. In *Recommender Systems Handbook*. Springer, 2011, pp. 217–253.

[3] BALTRUNAS, L., AND RICCI, F. Context-based splitting of item ratings in collaborative filtering. In *Proceedings of the Third ACM Conference on Recommender Systems* (New York, NY, USA, 2009), RecSys '09, ACM, pp. 245–248.

[4] BETTINI, C., BRDICZKA, O., HENRICKSEN, K., INDULSKA, J., NICKLAS, D., RANGANATHAN, A., AND RIBONI, D. A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing 6*, 2 (2010), 161–180.

[5] CAMPOS, P. G., CANTADOR, I., AND DÉZ, F. Exploiting time contexts in collaborative filtering: An item splitting approach. In *Proceedings of the 3rd Workshop on Context-awareness in Retrieval and Recommendation* (New York, NY, USA, 2013), CaRR '13, ACM, pp. 3–6.

[6] CANDILLIER, L., JACK, K., FESSANT, F., AND MEYER, F. State-of-the-art recommender systems. *Collaborative and Social Information Retrieval and Access-Techniques for Improved User Modeling* (2009), 1–22.

[7] CASTELLANOS, A., GARCÍA-SERRANO, A., AND CIGARRÁN, J. Linked Data-based Conceptual Modelling for Recommendation: A FCA-based Approach. In *E-Commerce and Web Technologies*, M. Hepp and Y. Hoffner, Eds., vol. 188 of *Lecture Notes in Business Information Processing*. Springer International Publishing, 2014, pp. 71–76.

[8] CREMONESI, P., GARZA, P., QUINTARELLI, E., AND TURRIN, R. Top-n recommendations on unpopular items with contextual knowledge. In *2011 Workshop on Context-aware Recommender Systems. Chicago* (2011).

[9] DU BOUCHER-RYAN, P., AND BRIDGE, D. Collaborative recommending using formal concept analysis. *Knowledge-Based Systems 19*, 5 (2006), 309–315.

[10] FANG, P., AND ZHENG, S. A research on fuzzy formal concept analysis based collaborative filtering recommendation system. *Knowledge Acquisition and Modeling, International Symposium on 3* (2009), 352–355.

[11] GANTER, B., WILLE, R., AND FRANZKE, C. *Formal concept analysis: mathematical foundations*. Springer-Verlag New York, Inc., 1997.

[12] IGNATOV, D., KAMINSKAYA, A., BEZZUBTSEVA, A., KONSTANTINOV, A., AND POELMANS, J. Fca-based models and a prototype data analysis system for crowdsourcing platforms. In *Conceptual Structures for STEM Research and Education*, H. Pfeiffer, D. Ignatov, J. Poelmans, and N. Gadiraju, Eds., vol. 7735 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2013, pp. 173–192.

[13] IGNATOV, D., KAMINSKAYA, A., BEZZUBTSEVA, A., KONSTANTINOV, A., AND POELMANS, J. FCA-Based Models and a Prototype Data Analysis System for Crowdsourcing Platforms. In *Conceptual Structures for STEM Research and Education*, H. Pfeiffer, D. Ignatov, J. Poelmans, and N. Gadiraju, Eds., vol. 7735 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2013, pp. 173–192.

[14] IGNATOV, D. I., KUZNETSOV, S. O., AND POELMANS, J. Concept-based biclustering for internet advertisement. In *Data Mining Workshops (ICDMW), 2012 IEEE 12th International Conference on* (2012), IEEE, pp. 123–130.

[15] JANNACH, D., ZANKER, M., FELFERNIG, A., AND FRIEDRICH, G. *Recommender systems: an introduction*. Cambridge University Press, 2010.

[16] JEONG, H.-Y., OBAIDAT, M. S., YEN, N. Y., AND PARK, J. J. J. H. *Advances in Computer Science and its Applications: CSA 2013*, vol. 279. Springer Science & Business Media, 2013.

[17] KOŠIR, A., ODIC, A., KUNAVER, M., TKALCIC, M., AND TASIC, J. F. Database for contextual personalization. *Elektrotehniski Vestnik 78*, 5 (2011), 270–274.

[18] LAU, R. Y. K., CHUNG, A. Y. K., SONG, D., AND HUANG, Q. Towards fuzzy domain ontology based concept map generation for e-learning. In *Proceedings of the 6th international conference on Advances in web based learning* (Berlin, Heidelberg, 2008), ICWL'07, Springer-Verlag, pp. 90–101.

[19] LI, X., AND MURATA, T. A knowledge-based recommendation model utilizing formal concept analysis and association. In *Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference on* (2010), vol. 4, pp. 221–226.

[20] MAIO, C. D., FENZA, G., GAETA, M., LOIA, V., ORCIUOLI, F., AND SENATORE, S. Rss-based e-learning recommendations exploiting fuzzy {FCA} for knowledge modeling. *Applied Soft Computing 12*, 1 (2012), 113 – 124.

[21] MOBASHER, B. Contextual user modeling for recommendation. In *Keynote at the 2nd Workshop on Context-Aware Recommender Systems* (2010).

[22] ODIC, A., TKALCIC, M., TASIC, J. F., AND KOSIR, A. Personality and social context: Impact on emotion induction from movies. In *UMAP Workshops* (2013).

[23] ODIC, A., TKALCIC, M., TASIC, J. F., AND KOŠIR, A. Predicting and detecting the relevant contextual information in a movie-recommender system. *Interacting with Computers 25*, 1 (2013), 74–90.

[24] PANNIELLO, U., TUZHILIN, A., AND GORGOGLIONE, M. Comparing context-aware recommender systems in terms of accuracy and diversity. *User Modeling and User-Adapted Interaction 24*, 1-2 (2014), 35–65.

[25] SAID, A., LUCA, E. W. D., AND ALBAYRAK, S. Inferring contextual user profiles - improving recommender performance. In *Proceedings of the 3rd RecSys Workshop on Context-Aware Recommender Systems* (2011).

[26] SENATORE, S., AND PASI, G. Lattice navigation for collaborative filtering by means of (fuzzy) formal concept analysis. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing* (New York, NY, USA, 2013), SAC '13, ACM, pp. 920–926.

[27] SIMOVICI, D., FOMENKY, P., AND KUNZ, W. Polarities, axiallities and marketability of items. In *Data Warehousing and Knowledge Discovery*, A. Cuzzocrea and U. Dayal, Eds., vol. 7448 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2012, pp. 243–252.

[28] WILLE, R. Concept lattices and conceptual knowledge systems. *Computers & mathematics with applications 23*, 6 (1992), 493–515.

[29] ZHENG, Y., BURKE, R., AND MOBASHER, B. Splitting approaches for context-aware recommendation: An empirical study. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing* (New York, NY, USA, 2014), SAC '14, ACM, pp. 274–279.