OPEN ACCESS
International Journal of Uncertainty,
Fuzziness and Knowledge-Based Systems
Vol. 28, No. 6 (2020) 981–1002
© The Author(s)
DOI: 10.1142/S0218488520500385

Int. J. Unc. Fuzz. Knowl. Based Syst. 2020.28:981-1002. Downloaded from www.worldscientific.com
by UNIVERSIDADE DA CORUNA on 06/13/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

# A Content-Based Approach to Profile Expansion

Diego Fernández*,‡, Vreixo Formoso†,§, Fidel Cacheda*,¶ and Victor Carneiro*,‖

*Department of Computer Science, University of A Coruña, CITIC,
Campus de Elviña, 15071, A Coruña, Spain
†VEVI Systems S.L., Torreiro, 13, 6 D, 15001, A Coruña, Spain
‡diego.fernandez@udc.es
§vreixo@vevisystems.com
¶fidel.cacheda@udc.es
‖victor.carneiro@udc.es

Received 24 July 2019
Revised 9 May 2020

Collaborative Filtering algorithms suffer from the so-called cold-start problem. In particular, when a user has rated few items, recommendations offered by these algorithms are not too accurate. Profile Expansion techniques have been described as a way to tackle this problem without bothering the user with additional information requests by increasing automatically the size of the user profile. Up to now, only collaborative approaches had been proposed for Profile Expansion. However, content-based techniques can also be used. We perform a manual analysis of a movie dataset to analyze how content features behave. According to this analysis, we propose a content-based approach, which is also combined with collaborative information. Concretely, we expose the advantages and disadvantages of the combination with a popularity feature. Moreover, a comparison to pure collaborative approaches is performed. Our approach is evaluated in a new system situation. That is, not only the active user has few ratings, but also most of the users. The results show that content-based information is useful for rating prediction. In addition, recommendations are less personalized as popularity feature acquires more relevance for item selection.

*Keywords*: Collaborative filtering; content-based; profile expansion; cold-start.

## 1. Introduction

Users can have different tastes and needs that they want to satisfy. Many e-commerce sites recommend items to the users according to these different tastes to increase their profits. With this purpose, these sites make use of Recommender Systems.

During the last years Recommender Systems have become very popular. In the literature much work about these systems is focused on the *annotation in context*

This is an open access article published by World Scientific Publishing Company. It is distributed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs 4.0 (CC BY-NC-ND) License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.

Int. J. Unc. Fuzz. Knowl. Based Syst. 2020.28:981-1002. Downloaded from www.worldscientific.com
by UNIVERSIDADE DA CORUNA on 06/13/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

(or prediction) task and the *find good items* (or recommendation) task.[14] The former consists in predicting how much a particular item can be liked by a user. On the other hand, in the latter task the system provides the user with a list of relevant items.

Unfortunately, when a user has just arrived to the system or when she has rated very few items, generally Recommender Systems do not have enough information to recommend items accurately. They suffer the well-known cold-start problem and, more concretely, the new user problem. The recommendations proposed by the system are not precise enough to become useful for the user. Different alternatives have been considered in the literature, all of them focused on increasing the size of the user profile (the available information about the user tastes). However, sometimes these approaches do not supply users with recommendations personalized enough. In other situations, the systems require additional information to the users, who do not always provide this information. One reason can be simply that introducing more information bothers them. Privacy concerns are another reason.

We presented the *Profile Expansion*[9] techniques, which use collaborative information, as a way to address this problem without bothering the user.

In this work we focus on the *find good items* task, as many applications nowadays present to each user a list of interesting items according to her interests. We address the new user problem by means of *Content-based Profile Expansion* techniques. These techniques search for items similar to those rated by the active user among all the items available. In fact, we propose a new system situation, that is, an extreme and sparse situation where all the users present few ratings, not only the active user.

This article is organized as follows. In the next section we study the state of the art. Then, we explain in detail our content-based approach. The methodology and experiments are described in Section 4. Next, the results are discussed. Finally, the last sections are focused on the conclusions and the future works.

## 2. Related Work

Collaborative Filtering techniques deal with ratings that users made on items. In order to make a recommendation for a user, they are only based on these ratings.

The $k$-Nearest Neighbors (kNN) algorithms[13] are a popular technique. They can be user-based[26,29] or item-based[27] depending on whether they work by finding a set of users similar to the active user, or a set of items similar to those in the active user profile. More formally, to recommend an item list to a user $u \in U$, where $U$ is the set of users in the system, a user-based kNN algorithm first compute a neighborhood $N(u)$, with the $k$ users most similar to the user $u$ who have a minimum number of items rated in common with her. The similarity is computed according to a previously defined similarity function $s : U \times U \to \Re$. Pearson correlation[26] or cosine vector similarity[5] are commonly used. The items to recommend are chosen among those rated by the neighbors.

Int. J. Unc. Fuzz. Knowl. Based Syst. 2020.28:981-1002. Downloaded from www.worldscientific.com
by UNIVERSIDADE DA CORUNA on 06/13/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

On the other hand, content-based techniques[23] consider the features that characterize the items rated by a user and they take into account these features to find items to be recommended. However, because they are based on content features and not on user opinions, they cannot distinguish the quality of an item. That is, if two items present the same values for all the content features, they will be considered equally, although their quality can be really different.

### 2.1. *Cold-start problem*

Collaborative Filtering techniques obtain pretty good results when many ratings are available. However, in real systems, information is usually scarce, since users only rate a small subset of items. For example, users watch few movies among the hundreds of movies that are released every year, so data in a movie recommendation system is sparse because of its own nature. Data from other domains, such as web page recommendation, is even sparser. While state-of-the-art approaches handle pretty well the common, relatively sparse, conditions, their accuracy diminishes with increasing sparsity.[7] particular, there are three situations where the information is extremely sparse independently of the domain nature:[20]

- *new item* or recommending an item recently introduced in the system.
- *new user* or generating recommendations for users who have joined the system recently.
- *new system* or generating recommendations when the system has just been deployed.

In general, these situations cannot be properly handled by Collaborative Filtering techniques. This is the well-known cold-start problem, which has been a challenge for researchers since the early days of Collaborative Filtering. In the literature we can find different techniques to address each particular situation, many of which consider other information in addition to user ratings. They are the so called hybrid approaches.[1,6]

Regarding the new item problem, ratings can be combined with content-based information. This information about an item can be easily obtained, so hybrid approaches are very useful for new items, for which the system has little or none collaborative information. Moreover, content-based algorithms do not suffer from the new item problem, because they only need a suitable user profile to provide recommendations, including items not rated by any user yet.

Some previous solutions use the so called collaboration via content, which is a paradigm that consists in using the content-based user profiles in order to detect user similarities.[22] This extra content-based information can be simply used to estimate user ratings and improve the user profile. Melville *et al.*[17] used an extended naive Bayesian text classifier to estimate unknown user ratings using several movie features. Then, the full profile is used by a regular Collaborative Filtering

Int. J. Unc. Fuzz. Knowl. Based Syst. 2020.28:981-1002. Downloaded from www.worldscientific.com
by UNIVERSIDADE DA CORUNA on 06/13/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

algorithm. Other authors have studied the utility of bots,[11,21] that is, fake users that automatically rate items according to content-based information.

Other systems have been designed where the content information is part of the model. Schein *et al.*[28] modeled the content data by means of a latent-factor aspect model that relates users with movie casts, in addition to a purely collaborative aspect-model to relate users with movies. More recently, Gunawardana and Meek[12] employed Boltzman machines to model user-item relationships, using content information to estimate model parameters. They modeled the content as vectors of features, where a feature could represent term weight, presence of an actor, etc. Several other techniques have also been studied.[16,20]

The same idea (that is, to combine collaborative information with additional data) has also been proposed as a solution to the new user problem. In this case, sources of information other than content-based have been studied, such as demographic data[15,18] or trust relationships in a social network.[19]

Contrary to the new item case, where the extra information for the item is easily available, obtaining additional information for the new user case usually requires to ask the user. It does not matter whether the user is requested to provide her age, sex, or just to rate some initial movies. So, many systems have chosen to directly require the user to rate some items before any recommendation. In that case, a pure Collaborative Filtering algorithm can be used.

Actually, requiring users to provide additional information is an interesting solution. As the user is expected to get more accurate recommendations in exchange for the additional information, she will probably be willing to do it.

However, requiring too much information at early stages can be a problem. Using Collaborative Filtering, if the items the user is required to rate are not properly chosen, the initial profile may be useless (for example, if the user does not know about most proposed items), and the user will be soon disappointed. In the same manner, if the number of items is not suited to provide the user with accurate enough recommendations, the user can leave the system. To avoid this, some authors have proposed techniques to optimize this information gathering process, also known as *preferences elicitation*.[10,25]

In many cases, however, it is still better to take advantage of the available information, even if it is scarce, than bothering the user asking for more information. If the recommendations are reasonably accurate from the beginning, the user will continue to use the system, providing increasingly more information. For example, if a system is able to provide reasonably accurate recommendations for a user with a single rating, the user will be probably willing to rate the recommended movies in order to improve her profile.

Some authors realized this fact, and they attempted to perfect algorithms for the new user situation, instead of requiring new users to provide additional information. For example, Ahn[2] proposed a new similarity measure for kNN algorithms that improves performance on new users. It is based on proximity, impact and popularity

Int. J. Unc. Fuzz. Knowl. Based Syst. 2020.28:981-1002. Downloaded from www.worldscientific.com
by UNIVERSIDADE DA CORUNA on 06/13/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

of user ratings. It was designed as a replacement for traditional measures such as cosine or Pearson correlation. Bobadilla *et al.*[4] presented another new similarity measure based on the combination of six individual similarity measures. Piccart *et al.*[24] proposed a probabilistic graph-based method that use information present in indirect neighbors.

### 2.2. *Profile Expansion*

Profile Expansion techniques are focused on increasing the initial user profile with additional items. In order to achieve this expansion, different techniques can be applied. We showed the advantages of using Profile Expansion techniques based on Collaborative Filtering to tackle the new user problem.[9] Several alternatives were proposed in order to obtain the items to be included in the expanded profile:

- Item-local. The items are chosen from an initial recommendation, which is not shown to the user.
- Item-global. An algorithm is used for obtaining items in the whole dataset that are similar to those in the initial profile.
- User-local. After computing the neighborhood for the active user, the items are selected from those rated by the neighbors.
- User-global. Since there are similarities between users, these similarities can be computed for the active user paying attention to demographic data, for example. The items are chosen from those rated by these similar users.

In this article, we continue our previous work We tackle the new user problem in a movie recommendation domain without bothering the user with additional information requests.

In the recommendation process two algorithms are involved: the expansion and the recommendation algorithms. In this article we continue our previous work,[9] but, rather than using only Collaborative Filtering techniques, the recommender algorithm remains a Collaborative Filtering algorithm, while the expansion algorithm is content-based.

### 2.3. *Notation*

Collaborative Filtering techniques are based on the opinions users have on items. With these opinions, collaborative user profiles are created. More formally, we define the user profile as a list of item-rating pairs, where each pair is made of an item $i \in I$ the user has rated, and the corresponding rating $r_{ui}$, as shown in Eq. (1). We denote as $P_u^c$ the collaborative user profile of a user $u \in U$. The user profile is comprised of $n$ item-rating pairs, being $n$ the number of items rated by the user.

$$P_u^c = \langle (i_1, r_{ui_1}), (i_2, r_{ui_2}), \ldots, (i_n, r_{ui_n}) \rangle \ . \tag{1}$$

As Profile Expansion techniques are focused on increasing the initial user profile, we define the *expanded profile*, $P_u^{c'}$, as the union of the original profile with up to $l$

additional items. That can be seen in Eq. (2).

$$P_u^{c'} = P_u^c \cup \left\langle \left(i_{n+1}, r_{ui_{n+1}}\right), \left(i_{n+2}, r_{ui_{n+2}}\right), \ldots, \left(i_{n+l}, r_{ui_{n+l}}\right) \right\rangle . \tag{2}$$

Additionally, we define the set of items already rated by the user as $I_u$, and the items added to the profile as $I_u^e$. Of course, $I_u \cap I_u^e = \emptyset$.

## 3. Content-Based Profile Expansion

The idea of applying a content-based technique for Profile Expansion is justified by the fact that it is capable of finding semantic relations between items. In a new user situation, the information about user tastes is scarce, while content information about items can be relatively important. This information can be represented by means of a set of features. The user will have a profile for each feature. Based on these content profiles, a content-based technique searches for items similar to those rated by the user. This way, we deal with the new user problem searching for items similar to user tastes. In order to do that, first we identify the features that best define these user tastes by means of a manual analysis. We also identify how to compare a content user profile with the content information about a particular item. Once these items are selected, they are added to the user profile as if they were actually rated by the user. So, these ratings must be estimated. After that, the recommender algorithm is executed as usual. Following, we analyze the main phases of the Content-based Profile Expansion.

### 3.1. *Features selection*

The aim of the expansion is to find items similar enough to the initial items of the user profile. Firstly, it is necessary to analyze the available features in order to determine which are more useful for comparing items or, in other words, which features are the most relevant to determine the user tastes and to characterize the items.

In this work, we use a filtered Netflix[3] dataset, which will be described in Section 4. In order to identify the most useful features to characterize the user profiles, we have performed a manual analysis by randomly selecting 25 users. Moreover, two items per user have been chosen as the initial user profiles. This first analysis led us to several conclusions, which were confirmed with an automatic analysis of 1000 users (also randomly chosen). The mean and the standard deviation of the number of items rated per user are 99.93 and 135.66, respectively. In fact, only 33% of user profiles have rated more than 100 items.

Following, we explain the features we have selected in this first analysis: genre, country, director and cast.

#### 3.1.1. *Genre*

Each movie is categorized according to certain settings, mainly related to the type of the most abundant scenes in it. This way, a movie is considered to fit a set of

Int. J. Unc. Fuzz. Knowl. Based Syst. 2020.28:981-1002. Downloaded from www.worldscientific.com
by UNIVERSIDADE DA CORUNA on 06/13/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

Int. J. Unc. Fuzz. Knowl. Based Syst. 2020.28:981-1002. Downloaded from www.worldscientific.com
by UNIVERSIDADE DA CORUNA on 06/13/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

molds. However, the belonging to certain genres such as action, comedy, drama, etc. is subjective to a great extent. In fact, there are several aspects that are not included in the genre feature. For example, movies having the same genres can consider different topics, cultures, etc. Therefore, this feature allows to classify the movies in groups, although this classification presents certain limitations since it is not purely objective. Despite that, this is the content feature that a priori best defines the user interests and the similarities between movies.

If a genre value appears more frequently in a user profile than in the dataset, it denotes the user preference by this particular genre value. Otherwise, if a genre value appears less frequently, it usually means that the user is not interested in this kind of movies. User distribution approximates dataset distribution when user profile size exceeds the mean size. In addition to this, items sharing genre values and having similar quality are rated similarly by the users.

#### 3.1.2. *Country*

A movie gives insights into a particular culture, a way of life, etc. Principally the country of the movie is which determines this perception. This can accentuate the differences between movies. For example, among Indian movies there are a lot of musicals. However, these musicals can differ a lot from a French musical. Users do not usually watch movies from all the countries, but from a subset of them.

In the dataset we analyzed, when an initial user profile presents a particular country value, it tends to appear frequently in the whole user profile.

#### 3.1.3. *Director*

Every director has a particular style making the movies. That can be to the user taste or not. In the profiles studied, users do not usually rate movies from the same director. However, that does not imply that this feature cannot be useful. For example, if a user has rated "Rear Window" (directed by Alfred Hitchcock), it is reasonable to think that she would rate "Vertigo" similarly. Therefore, this movie becomes a candidate to the expansion. Nevertheless, the directors do not appear more than once in the initial profile. In fact, the values in the initial user profile hardly ever appear in the whole user profile, although, when they appear, they are rated similarly.

#### 3.1.4. *Cast*

Like directors, actresses and actors are not usually repeated in the user profiles studied. Following the same reasoning as for directors, when a movie in a user profile presents a relevant coincidence in their casts with other movie not rated by the user yet, it seems reasonable to expand the profile with it. However, the cast values in the initial profile seldom appear in the whole profile again, despite the fact that initial profile usually contains around 43 cast values on average.

Int. J. Unc. Fuzz. Knowl. Based Syst. 2020.28:981-1002. Downloaded from www.worldscientific.com
by UNIVERSIDADE DA CORUNA on 06/13/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

The reason is that actresses and actors only appear 4.09 times on average, being the standard deviation 3.87. Moreover, in general, movies from the same actors do not vary excessively since a lot of them are better at certain roles (they are typecast). So, users tend to rate movies from the same actors in a similar way. Therefore, this feature can be useful to estimate item ratings.

### 3.2. *Feature representation*

After having selected the features, the next phase is to decide how to represent them. We denote as $F$ the set of content features taken into account. Let be $L^f$ the set of different values or levels that a feature can take. For example, we consider the genre as a feature in the set $F$. $L^f$ will contain values such as action, comedy, drama, etc. Moreover, an item can present a subset of all the levels that can take a feature. Therefore, we can define $L_i^f$ as the levels that a feature takes for a particular item.

We denote as $P_u^f$ the content user profile of a user $u \in U$ for a content feature $f \in F$. Each user profile is comprised of $m$ pairs of a value $l_i \in L^f$, and the corresponding weight $w_{ui}$, being $m$ the number of levels for the feature $f$, $\left|L^f\right|$. The definition is shown in Eq. (3).

$$P_u^f = \langle (l_1, w_{u1}), (l_2, w_{u2}), \ldots, (l_m, w_{um}) \rangle \ . \tag{3}$$

In this first approach, these weights are just the number of times that each level appears in the $L_i^f$ of the items in the user profile. For example, if a user $u_1$ has rated one comedy movie and a comedy and drama movie, then the weight for comedy will be 2, for drama will be 1 in her user profile, and 0 for the remaining levels.

Similarly, we denote as $P_i^f$ the content item profile of an item $i \in I$ for a content feature $f \in F$.

### 3.3. *Similarity measures*

In this work we propose two measures to compute the similarity between a user profile and an item profile. These measures are the traditional cosine vector similarity and a heuristic function that we call *weighted similarity*.

#### 3.3.1. *Cosine vector similarity*

This similarity measures the distance between the content user profile and the content item profile. It is computed according to Eq. (4).

$$s^f(u,i) = \frac{\sum_{l \in L^f} w_{ul} w_{il}}{\sqrt{\sum_{l \in L^f} w_{ul}^2} \sqrt{\sum_{l \in L^f} w_{il}^2}} \ . \tag{4}$$

#### 3.3.2. *Weighted similarity*

In Section 3.1, it is mentioned that if a value appears more than once in the initial user profile, then it will probably appear more times in the whole user profile.

Int. J. Unc. Fuzz. Knowl. Based Syst. 2020.28:981-1002. Downloaded from www.worldscientific.com
by UNIVERSIDADE DA CORUNA on 06/13/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

Weighted similarity represents this fact. It is depicted in Eq. (5), where $u$ is a user, $i$ is an item, $w_{ul}$ is the weight that takes the level $l$ in the content user profile $P_u^f$, and $n$ is the number of items rated by the user $u$.

$$s^f(u,i) = \max\left\{ \frac{w_{ul}}{n} \mid (l, w_{ul}) \in P_u^f \wedge l \in L_i^f \right\} . \tag{5}$$

For example, the similarity between the user $u_1$ and an item $i_1$, with the levels comedy, drama and romance for the genre feature, will be the maximum of weights associated to the three genre values divided by 2, the number of movies that $u_1$ rated. Since the weights are 2/2, 1/2 and 0/2, respectively, the similarity will be 1.

### 3.4. *Items selection*

Items are sorted according to a certain weight. Several features are used, so the different similarity values must be combined to produce a unique value to sort the ordered list. In particular, we have chosen genre and country features for the item selection since we concluded in the analysis that these features are the most suitable to determine the user tastes.

The content similarity, $w_c$, between an item and the user profile will be a weighted combination of the similarities computed for every feature. Then, the items with the highest weights will be chosen. We have decided to employ an heuristic method for this.

Moreover, we aim at maximizing the expansion size, since, in general, the recommendations are more accurate as the user profile is bigger. Unlike elicitation techniques, where users explicitly rate new items suggested by the system, expansion techniques increase the size of the user profile without user intervention. Logically, this can cause errors, which must be minimized without excessively decreasing the accuracy. Therefore, the selection of the number of elements to expand becomes a key decision.

#### 3.4.1. *Collaborative information*

In cold-start situations and, more precisely, with the new user problem, there are few items in the user profile. This fact leads to situations where it is not easy to find enough users in the system who share with the active user a minimum number of items. So, the neighborhood will be based on few users. With a small neighborhood, it is complicated to reach an accurate recommendation list.

Once the expansion has been computed, with a bigger user profile, if these items seldom appear in other user profiles, despite being very similar to those in the initial user profile, the neighborhood will still be based on few users. So, it seems a good idea to consider the *popularity* of the items when choosing the items to expand. This way, not only we will expand with similar items but also with items that appear frequently in the user profiles, leading to a neighborhood large enough.

Int. J. Unc. Fuzz. Knowl. Based Syst. 2020.28:981-1002. Downloaded from www.worldscientific.com
by UNIVERSIDADE DA CORUNA on 06/13/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

We will model the popularity with a linear function. Particularly, popularity is calculatedq by dividing the number of ratings that an item received by the total number of users.

In summary, we will compute a total weight, $w_t$, considering both content and popularity similarities. It is described in Eq. (6), where $w_c$ is the content weight, $w_p$ is the popularity weight normalized with min-max normalization and $\alpha$ is a parameter that controls the contribution of each weight.

$$w_t = \alpha w_c + (1 - \alpha) w_p \,. \tag{6}$$

Unlike recommendation, where popularity can cause several problems like a decrease in coverage, novelty or diversity when only popular items are recommended, in Profile Expansion these problems are reduced.

### 3.5. *Rating estimation*

After selecting the items to expand, it is necessary to compute each item rating. Different approaches can be considered for determining the rating to assign to each item. Collaborative information can be taken into account: from the mean of the item to a weighted combination of the ratings given by the most similar neighbors of the active user/item, for example. But also content based information can be helpful. The contribution of each feature for determining a rating may differ from the contribution for selecting the items for the expansion. In this work, the combination of genre, cast and director has been selected, based on the conclusions extracted from the features selection analysis. Moreover, they presented the best results in accuracy. Apart from that, and unlike items recommendations where only good items are proposed to the user, a relevance threshold is not set for items in the expansion list.

Several alternatives have been proposed in order to understand whether content information is useful for rating estimations.

#### 3.5.1. *Collaborative proposals*

The collaborative approach we have used is the *mean* rating of the movie. It generally obtains acceptable values for the metrics usually employed in the prediction task, as Mean Absolute Error (MAE), for instance.

#### 3.5.2. *Content-based proposals*

We have also tested three content-based proposals: simple, weighted and z-score.

We can define a content rating profile for every content feature, where each pair is made of a level $l_i \in L^f$, and a value associated to it, $v_i$, as shown in Eq. (7). This value will mean differently according to the kind of proposal. We denote as $R_u^f$ the content rating profile of a user $u \in U$. This content rating profile is comprised of $n$

Int. J. Unc. Fuzz. Knowl. Based Syst. 2020.28:981-1002. Downloaded from www.worldscientific.com
by UNIVERSIDADE DA CORUNA on 06/13/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

feature level-value pairs, being $n$ the cardinality of $\left|L^f\right|$.

$$R_u^f = \left\langle \left(l_1, v_{u1}^f\right), \left(l_2, v_{u2}^f\right), \ldots, \left(l_n, v_{un}^f\right) \right\rangle . \tag{7}$$

Then, we obtain one value for every level of a feature. Each one of these values is based on the similarity between the content user profile and the content levels for the item we are computing the rating for. For completeness, we need to define the set of items that present a particular level for a feature in a content user profile, $I_{ul}^f$, as it is shown in Eq. (8).

$$I_{ul}^f = \{i \mid i \in I_u, l \in L_i^f\} . \tag{8}$$

The final rating will be computed as a linear combination of the partial ratings according to the different weights assigned to each feature. Following, we explain the different content approaches considered.

- Simple. The numerical values of the content rating profile are computed directly from the ratings of the collaborative profile as it is shown in Eq. (9).

$$v_{ul}^f = \frac{\sum_{i \in I_{ul}} r_{ui}}{\left|I_{ul}^f\right|} . \tag{9}$$

For every feature a prediction of the rating that would give the user to the item $i$ is computed (see Eq. (10)).

$$\hat{r}_{ui}^f = \frac{\sum_{l \in L_i^f \cap L_u^f} v_{ul}^f}{\left|I_{ul}^f\right|} . \tag{10}$$

- Weighted. The flaw that the simple proposal presents is that every rating predicted will be between the minimum and the maximum ratings in the user profile. This approach attempts to solve this problem and takes into consideration as much the mean rating of the profile items as the mean of the item we are computing the rating for. So, the numerical values of the content rating profile will not be ratings but differences between ratings and mean ratings. That is, it normalizes the ratings with respect to the mean as it is shown in Eq. (11).

$$v_{ul}^f = \frac{\sum_{i \in I_{ul}} r_{ui} - \bar{r}_i}{\left|I_{ul}^f\right|} . \tag{11}$$

The feature ratings are computed according to Eq. (12).

$$\hat{r}_{ui}^f = \bar{r}_i + \frac{\sum_{l \in L_i^f \cap L_u^f} v_{ul}^f}{\left|I_{ul}^f\right|} . \tag{12}$$

- Z-score. This approach not only takes into account the mean, but also the standard deviation. Eqs. (13) and (14) show how the computations are accomplished.

$$v_{ul}^f = \frac{\sum_{i \in I_{ul}} \frac{r_{ui} - \bar{r}_i}{\sigma_{r_i}}}{\left|I_{ul}^f\right|} , \tag{13}$$

$$\hat{r}_{ui}^{f} = \bar{r}_i + \sigma_{r_i} \frac{\sum_{l \in L_i^f \cap L_u^f} v_{ul}^{f}}{\left| I_{ul}^{f} \right|} . \tag{14}$$

Finally, for the sake of clarity, the notation that has appeared so far is summarized in Table 1.

Table 1. Summary of the notation.

| Symbol | Description |
|---|---|
| $I$ | Set of items |
| $U$ | Set of users |
| $r_{ui}$ | Rating made by a user $u$ on an item $i$ |
| $P_u^c$ | Collaborative user profile for a user $u$ |
| $P_u^{c'}$ | Expanded collaborative user profile for a user $u$ |
| $n$ | Number of ratings in a user profile |
| $l$ | Number of ratings added to a collaborative user profile |
| $I_u$ | Set of items rated by a user $u$ |
| $I_u^e$ | Set of items added to the user profile for $u$ |
| $F$ | Set of content features |
| $L^f$ | Set of levels that a feature $f$ can take |
| $L_i^f$ | Set of levels that a feature $f$ takes for an item $i$ |
| $P_u^f$ | Content user profile of a user $u$ for a content feature $f$ |
| $w_{ul}$ | Weight that a level $l$ takes in the content user profile for a user $u$ |
| $P_i^f$ | Content item profile of an item $i$ for a feature $f$ |
| $s^f(u,i)$ | Similarity between a user and an item profiles considering a feature $f$ |
| $w_c$ | Content weight |
| $w_p$ | Popularity weight |
| $w_t$ | Total weight |
| $\alpha$ | Parameter that controls the contribution of $w_c$ and $w_p$ in $w_t$ |
| $R_u^f$ | Content rating profile of a user $u$ |
| $I_{ul}^f$ | Set of items with a level $l$ for a feature $f$ in the content profile for $u$ |
| $v_{ul}^f$ | Value for level $l$ in content rating profile for a user $u$ and a feature $f$ |
| $\hat{r}_{ui}^f$ | Rating prediction for a user $u$ and an item $i$ considering a feature $f$ |

## 4. Methodology and Experiments

The experiments have been performed using Netflix[3] dataset. It contains over 100 million ratings from 480,189 users to 17,770 movies collected between October 1998 and December 2005. We downloaded the content information from the Internet

Int. J. Unc. Fuzz. Knowl. Based Syst. 2020.28:981-1002. Downloaded from www.worldscientific.com
by UNIVERSIDADE DA CORUNA on 06/13/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

Int. J. Unc. Fuzz. Knowl. Based Syst. 2020.28:981-1002. Downloaded from www.worldscientific.com
by UNIVERSIDADE DA CORUNA on 06/13/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

Movie Database (IMDb)[a] However, not all the movies in Netflix could be perfectly matched with a movie in IMDb, so Netflix was filtered. Finally, the dataset used consists of 8,362 items and 478,458 users, who have made 48,715,350 ratings between 1 and 5. Content values that only appeared once in the dataset were removed.

We have randomly selected 1000 users for evaluation purposes. The new user problem has been simulated using a Given-2 strategy.[5] To complete the training set, from the remaining users we have randomly chosen 10% of ratings in order to simulate a new system problem. This way, we can evaluate how the algorithms behave with little information about users and, therefore, very sparse data.

### 4.1. *Evaluation and metrics*

We compare the content-based expansion method with the item-global and the user-local Profile Expansion techniques. These two techniques showed the best results in our previous work.[9] Item-global algorithm searches for items similar to those that belong to the active user profile. An item-based kNN algorithm is used for this purpose. On the other hand, user-local expansion techniques take the items to expand from the active user neighbors. In particular, we use the most-rated approach, that is, the user profile will be expanded with the items most rated in the neighborhood. Moreover, a user-based kNN algorithm without Profile Expansion is also studied.

The cosine vector similarity for neighbor selection and a weighted approach for rating prediction were used in both the expansion (when needed) and the recommendation algorithms. The number of neighbors chosen is $k = 25$, selected after a previous evaluation as the optimal value.

When evaluating the recommendation algorithms, it is a tough task to isolate a concrete problem. Analyzing how the algorithm behaves in the different phases of the recommendation process (expansion and recommendation) can help to determine this concrete problem, especially when the system presents a new system situation.

#### 4.1.1. *Expansion*

A user profile is expanded by means of an expansion algorithm. It will return an item list with the additional items for the user profile. These items will be assigned a rating. So, Profile Expansion techniques provide recommender algorithms with an input where information about users has been increased. This information can present errors. In order to simplify the recognition of the source for these errors, we have identified two sub-phases in the expansion process:

- The items selection. For the Profile Expansion evaluation, the relevance or non relevance of an item is determined by its presence or its absence in the user

[a]http://www.imdb.com.

Int. J. Unc. Fuzz. Knowl. Based Syst. 2020.28:981-1002. Downloaded from www.worldscientific.com
by UNIVERSIDADE DA CORUNA on 06/13/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

profile, regardless of its rating. We aim at obtaining an expanded profile with so many items in coincidence with the user profile as possible. Due to these reasons, we have considered the metric Precision-at-$l$, being $l$ the size of the expansion proposed, to ensure the quality of the items selection.

- The ratings estimation. We also aim at determining how accurate is the assignment of ratings and how this accuracy evolves just as the expansion list varies. Several metrics can measure the precision accuracy. We will select MAE as it is a commonly used metric for this objective. We have not taken into account for the metric those items that did not belong to the user profile since the absolute error could not be computed for them.

#### 4.1.2. *Recommendation*

Together with the expansion algorithm, a recommender algorithm is employed. Using the expanded profile, it produces a recommendation list. As in the items selection, since we have opted for a find good items task, we have considered Precision-at-$n$ as the main metric in the recommendation. That is, the ratio of relevant items in the first $n$ positions of the recommendation list. Unlike in the expansion evaluation where everything was relevant, in the recommendation we only consider relevant those items whose rating is greater than or equal to 4 in the evaluation subset.

## 5. Results

Throughout this section the most significant results will be highlighted. In fact, we report on statistical significance using a standard one-sided t test with significance level $\alpha = 0.05$ when improving performance of the baseline models.

As already stated, the recommendation process involves two steps: the expansion and the recommendation. Next, we show the results obtained in both steps.

### 5.1. *Expansion*

The expansion algorithm is a content-based technique. Figure 1 shows precision and recall in expansion for different content features. As it was explained in Section 3.3 we considered two similarity measures: weighted and cosine. It can be seen how the curves for genre and country features are better than the curves for cast and director features. That is, genre and country features are repeated more frequently than the other two features in the user profile, as it had already been confirmed in our initial manual analysis.

Weighted similarity measure performs slightly better than cosine similarity measure, despite being simpler. In situations of an extreme new user problem, where the information is scarce and the user profile is not well defined, traditional functions like cosine vector similarity or Pearson correlation are too strict and lead to wrong results.[8] This produces that the weighted similarity obtains a better performance.

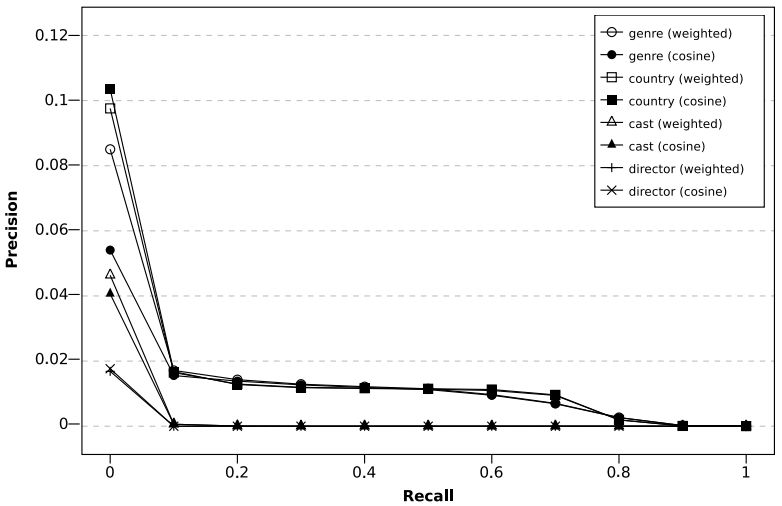



Fig. 1. P-R curves in expansion for different features.

Table 2. Precision in expansion for different Profile Expansion techniques, by the number of items added to the profile, $l$. Significant improvements over the item-global technique are highlighted.

| | $\alpha = 1$ | $\alpha = 0.75$ | $\alpha = 0$ | Item-global | User-local |
|---|---|---|---|---|---|
| $l = 5$ | 0.032 | 0.349 | **0.401** | 0.350 | 0.266 |
| $l = 10$ | 0.019 | 0.312 | **0.365** | 0.324 | 0.236 |
| $l = 15$ | 0.017 | 0.295 | **0.348** | 0.308 | 0.227 |
| $l = 20$ | 0.014 | 0.277 | **0.334** | 0.299 | 0.215 |
| $l = 50$ | 0.014 | 0.218 | **0.291** | 0.256 | 0.184 |

Table 2 shows the precision for the expansions computed with the content-based approaches and with the purely collaborative approaches. As it can be seen in the results, the precision decreases as the profile expansion size increases. The reason is that first items in the expansion list are given a higher weight, implying that the content-based expansion algorithms are working properly.

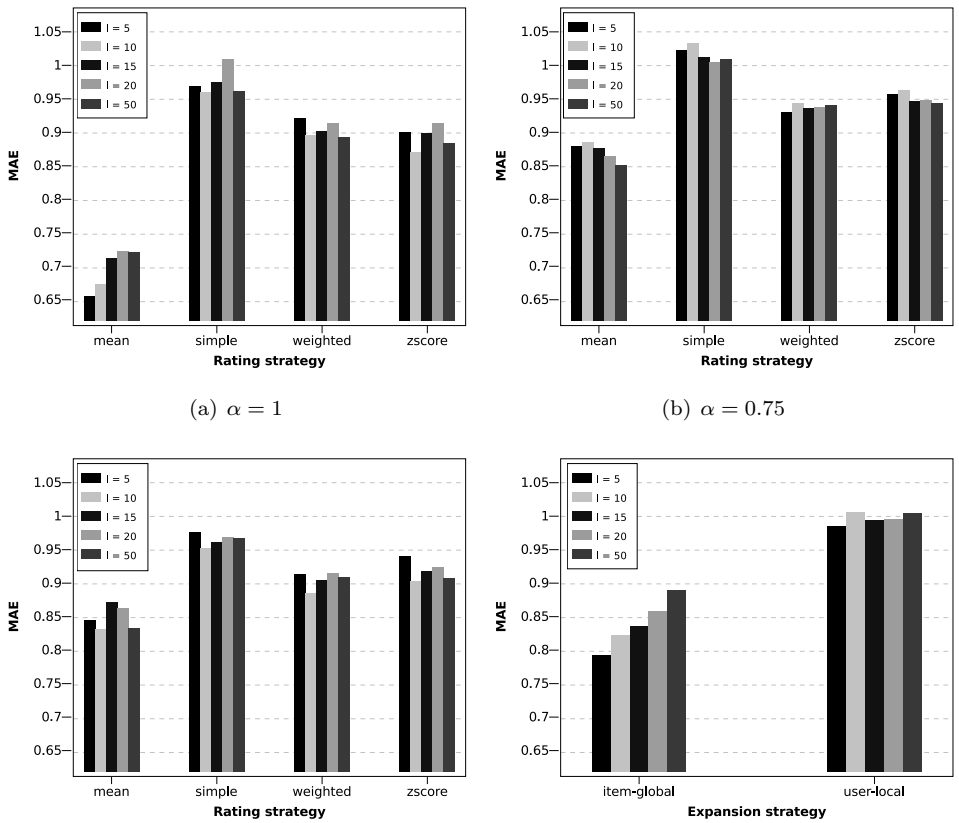
On the other hand, with Table 2 we can also study the behavior of the content-based approach according to $\alpha$, the parameter that controls the contribution of content and popularity weights. With $\alpha = 1$ (no popularity) our content approach presents very low precision. Actually, we are expanding with items similar to those in the user profile. However, these items can be rare in the dataset. So, the precision decreases, despite being expanding with similar items. In fact, when the popularity is taken into account, this precision is significatively increased. For example, with $\alpha = 0.75$ (that is, 75% content information and 25% popularity) the content information still has a big importance in the final weight of the item. But, with

Int. J. Unc. Fuzz. Knowl. Based Syst. 2020.28:981-1002. Downloaded from www.worldscientific.com
by UNIVERSIDADE DA CORUNA on 06/13/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

Int. J. Unc. Fuzz. Knowl. Based Syst. 2020.28:981-1002. Downloaded from www.worldscientific.com
by UNIVERSIDADE DA CORUNA on 06/13/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

popularity, all the ties (in weight) choosing the items to expand are solved in favor of the popular items. This way, precision increases, because these popular items are likely to appear in the user profiles. In particular, the p-values obtained for $\alpha$ equal to 0 after having analyzed the statistical significance of the results have been less than 2.0e-3. In this case, the baseline has been the item-global technique, and the algorithms have been compared according to the $l$ value used.

Regarding the accuracy in the rating prediction, in Figures 2(a), 2(b), and 2(c) it is shown that the best MAE in expansion is reached with mean strategy. Moreover, with $\alpha$ equal to 1 the mean strategy even improves the MAE obtained by the collaborative approaches in Figure 2(d), where item-global technique presents a good MAE (mainly, when $l$ is equal to 5). All the p-values computed are less than 2.0e-3 when comparing mean strategy (using $\alpha$ equal to 1) with item-global technique, independently of $l$. In addition to this, all the results with mean strategy and $l$ equal to 50 improve item-global algorithm.




(a) $\alpha = 1$

(b) $\alpha = 0.75$

(c) $\alpha = 0$

(d) Collaborative

Fig. 2. MAE in expansion for different Profile Expansion techniques, by the number of items added to the profile, $l$.

Int. J. Unc. Fuzz. Knowl. Based Syst. 2020.28:981-1002. Downloaded from www.worldscientific.com
by UNIVERSIDADE DA CORUNA on 06/13/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

Z-score and weighted strategies obtain better results than simple strategy. To understand the reason for the differences between the proposals, we analyze individually for every user in the evaluation set how the rating predictions were. We confirm that collaborative proposal and content proposals behave differently. With $\alpha$ equal to 0.75 and equal to 0, while mean strategy tends to underestimate the real ratings, the remaining strategies tend to overestimate. With $\alpha$ equal to 1, this behavior varies because only content information is considered to select the items to expand. In this case, mean and simple strategies overestimate, while weighted and z-score strategies underestimate. It is important to take into account that the best results with mean strategy are obtained when MAE values are based on very few items. As $\alpha$ is decreased and more items in the expansion also appear in the initial user profile, MAE results with mean strategy are closer to those obtained with collaborative techniques.

### 5.2. *Recommendation*

Once the expansion algorithm has been evaluated, the next step is to evaluate the recommender algorithm.

The results in Precision-at-5 obtained by the content-based approach with $\alpha$ equal to 1 are depicted in Table 3. These results improve slightly the precision obtained by the kNN algorithm without Profile Expansion (bottom row). Z-score strategy with 20 items added to the user profile is the best case. In particular, this rating strategy, apart from using content information, also uses some collaborative information. However, only the collaborative approaches (item-global and user-local) obtain better results. The reason is that not only both of them use collaborative information, but also the algorithms are more complex. Although collaborative information is important to determine the quality of an item, these results show that content information for rating estimation is useful, too. In fact, Tables 4 and 5 confirm that, since the content approaches present the highest precisions in these tables. However, although content information helps in rating estimation, the worst precisions are obtained when only content information is used for item selection.

Table 3. P@5 in recommendation for different Profile Expansion techniques, by the number of items added to the profile, $l$, with $\alpha = 1$.

| | Mean | Simple | Weighted | Z-score | Item-global | User-local |
|---|---|---|---|---|---|---|
| $l = 5$ | 0.103 | 0.100 | 0.096 | 0.096 | 0.181 | 0.161 |
| $l = 10$ | 0.102 | 0.099 | 0.101 | 0.100 | 0.165 | 0.161 |
| $l = 15$ | 0.101 | 0.101 | 0.104 | 0.103 | 0.149 | 0.154 |
| $l = 20$ | 0.107 | 0.101 | 0.109 | 0.112 | 0.133 | 0.138 |
| $l = 50$ | 0.089 | 0.076 | 0.090 | 0.091 | 0.110 | 0.120 |
| No PE | 0.100 | 0.100 | 0.100 | 0.100 | 0.100 | 0.100 |

Int. J. Unc. Fuzz. Knowl. Based Syst. 2020.28:981-1002. Downloaded from www.worldscientific.com
by UNIVERSIDADE DA CORUNA on 06/13/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

Table 4. P@5 in recommendation for different Profile Expansion techniques, by the number of items added to the profile, $l$, with $\alpha = 0.75$.

| | Mean | Simple | Weighted | Z-score |
|---|---|---|---|---|
| $l = 5$ | 0.178 | 0.172 | 0.178 | 0.178 |
| $l = 10$ | 0.176 | 0.182 | 0.183 | 0.184 |
| $l = 15$ | 0.167 | 0.164 | 0.172 | 0.170 |
| $l = 20$ | 0.129 | 0.125 | 0.126 | 0.128 |
| $l = 50$ | 0.117 | 0.118 | 0.121 | 0.121 |

Table 5. P@5 in recommendation for different Profile Expansion techniques, by the number of items added to the profile, $l$, with $\alpha = 0$. Significant improvements over the item-global technique (with $l = 5$) are highlighted.

| | Mean | Simple | Weighted | Z-score |
|---|---|---|---|---|
| $l = 5$ | **0.228** | **0.225** | **0.233** | **0.230** |
| $l = 10$ | 0.167 | 0.190 | 0.187 | 0.186 |
| $l = 15$ | 0.116 | 0.171 | 0.161 | 0.160 |
| $l = 20$ | 0.074 | 0.117 | 0.098 | 0.099 |
| $l = 50$ | 0.141 | 0.120 | 0.115 | 0.117 |

When two movies share the same values for all their content features, it is not possible to distinguish a good from a bad movie. Since content features cannot identify the quality, it is interesting to study how to complement content information with any other source of data. So, it would be useful to consider more data to help these techniques to determine whether to put a concrete item on the expansion list. Therefore, we decided to consider some collaborative information, and, more concretely, popularity.

In Section 3.4.1 popularity was mentioned as a way to improve the expansion in cold-start situations. Tables 4 and 5 confirm that. With $\alpha$ equal to 0.75 the results are even slightly better to those presented by the item-global algorithm. In fact, when $\alpha = 0$ and the profile is expanded with few items, the results are considerably better. The p-values obtained with $\alpha$ equal to 0 and $l$ equal to 5 when analyzing the statistical significance of the results have been less than 2.0e-5.

Figure 3 also shows the improvement obtained with popularity. In fact, the precision of the recommendation increases as the weight of the popularity is bigger. However, it is necessary to study the reason why this happens. Analyzing the dataset, we realized that there are several (popular) movies that have been rated by around fifty percent of the users. These popular items can be recommended, as long as their ratings reach the relevance rating threshold (popular item does not necessarily imply good item). If we only recommend these items, we obtain good results since it is very probable that the items will form part of the profile.

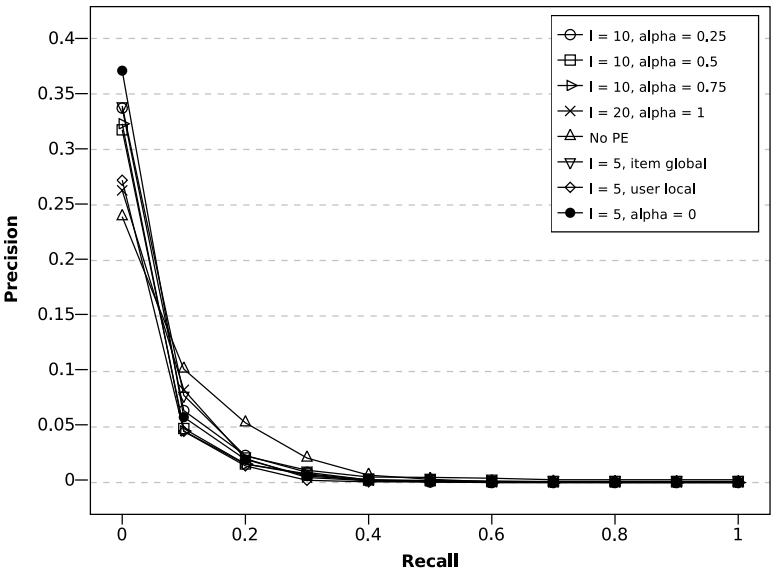



Fig. 3. P-R curves in recommendation for different algorithms with their optimal number of items added to the profile, l.

In fact, in the expansion when $\alpha$ is equal to zero (only popularity), the precision reaches the value of 0.60 when the recall is very low. Nevertheless, this expansion can cause that the recommender algorithm biases towards the most popular items. The results stop being personalized. This high precision is due to a limitation of the offline evaluation, because an item is only considered relevant if it appears in the user profile.

On the other hand, using $\alpha$ equal to one, the algorithm performs slightly better than the algorithm without Profile Expansion and very similarly to the user-local algorithm, when the recall is low. Actually, in the find good items task, only a small amount of relevant items will be shown to the active user. So, although the whole curve is important, we should pay attention specially to the first part of it, where it can be seen that the curve for the algorithm without expansion decreases more slowly than the others.

Finally, it is important to highlight another aspect in the differences between the behavior with $\alpha$ equal to 0 and with $\alpha$ equal to 1. With $\alpha$ equal to 0 the algorithm reaches a precision above 0.40 in expansion, that diminishes in recommendation (0.22). On the other hand, with $\alpha$ equal to 1 the precision is around 0.03 in expansion, but it increases up to 0.11 in recommendation. Moreover, the MAE obtained in expansion is slightly better when the algorithm does not use popularity, as it was mentioned previously. So, despite reaching high precision values in expansion, the items used for expansion with popularity are not personalized, and, in consequence, using the kNN algorithm the precision in recommendation decreases. On the other

Int. J. Unc. Fuzz. Knowl. Based Syst. 2020.28:981-1002. Downloaded from www.worldscientific.com
by UNIVERSIDADE DA CORUNA on 06/13/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

hand, although using only content information the precision in expansion is low, this precision increases in the recommendation, since the items are similar to those in the user profile. That is, several items could be considered relevant despite not being in the user profile. However, an offline evaluation does not allow to distinguish these cases.

## 6. Conclusions

Profile Expansion can be seen as a set of techniques to alleviate the new user problem. In this work we have shown how these techniques can be used to tackle the new system problem, too.

Particularly, we have proposed a content-based technique whose content features have been selected using a manual analysis. Several strategies have been used for computing the weights and the ratings for the expansion items. A weighted similarity has been employed for every feature. Moreover, content strategies for computing the ratings performed slightly better in recommendation than just the mean rating. Therefore, we have shown how content information can be useful not only to compute the expansion list, but also the ratings.

Apart from that, we have mentioned the popularity as an additional feature. We have shown that including this feature improves the results but it also leads to certain problems. First, it biases the results towards only the most popular items. Furthermore, the results are not personalized. Thus, it would be interesting to reach an agreement between popularity and content features according to the level of personalized results wanted in the system. Popularity could be suitable when the new system problem is extreme and the user profiles are too difficult to obtain.

## 7. Future Work

In the future this work could be completed in different ways. For example, in a system where not all the users are in a new user situation, it could be interesting to add different number of items to the expansion according to the user. Apart from that, a deeper analysis could be done about the differences between expanding only the active user and a global expansion.

Some issues related to kNN algorithms can also be studied, like similarity measures, not only in the recommender algorithm, but also in the expansion algorithms for different approaches. Furthermore, the number of neighbors can be modified according to their similarity with the active user.

Moreover, content information can be useful to filter the items obtained with collaborative techniques, like user-local. This way, only items sharing content information with the user profile would be part of the expansion.

Finally, in the future we plan to compare the solution proposed with other state of the art methods.

Int. J. Unc. Fuzz. Knowl. Based Syst. 2020.28:981-1002. Downloaded from www.worldscientific.com
by UNIVERSIDADE DA CORUNA on 06/13/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

**Acknowledgements**

This study was supported by the Ministry of Economy and Competitiveness of Spain and FEDER funds of the European Union (Project TIN2015-70648-P), by the Xunta de Galicia (Singular research center of Galicia, accreditation ED431G/01 2016-2019) and the European Union (European Regional Development Fund). This research was supported by the Ministry of Science and Innovation, Spain's National Research and Development Plan, through the PID2019-111388GB-I00 project.

Int. J. Unc. Fuzz. Knowl. Based Syst. 2020.28:981-1002. Downloaded from www.worldscientific.com
by UNIVERSIDADE DA CORUNA on 06/13/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

## References

1. C. C. Aggarwal, *Recommender Systems: The Textbook*, 1st edn. (Springer Publishing Company, Inc., 2016).
2. H. J. Ahn, A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem, *Inf. Sci.* **178** (2008) 37–51.
3. J. Bennett and S. Lanning, The netflix prize, in *Proc. KDD Cup and Workshop* (*KDDCup '07*) (ACM, San Jose, CA, USA, 2007), pp. 3–6.
4. J. Bobadilla, F. Ortega, A. Hernando, and J. Bernal, A collaborative filtering approach to mitigate the new user cold start problem, *Knowl.-Based Syst.* **26** (2012) 225–238.
5. J. S. Breese, D. Heckerman, and C. Kadie, Empirical analysis of predictive algorithms for collaborative filtering, in *Proc. Fourteenth Conf. Uncertainty in Artificial Intelligence* (*UAI'98*) (Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998), pp. 43–52.
6. R. Burke, Hybrid recommender systems: Survey and experiments, *User Modeling and User-Adapted Interaction* **12**(4) (2002) 331–370.
7. F. Cacheda, V. Carneiro, D. Fernández, and V. Formoso, Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems, *ACM Trans. Web* **5** (2011) 2:1–2:33.
8. F. Cacheda, V. Carneiro, D. Fernández, and V. Formoso, Improving k-nearest neighbors algorithms: practical application of dataset analysis, in *Proc. 20th ACM Int. Conf. Information and Knowledge Management* (*CIKM '11*) (ACM, New York, NY, USA, 2011), pp. 2253–2256.
9. V. Formoso, D. Fernández, F. Cacheda, and V. Carneiro, Using profile expansion techniques to alleviate the new user problem. *Inf. Process. Manage.* **49**(3) (2013) 659–672.
10. N. Golbandi, Y. Koren, and R. Lempel, Adaptive bootstrapping of recommender systems using decision trees, in *Proc. Fourth ACM Int. Conf. Web Search and Data Mining* (*WSDM '11*) (ACM, New York, NY, USA, 2011), pp. 595–604.
11. N. Good, J. B. Schafer, J. A. Konstan, A. Borchers, B. Sarwar, J. Herlocker, and J. Riedl, Combining collaborative filtering with personal agents for better recommendations, in *Proc. Sixteenth National Conf. Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conf. Innovative Applications of Artificial Intelligence* (*AAAI '99/IAAI '99*) (American Association for Artificial Intelligence, Menlo Park, CA, USA, 1999), pp. 439–446.
12. A. Gunawardana and C. Meek, Tied boltzmann machines for cold start recommendations, in *Proc. 2008 ACM Conf. Recommender Systems* (*RecSys '08*) (ACM, New York, NY, USA, 2008), pp. 19–26.
13. J. Herlocker, J. A. Konstan, and J. Riedl, An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms, *Inf. Retr.* **5**(4) (2002) 287–310.

Int. J. Unc. Fuzz. Knowl. Based Syst. 2020.28:981-1002. Downloaded from www.worldscientific.com
by UNIVERSIDADE DA CORUNA on 06/13/24. Re-use and distribution is strictly not permitted, except for Open Access articles.

14. J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, Evaluating collaborative filtering recommender systems, *ACM Trans. Inf. Syst.* **22**(1) (2004) 5–53.
15. X. N. Lam, T. Vu, T. D. Le, and A. D. Duong, Addressing cold-start problem in recommendation systems, in *Proc. 2nd Int. Conf. Ubiquitous Information Management and Communication* (*ICUIMC '08*) (ACM, New York, NY, USA, 2008), pp. 208–211.
16. C. W.-K. Leung, S. C.-F. Chan, and F.-L. Chung, Applying cross-level association rule mining to cold-start recommendations, in *Proc. 2007 IEEE/WIC/ACM Int. Conf. on Web Intelligence and Intelligent Agent Technology — Workshops* ( *WI-IATW '07*) (IEEE Computer Society, Washington, DC, USA, 2007), pp. 133–136.
17. P. Melville, R. J. Mooney, and R. Nagarajan, Content-boosted collaborative filtering for improved recommendations, in *8th National Conf. Artificial Intelligence* (American Association for Artificial Intelligence, Menlo Park, CA, USA, 2002), pp. 187–192.
18. A.-T. Nguyen, N. Denos, and C. Berrut, Improving new user recommendations with rule-based induction on cold user data, in *Proc. 2007 ACM Conf. Recommender Systems* (*RecSys '07*) (ACM, New York, NY, USA, 2007), pp. 121–128.
19. M. Papagelis, D. Plexousakis, and T. Kutsuras, Alleviating the sparsity problem of collaborative filtering using trust inferences, *Trust Management* **3477** (2005) 224–239.
20. S.-T. Park and W. Chu, Pairwise preference regression for cold-start recommendation, in *Proc. Third ACM Conf. Recommender Systems* (*RecSys '09*) (ACM, New York, NY, USA, 2009), pp. 21–28.
21. S.-T. Park, D. Pennock, O. Madani, N. Good, and D. DeCoste, Naive filterbots for robust cold-start recommendations, in *Proc. 12th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining* (*KDD '06*) (ACM, New York, NY, USA, 2006), pp. 699–705.
22. M. J. Pazzani, A framework for collaborative, content-based and demographic filtering, *Artif. Intell. Rev.* **13**(5-6) (1999) 393–408.
23. M. J. Pazzani and D. Billsus, Content-based recommendation systems, *The Adaptive Web* (Springer-Verlag, Berlin, Heidelberg, 2007), pp. 325–341.
24. B. Piccart, J. Struyf, and H. Blockeel, Alleviating the sparsity problem in collaborative filtering by using an adapted distance and a graph-based method, in *SIAM Int. Conf. Data Mining* (*SDM 2010*), 2010, pp. 189–198.
25. A. M. Rashid, I. Albert, D. Cosley, S. K. Lam, S. M. McNee, J. A. Konstan, and J. Riedl, Getting to know you: Learning new user preferences in recommender systems, in *Proc. 7th Int. Conf. Intelligent User Interfaces* (*IUI '02*) (Association for Computing Machinery, New York, NY, USA, 2002), pp. 127–134.
26. P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, Grouplens: an open architecture for collaborative filtering of netnews, in *Proc. 1994 ACM Conf. Computer Supported Cooperative Work* (*CSCW '94*) (ACM, New York, NY, USA, 1994), pp. 175–186.
27. B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, Item-based collaborative filtering recommendation algorithms, in *Proc. 10th Int. Conf. World Wide Web* (*WWW '01*) (ACM, New York, NY, USA, 2001), pp. 285–295.
28. A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, Methods and metrics for cold-start recommendations, in *Proc. 25th Annual Int. ACM SIGIR Conf. Research and Development in Information Retrieval* (*SIGIR '02*) (ACM, New York, NY, USA, 2002), pp. 253–260.
29. U. Shardanand, Social information filtering for music recommendation, Master's thesis, Massachussets Institute of Technology, September 1994.