Tuning metadata for better movie content-based recommendation systems

Márcio Soares · Paula Viana

Abstract The increasing number of television channels, on-demand services and online content, is expected to contribute to a better quality of experience for a costumer of such a service. However, the lack of efficient methods for finding the right content, adapted to personal interests, may lead to a progressive loss of clients. In such a scenario, recommendation systems are seen as a tool that can fill this gap and contribute to the loyalty of users. Multimedia content, namely films and television programmes are usually described using a set of metadata elements that include the title, a genre, the date of production, and the list of directors and actors. This paper provides a deep study on how the use of different metadata elements can contribute to increase the quality of the recommendations suggested. The analysis is conducted using Netflix and Movielens datasets and aspects such as the granularity of the descriptions, the accuracy metric used and the sparsity of the data are taken into account. Comparisons with collaborative approaches are also presented.

Keywords Recommendation algorithms · Collaborative · Content-based · Metadata

1 Introduction

The proliferation of video distribution services contributed to the gradual increase of the number of available channels and on demand content. Overloaded with too much information, many viewers systematically give up watching a program and tend to jump between different channels or always watching the same one. Traditional tools such as the Electronic Program Guides (EPGs) still have strong limitations as usually only provide extensive lists of programs,

M. Soares (✉) · P. Viana
INESC TEC (formerly INESC Porto), Campus da FEUP, Rua Dr. Roberto Frias, 378, 4200-465 Porto, Portugal
e-mail: mmsoares@inescporto.pt

P. Viana
ISEP/IPP–School of Engineering, Polytechnic Institute of Porto, Rua Dr. António Bernardino de Almeida, 431, 4200-072 Porto, Portugal
e-mail: paula.viana@inescporto.pt

which require the user to spend too much time searching for potentially interesting content. In this scenario, recommendation systems stand out as a possible solution to assist a watcher on the selection of the content that best fits his/her preferences.

In recent years, recommender systems (RS) have been used in quite different areas of application, including e–commerce (e.g. amazon), news and television services. In the context of multimedia services, Netflix,[1] Hulu[2] and even IMDB[3] have their own recommendation systems. These recommendation systems can take into account different aspects including information on the program itself, viewers' profiles and past shown interests or simply program popularity. Designing an accurate RS requires, as a first implementation decision, analysing the available data and deciding which parameters to be used given that they can affect the list of recommendations: can the recommendation of a film based on their actors be more effective than the recommended programs based on genre?; Programs described in more detail can provide more accurate recommendations?

Although work such as the one presented in [16, 18, 41] has already discussed the thematic of performance and accuracy of recommendations algorithms, their main focus is on collaborative algorithms. Evaluation of content-based algorithms and how metadata elements may affect the results of TV RS has been little explored.

In this paper we compare the performance of collaborative and content–based algorithms using different metadata elements. The impact of using, independently, information on the genre, list of actors or directors or a more complete set of elements, on the quality and accuracy of the recommendation is analysed. Netflix and Movielens datasets are used to evaluate the performance of the different approaches in respect to accuracy metrics that include Mean Average Error (MAE) and Precision.

Given that there is no universal metadata scheme and that granularity of the different elements varies from application to application, it is also important to measure how these differences can influence the quality of the results. Based on this assumption, the paper also provides results on how the use of different levels of sub-elements affects the quality of the RS.

Simulations have been run using different content-based algorithms and several partitions approaches of the main Netflix and Movielens datasets. The main results show that the conclusions of our work can be generalised and are not influenced by these issues.

2 Related work

Recommender systems are systems with the ability of providing suggestions or directing a person to a service, product or content, that has a potential of interest among a number of different alternatives [12, 39]. Examples can be found in different domains including book (Amazon[4]), music (Pandora,[5] Last.fm[6]), video (Youtube[7]) and product recommendation (eBay[8]). Although the first recommendation systems date from the late 70's, only in the early '90s the first commercial applications of this type of systems were deployed [2].

In the multimedia domain, Netflix, a commercial service providing access to movies and TV shows, presents predicted ratings for every displayed movie in order to assist the user deciding on the service to rent. Movilens[9] [32], a free, non-commercial, tool also provides services in the area of movie recommendations.

Television service providers have also demonstrated interest in enhancing their traditional programme guides and over the years several applications have emerged. The PTV project (Intelligent Personalized TV Guides) [15] was one of the first implementations in this area and is a reference to other solutions that came later.

A more sophisticated approach has been considered in [21], where not only historical information (e.g. ratings or gender preferences) is used but also information that can change in each access to the system (e.g. mood). The recommending mechanism is based on some user characteristics such as Activities, Interests, Moods, Experiences, and Demographic information (AIMED). Based on the idea that very often several people share the same living room and watch television at the same time, the work in [50] takes into account not a single user profile but handles a set of profiles in order to consider a group of people watching TV together.

Recommender systems are usually classified according to the approach that is used to find information that may fit the user's interest. The most popular recommendation approaches are [1]: (1) content-based filtering, (2) collaborative filtering and (3) hybrid.

Content-based systems try to recommend items that are similar to the ones that the user has demonstrate interest in the past. The similarity between the content is measured, in most cases, through the analysis of information that describes the contents, such as a film genre or the author of a book. The description of user's interests is obtained from information provided by the user himself, or alternatively by automatically creating his profile based on past actions.

Content-based systems can be built on text-based information about the items (keywords, title, genre, etc.) or on extracted features from the multimedia items. NewsWeeder [29], a content-based recommendation system for news on the Web, is an example of this type of RS: if the user demonstrates a preference for news related to sports, the system will recommend other news with sports content.

In collaborative algorithms, recommendations are based on the analysis of the similarity between users and performance is usually highly influenced by the number of active users in the system. GroupLens [38] was one of the first systems to adopt such algorithms. After reading netnews articles, users assign them numeric ratings which are later used to enable correlating users whose ratings are most similar and to predict how well users will like new articles, based on ratings from similar users.

Collaborative algorithms enjoyed a surge of interest with the Netflix Prize competition [8]. Among the most popular approaches, the nearest neighbour methods on either users, items or both, and methods using matrix factorization (MF) [27, 37] can be found. In fact, the Netflix Prize competition showed that advanced matrix factorization methods can be particularly helpful to improve the predictive accuracy of recommender systems. According to the recent progress in Collaborative Filtering (CF) techniques, current CF algorithms are primarily based on Neighbourhood Based Models (NBM) [10, 25] and MF with some variations [26, 52, 53], as the work presented in [51] where factors that can influence rating–such as mood, environment and time of day–are considered.

Content and Collaborative based algorithms are known to have advantages and disadvantages which Table 1 tries to summarize. By combining two or more recommendation techniques, hybrid approaches [12, 28] try to improve system performance by reducing the disadvantages of each technique used individually.

---

[9] www.movielens.umn.edu

Table 1 Comparative analysis of content and collaborative RS [3]

	Content-based algorithm	Collaborative filtering algorithm
Advantages	• Easy: it is easy to understand by user. • Cheap: resource consumption for computation is low. Light: non-personalized service is possible without the user preference profile.	• Diversity: it can deal with any kind of content. • Serendipity: it provides items with dissimilar content with those experienced in the past.
Disadvantages	• Shallow: only a very shallow analysis of certain kinds of content can be supplied. • Over-specialization: the user is restricted to seeing items similar to those already experienced.	• Sparsity: the lack of user preference data causes a performance decline, and makes it difficult to find nearest–neighbours for users with peculiar taste. • Scalability: increase of user preference data leads to a performance improvement, but much more resources are consumed.

The implementation of any of these approaches requires gathering information concerning the satisfaction of the users regarding the watched items. Two different approaches have been proposed: a classification range is defined and users are required to *explicitly* input their degree of enjoyment, or the system *implicitly* infers user's preferences by monitoring his activity while using the service. In our previous work [44] we developed a web based recommender system that helps the user navigating on broadcasted and online television content. In this system the user profile is constructed using information collected both explicitly and implicitly. Explicit information corresponds to classifications given to watched programs (1 to 5). The user can however decide voluntarily not to assign any classification to a program and, in such a case, the system automatically ascertains the amount of time that he remains watching a program. This time is converted into a quantitative classification ranging from 1 to 5 and assumed as the rating that the user would have given to that program.

Recent work tries also to consider additional information in order to improve RS results. Contextual information [19, 22, 23, 35, 40] and social relations [14, 24, 45, 48, 49], that can be obtained e.g. from sensors in mobile devices and from social networks, have been used as further inputs to RS algorithms. This has been applied for different purposes, including tag recommendation systems used to improve metadata describing resources in the Internet [14, 30, 47].

Although a lot of effort has been put on developing new algorithms and using additional information for RS, most of the work has been concentrated on CF and the way metadata elements affect the performance in CB approaches has been rarely explored. Some relevant work on this topic can be found in [31, 34, 36, 44]. Lommatzsch [31] compares different approaches for aggregating semantic movie knowledge and discusses the gain of combining different metadata attributes. In [36], the authors investigate the value of movie metadata compared to movie ratings in regards to predictive power. They show that by using collected movie metadata, prediction performance for the implemented methods is comparable to CF and can be used to predict preferences on new movies. The integration of semantic and emotion information along with the ratings is analysed in [34]. Performance of these CF models is tested using different combinations of the features spaces, including movie metadata, for different training datasets constructed from the original Movielens data. Symeonidis [44] developed a feature-weighted user profile to disclose the duality between users and features. The main objective is to exploit the correlation between users and semantic features that should reveal the real reason of users' rating behaviour. The developed approach is compared against well-known CF and CB, considering different metadata, and a hybrid algorithm with a real dataset.

The work presented in this paper adds new considerations in the area of CB approaches for movie recommendation and complements the previously published work. For validation purpose, we conducted simulations using two distinct datasets, namely, Movielens and Netflix. This allows result's generalizability, by confirming the achievements in independent samples, which was not provided in previously related work. Furthermore, given that metadata attributes can contain different levels of granularity (e.g. for the Netflix dataset, the movie genre is described in much more detail), and that results could be affected by this, rather than by the metadata element by itself, we also conduct different experiments that enable eliminating this hypothesis. The impact of the datasets sparsity is also deeply evaluated and, as a result, it's quite likely that the results presented in our study can be generalized to all datasets and metadata schemas within the field of movies and multimedia programs.

Finally, we also examine how many of top-N recommended items are the same for each of the studied cases. For example, suppose an algorithm that uses directors and another that uses genre to make recommendations and have the same performance. Will they both recommend the same items at top-10?

3 Recommendation approaches

Two recommenders, that implement collaborative-based and content-based approaches, were implemented based on the work described in [1, 42]. This enables comparing performances and to investigate if content-based systems can approach or even outperform the collaborative algorithm. Given that the aim of this work is to thoroughly analyse the use of metadata in CB algorithms, rather than comparing the performance of CF approaches, tests were run using a standard implementation based on the nearest neighbour for the CF algorithm and two implementations of CB systems, namely a nearest neighbour and a genre learning process. The next sections briefly describe each of the algorithms.

3.1 Nearest neighbour collaborative algorithm

The main objective of the user-to-user collaborative filtering technique is to estimate the rating that a user u would assign to a particular item i based on ratings assigned to that same item by other users having a profile similar to the user u under consideration. Being $R(u',i)$ the rating that user u' (similar to user u) gave to item i, the rating to be calculated, represented by $R(u,i)$, is given by:

$$R(u, i) = \frac{\sum_{u' \in N(u)} sim(u, u') \cdot R(u', i)}{\sum_{u' \in N(u)} |sim(u, u')|} \tag{1}$$

$N(u)$, the set of users considered similar to user u (user neighbours), can range from one to all users in the dataset. Limiting the size to some specific number (e.g. two) will determine how many similar users will be used in the computation of the rating prediction $R(u,i)$.

The similarity between two users, $sim(u,u')$, can be calculated using different metrics. In our implementation, the cosine similarity was used:

$$sim(u, u') = \frac{\sum_{i=1}^{n} R(u, i) R(u', i)}{\sqrt{\sum_{i=1}^{n} R(u, i)^2} \sqrt{\sum_{i=1}^{n} R(u', i)^2}} \tag{2}$$

3.2 Content-based algorithms

In order to try generalising the results for different CB approaches, simulations using two different algorithms presented in the literature were conducted. The next sections briefly describe each of the methods.

3.2.1 Nearest neighbour content-based

Content-based approaches estimate the similarity between items, using metadata information that describes them. In our work, different distance measures were used in the simulations, depending on the metadata element under consideration.

When comparing words' sequences where the order is not relevant, the cosine distance was used. One example is the analysis of the genre of a movie, where {Romance,Comedy} is considered alike to {Comedy,Romance}.

For other metadata attributes, as the list of actors or directors, in which the order may have some relevance, the Inverse Rank Measure was used. This metric calculates the similarity between two sequences taking into account the order of the elements and assigning different weights depending on the position of each element, according to the following expressions [4]:

$$
\begin{aligned}
\mathrm{N}^{(k_1,k_2)}(\sigma_1,\sigma_2) = & \sum_Z \left| \frac{1}{\sigma_1(i)} - \frac{1}{\sigma_2(i)} \right| \\
& + \sum_S \left| \frac{1}{\sigma_1(j)} - \frac{1}{k_2+1} \right| \\
& + \sum_T \left| \frac{1}{\sigma_2(j)} - \frac{1}{k_1+1} \right|
\end{aligned}
\tag{3}
$$

where Z is the set of the overlapping elements, $\sigma_1(i)$ is the rank of document i in the first set and $\sigma_2(i)$ is its rank in the second set (both ranks are defined for elements belonging to Z). In addition, S is the set of documents that appear in the first list but not in the second, while T is the set of elements that appear in the second list, but not in the first [5]; k_1 and k_2 are the number of elements of each set.

This measure is normalized as follows:

$$
M = 1 - \frac{\mathrm{N}^{(k_1,k_2)}}{\max \mathrm{N}^{(k_1,k_2)}}
\tag{4}
$$

where

$$
\max \mathrm{N}^{(k_1,k_2)} = \sum_{i=1}^{k_1} \left| \frac{1}{i} - \frac{1}{k_2+1} \right| + \sum_{i=1}^{k_2} \left| \frac{1}{i} - \frac{1}{k_1+1} \right|
\tag{5}
$$

For the example in Table 2, this metric results in a value of 0.31 for the pair Movie 1/Movie 2 and of 0.62 for Movie 1/Movie 3 as illustrated below for the first case:

$$
\begin{aligned}
\delta_1 &= (A,B,C,D) \\
\delta_2 &= (E,A,B) \\
\mathrm{N}^{(k_1,k_2)}(\sigma_1,\sigma_2) &= \left(\left| \frac{1}{1}-\frac{1}{2} \right| + \left| \frac{1}{2}-\frac{1}{3} \right| \right) + \left| \frac{1}{3}-\frac{1}{3+1} \right| + \left| \frac{1}{1}-\frac{1}{4+1} \right| \approx 1.54 \\
\max \mathrm{N}^{(k_1,k_2)} &= \left(\left| \frac{1}{1}-\frac{1}{3+1} \right| + \left| \frac{1}{2}-\frac{1}{3+1} \right| + \left| \frac{1}{3}-\frac{1}{3+1} \right| \right) + \left(\left| \frac{1}{1}-\frac{1}{4+1} \right| + \left| \frac{1}{2}-\frac{1}{4+1} \right| + \left| \frac{1}{2}-\frac{1}{4+1} \right| \right) \approx 2.23 \\
M &\approx 0.31
\end{aligned}
$$

Table 2 Example of movies and actors to exemplify Inverse Rank Measure

Movie	Actors
Movie 1	A, B, C, D
Movie 2	E, A, B
Movie 3	A, F, G, H

Although Movie 2 has two mutual actors with Movie 1 (A and B), while Movie 3 only shares actor A with Movie 1, this is in a more prominent position. Thus, Movie 3 is considered, by the Inverse Rank Measure, to be more similar to Movie 1 since they have the same main actor.

The final similarity between the items under analysis is obtained by weighting, with different factors (p_a), the individual values obtained for each of the attributes considered (such as genre, actors and directors) as presented in (6).

$$sim(i, i') = \frac{\sum_{a \in A} sim_a(i, i') * p_a}{\sum_{a \in A} p_a} \tag{6}$$

Being i' an item similar to item i (not yet rated), $sim(i,i')$ the similarity between items, and $R(u,i')$ the rating that the user u assigned to i', the rating that user u will give to item i, is given by:

$$R(u, i) = \frac{\sum_{i' \in N(i)} sim(i, i') \cdot R(u, i')}{\sum_{i' \in N(i)} |sim(i, i')|} \tag{7}$$

3.2.2 Genre learning technique

Items are often grouped into one or more categories such as genres or actors of movies and TV programs, or authors of books. In attribute-based prediction techniques, each attribute has an importance weight that can vary per user. Based on this importance weights' predictions can be generated. GenreLMS [42] learns how interested a user is in the genre, actors, directors or other attributes assigned to items and calculates a prediction using a linear function over different attributes (Eq. (8)).

$$P = w_0 + \sum_{a=0}^{n} w_a x_a \tag{8}$$

For each attribute a the algorithm learns a weight w_a indicating the relative importance of each attribute to the user, whereas w_0 is a constant value for the user. The extent (percentage) to which attribute a belongs to the item, is indicated by x_a, with:

$$\sum_{a}^{n} x_a = 1$$

Learning weights for each attribute takes place the moment a user rates an item. The learning algorithm uses the basic least Mean Square Method [33] (originating the name LMS

used for this technique). With LMS, each weight is updated using the difference between the actual rate R, provided by the user, and the predicted rate P:

$$w_{a_{new}} = w_a + \mu(R{-}P)x_a \tag{9}$$

here μ is a constant moderator determining the rate in which weights are updated.

4 Metrics for performance evaluation

Several metrics have been proposed to evaluate the performance of recommenders [7, 13, 17, 20, 46]. One of the most commonly used approaches is the Mean Absolute Error (MAE) that calculates the difference between the classification predicted by the system and the real rating assigned by the user to this same item, providing and estimation of the average error associated to recommendations (Eq. (10)).

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |p_i{-}a_i| \tag{10}$$

Although MAE is widely used due to its simplicity, it may not be appropriate for evaluating the quality of the top N recommendations [11, 20] as performance analysis should focus on the list of recommendations provided to the users (the top N items with the highest potential).

When not interested in the exact prediction value, but only in finding out if the active user will like or not the current item, classification accuracy metrics can be used. Widely used in binary classification systems, this metrics try to estimate whether the like/dislike estimated classification, matches the real user tastes rather than to analyse the exact value of the prediction. This approach may also be used in n-nary classification systems, by using an appropriate threshold that converts the results to a two level system. For instance, for a rating scale in the range 0 to 5, classifications above 4 could be considered as a like and below as a dislike.

To evaluate how well a recommendation list match the user's preferences, precision is commonly used [15, 20]. Items are first classified according to their real importance to the user and their place in the list of results provided by the system:

- *True Positive (TP)*–an interesting item is recommended to the user;
- *True Negative (TN)*–an uninteresting item is not recommended to the user;
- *False Negative (FN)*–an interesting item is not recommended to the user;
- *False Positive (FP)*–an uninteresting item is recommended to the user.

Two classes of recommendations–good and bad [20]–are then defined as illustrated in Table 3.

The Precision of a set of recommendations indicates the correct classification percentage and is given by:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{11}$$

Examples of application of precision are presented in [6, 9] or [41].

Table 3 Classification of the recommendation results

	Recommended	Not recommended
Actually good	TP	FN
Actually bad	FP	TN

5 Datasets

5.1 Datasets characterization and enhancement

Two datasets were used in the evaluation of the results presented in this paper: the Movielens 10 M and the Netflix Prize datasets.

Movielens 10 M[10] uses a rating scale in the range [1…5] and contains 10,000,054 ratings to 10,681 movies by 71,567 users. Each user rated at least 20 items and the average number of rating per user is 143.

Netflix dataset contains program ratings assigned by the costumers of the on-demand Internet streaming service. This dataset is composed of 100,480,507 ratings that 480,189 users gave to 17,770 movies. The rating scale adopted is also in the range of [1…5] and all the users rated at least 1 movie. The average number of ratings per user is 35.

These datasets have been previously used for comparing the performance of collaborative based recommendation systems. However, given that the purpose of the work presented in this papers is to analyse the impact of metadata in the construction of recommendation algorithms, additional information had to be added since these datasets only contain the classifications given by users to the programs, and do not have program description attributes (genres, actors and directors).

Figure 1 illustrates the process used for enhancing these two datasets. Metadata elements available from the Netflix[11] and IMDB[12] APIs were extracted and used to complement the initial available data. Given that two different data sources were used, some discrepancies can be noticed, for example, in the list and number of genres used to describe a film.

Table 4 presents the number of available genres, together with some examples that illustrate the differences between each of the datasets considered. Besides the considerable difference in the number of available genres (26 genres for describing programs in IMDB compared with 270 for Netflix), programs from Netflix are described with much more detail. For the class of horror programs, for example, IMDB provides only one possibility while Netflix allows horror films to be subcategorized using 3 genres.

Given these differences, a third dataset based on Netflix original ratings but enhanced with IMDB metadata was built. This enables conducting tests and comparing results that better illustrate the influence of the different metadata standards in the results of recommendation algorithms.

5.2 Datasets partitioning approaches

Since the datasets contain a large set of ratings (a few millions) a dataset resizing was done to reduce computational costs. This process took into account some factors directly related to the aspects under consideration in this work. Given that content-based filtering will be an

---

[10] http://movielens.umn.edu/
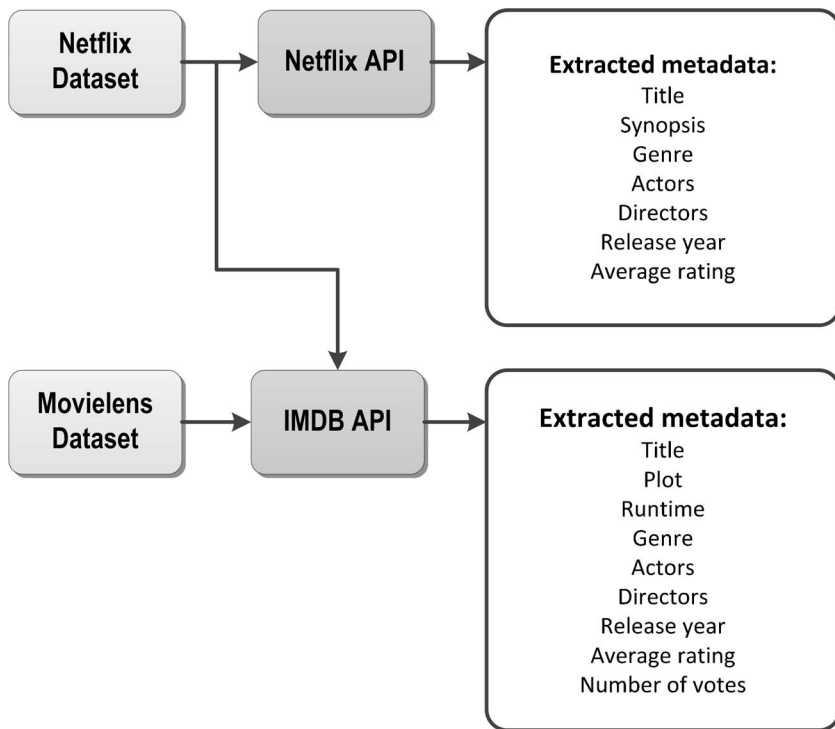[11] http://developer.netflix.com/
[12] http://www.imdbapi.com/

Fig. 1 Enhancement of Netflix and Movielens datasets with metadata

important component in this study, for each dataset, only programs whose actors and directors are present in at least two programs were selected. For example, if the actor Harrison Ford, the central actor of the movie Indiana Jones does not appear in at least two more films, Indiana Jones movie is eliminated.

After this initial filtering, and still with the objective of reducing the size of the dataset, two alternative approaches were used: (i) based on the number of ratings given by users to programs; (ii) according to a pre-defined test time used as the border between historical information and future ratings to be predicted.

	Dataset	
	IMDB	Netflix
Number of types of genres	26	270
Genre	Drama	Dramas
	Sci-Fi	Romantic dramas
	Action	Dramas based on a book
	Comedy	Sci-Fi & fantasy
	Adventure	Fantasy
	Horror	Horror movies
	Crime	Supernatural horror movies
	Family	Italian horror movies
	…	…

Table 4 Example of available genre classifier in IMDB and Netflix

Table 5 Characterisation of the sub-datasets constructed for the experiments (based on the number of ratings)

Dataset					
Name	Movielens	Sparsity	Name	Netflix	Sparsity
ml_25(1)	1,114 users, ~25 % of all the ratings	94 %	nflx_25(1)	972 users, ~25 % of all the ratings	89 %
ml_25(2)	872 users, ~25 % of all the ratings	93 %	nflx_25(2)	874 users, ~25 % of all the ratings	88 %
ml_50	1,025 users, ~50 % of all the ratings	87 %	nflx_50	1,197 users, ~50 % of all the ratings	82 %
ml_s	3,000 users	99 %	nflx_s	3,000 users	99 %

For approach (i), the top 3,000 users, that is, the ones that contributed with more ratings, were selected. These 3,000 users were further split into three groups (first three rows in Table 5), according to the percentage of ratings. To further study the impact of sparsity in the results, a highly sparse dataset was constructed considering the 3,000 users with less ratings in each of the dataset. Table 5 summarises the main characteristics of the sub-datasets used.

This approach for partition a dataset to allow reducing the costs of experimentation have been used in other published work [40]. However, it does not accurately represent the actual behaviour of a real world recommendation system: at a given time, recommendations should only use historical/known information from the past. Taking this aspect into account, an even cheaper alternative to split large datasets is to define a date to be considered as the border between past and future [40]: if a particular user rated a program on January 1, recommendation algorithms can only compare this program with programs that user rated before the day specified.

For our tests, the borderline was defined as the 1st of January 2005 and, as result, the original dataset was divided in two parts: (a) a list of programs for which available ratings date for after 2005 and for which we will try to make recommendations; and (b) programmes that were rated before 2005 and from which we will considerer the attributes (ratings and metadata) to make recommendations. The characterization of each of the datasets obtained is presented in Table 6.

6 Simulations and results

The main objective of the present work was to analyse the influence of some parameters in the recommendation process. For that, a set of simulations was conducted in order to enable:

• Comparing collaborative and content–based algorithms' performance;

Table 6 Characterisation of the sub-datasets constructed for the experiments (chronological division)

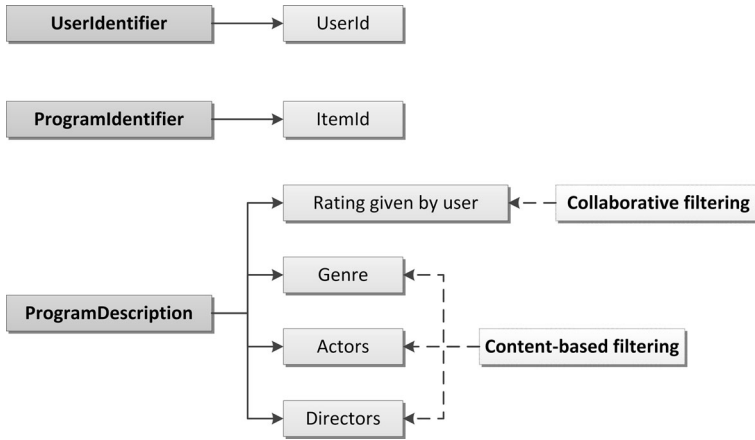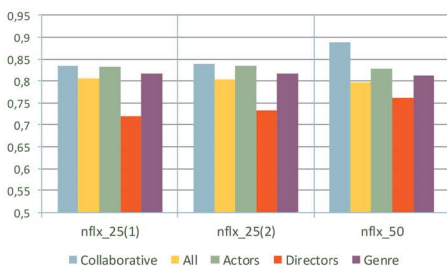|	Dataset	
	Movielens	Netflix
Date rating range	1995…2009	1999…2005
Time test	2005	2005
Number of users	20,278	18,899
Number of items	10,580	2,783
Number of ratings	3,430,566	8,448,565

Fig. 2 Metadata schema

- Checking how different metadata elements, used for computing the similarity between items in the content-based approach, influence the quality of obtained recommendations. Simulations using individually the genre, the list of actors or the list of directors were done. A final simulation considered the use of all the three elements together as well as all the possible two by two combinations;
- Analysing the use of different metrics on the evaluation of the performance of the algorithms (MAE and Precision);
- Checking how different metadata schemas having different granularity (in the case of this work, genre), influence the quality of obtained recommendations;
- The analysis of the sameness of the top-N predictions for the collaborative and content–based algorithms.

The tests were conducted for both the modified Netflix and Movielens datasets as described in Section 5. Figure 2 summarizes the metadata schema used in the simulations.

6.1 Impact of metadata attributes in algorithms' performance

To evaluate the performance of the collaborative and content-based algorithms two metrics were used: MAE and Precision. MAE measures how close the predicted results are to the
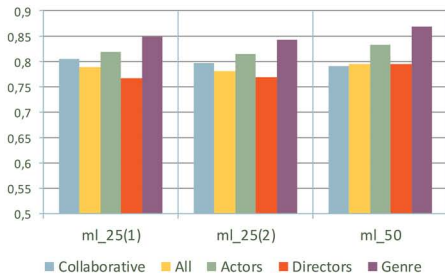

Fig. 3 MAE and precision for the Netflix dataset (partition based on the number of ratings)
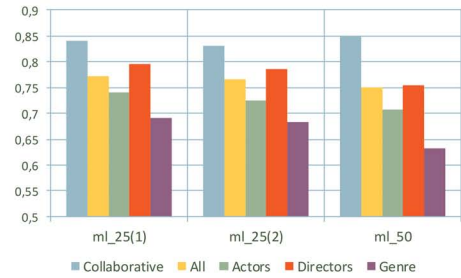
Fig. 4 MAE and precision for the Movielens dataset (partition based on the number of ratings)

user's real ratings and considers all the predictions made by the system for each user, while precision measures the ability of the algorithms to only recommend what is relevant. Since metrics as Precision are optimised to evaluate the top-N recommendation list, the top 10 recommendations were considered for calculating this evaluation parameter.
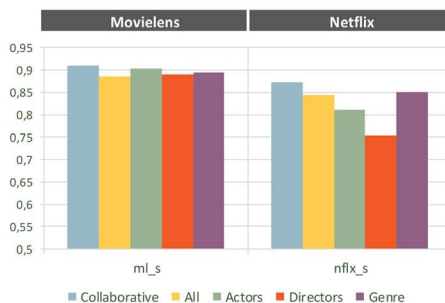
Figures 3 and 4 depict the MAE and Precision results on the Netflix and Movielens sub-dataset partitioned according to approach (i) described in Section 5.2.

One of the conclusions that can be drawn from the results presented is that using the information on directors rather than the, commonly considered relevant, genre and list of actors, enables a better performance of content-based algorithms. Moreover, the impact on using a more complete set of metadata (All: Actors+Directors+Genre) does not contribute to decrease MAE and may only slightly contribute to increase the precision. This observation is relevant since by using only one metadata element (the directors) the algorithm becomes less computationally expensive.

When comparing the two approaches for evaluating the performance of the algorithms, it is important to notice that the metric used can have some influence when comparing content-based and collaborative filtering: while the collaborative algorithm achieves a better performance for the Top-10 precision, the prediction error based on MAE is smaller for the content-director-based approach.

One of the aspects that this study proposed to analyse was how the sparsity of the dataset (small number of ratings or large number of new items) would affect the results. Results in Figs. 3 and 4 show similar behaviour independently of the dataset partition considered.

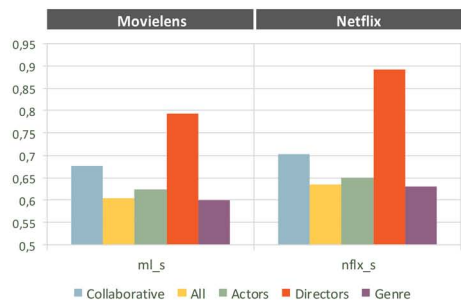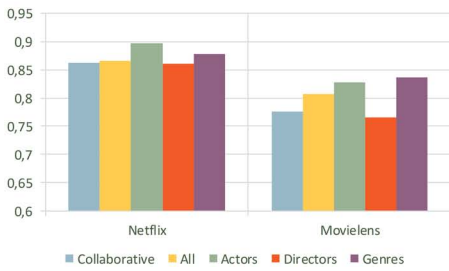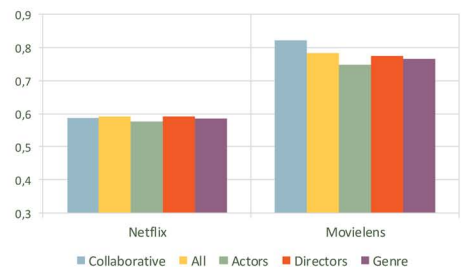**Fig. 5** MAE and precision for the Movielens and Netflix sparse sub-dataset

Fig. 6 MAE and precision for Netflix and Movielens datasets (chronological partition)

However, given that these sub-datasets were constructed based on the users with more rated movies, which makes sub-datasets quite similar, tests using a sub-dataset with high sparsity (ml_s and nflx_s datasets in Table 5) were performed. As expected and illustrated in Fig. 5, some decrease of the collaborative algorithm's performance for both tested datasets is noticed. The most relevant conclusion is, however, that CB approach based on the directors' information significantly outperforms the CF algorithm and is not negatively influenced by the sparsity. This conclusion provides important guidelines to deal with the cold start problem and to enable new items to be recommended.

Considering the Netflix and Movielens sub-dataset partitioned according to approach (ii), the results obtained were close to the ones achieved before as shown in Fig. 6. This confirms that the main conclusions are not influenced by the way the test dataset is constructed (either using the entire user profile history or just knowing his past preferences).
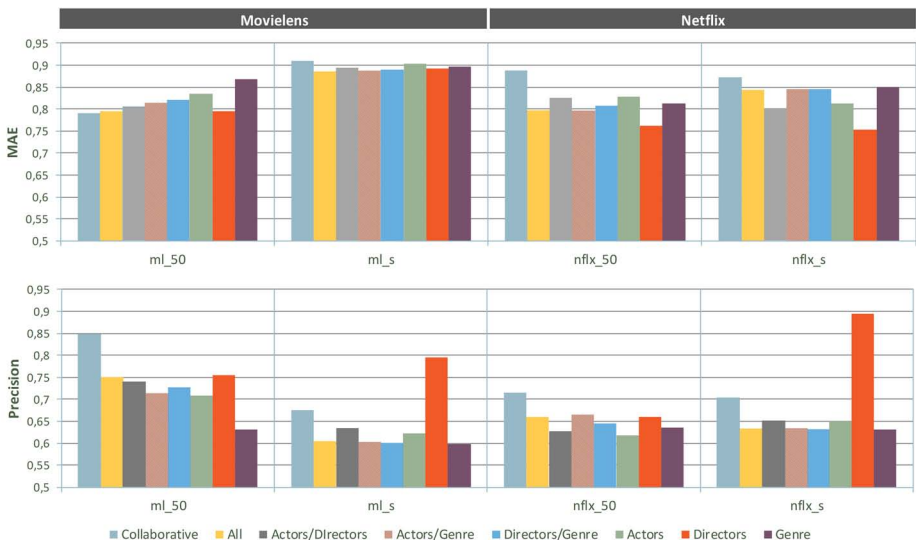

Fig. 7 Comparative study of all the possible combinations among metadata elements
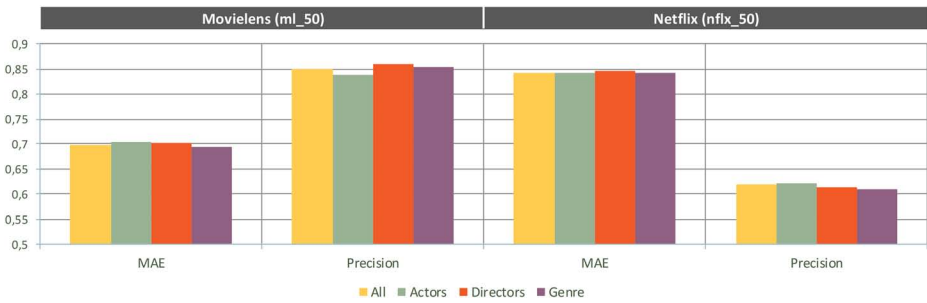
MAE & Precision - Movielens & Netflix

Fig. 8 MAE and precision for Movielens and Netflix sub-datasets for the GenreLMS algorithm

Figure 7 provides additional results that enable evaluating the impact of all the possible combinations of two metadata elements. The results show that, by combining metadata information, performance can be improved for the less relevant attributes (e.g. genre). However, directors individually still outperform all the combinations. This conclusion is rather important due to the fact that using additional information results in more computational costs that are not converted into performance gain.

In order to further study the influence of the different metadata elements in content-based RS, additional tests using a different algorithm (GenreLMS) were made. For the results depicted in (Fig. 8) we considered an optimal update moderator value (µ) calculated for each of the simulations.

The immediate conclusion that could be drawn from the pictures is that the GenreLMS algorithm is not noticeably affected by the metadata used. This might be regarded as a different behaviour when compared to the previous results. However, given that the execution time of this algorithm is highly affected by the number of attributes used, these results should be carefully analysed. Considering the Netflix results, one can conclude that by using all the metadata attributes, the results are slightly improved. However, given that this improvement is achieved at the expense of a great execution time and that the difference in performance towards using the directors' information is almost unnoticeable, the best attribute to be used can still be considered the director. As a similar analysis can be done for the Movielens dataset, the final conclusion is still that the directors attribute provide the best information to be used in CB algorithms.

6.2 Impact of the genre granularity on algorithms' performance

Although quite a lot of effort has been put on the standardization of a multimedia description schema, different solutions coming from different organizations and

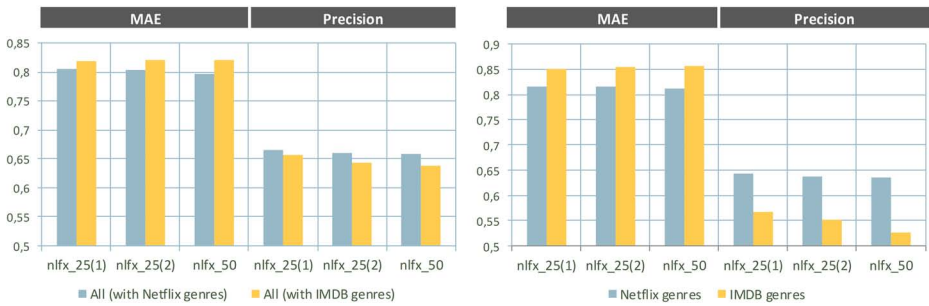| **Table 7** List of genres used by Netflix and IMDB for the film "alien vs predator"	Movie	IMDB genre	Netflix genre
	Alien vs. predator	Action Adventure Sci-Fi	Action & adventure Horror movies Sci-Fi & fantasy Monster movies Action Sci-Fi & fantasy Alien Sci-Fi Sci-Fi horror

Fig. 9 MAE and precision comparing algorithm's performance using original Netflix genres against IMDB genres

having different levels of details were published. Not only public metadata schemas like TV-Anytime, MPEG-7 or SMPTE are available, but private solutions custom-ized to fulfil individual requirements such as the ones used by Netflix and IMDB are also used.

The list of available genres or the use of a main genre and a set of sub-genres illustrates how differently a programme can be described. Table 7 exemplifies how the same content is described in IMDB and Netflix.

In order to analyse how this difference would affect the results, a new dataset was assembled: for the Netflix dataset, genre has been replaced by the genre of the IMDB database. New tests were performed for the nearest neighbour content-based algorithm using all metadata attributes and using the gender only.

Figures 9 and 10 compare the results obtained for the original and modified Netflix datasets. It is clear that the increase in the number of genres used to describe content enables better results, showing how the granularity of the metadata schema can influence the quality of the recommendation in content-based approaches.

6.3 Sameness of the top-N results

Besides the influence of metadata on the quality of the recommendations and how the metric used to compare algorithms can provide different views of the problem,
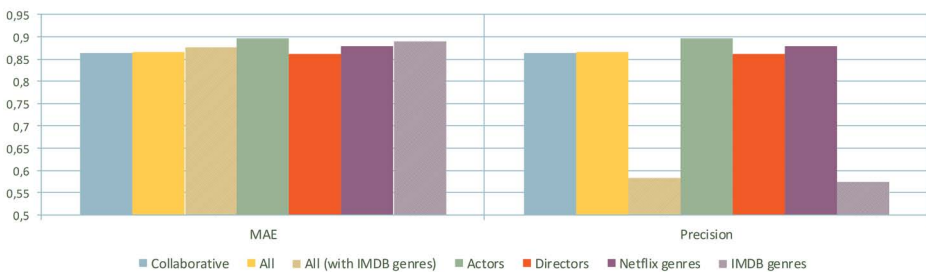

Fig. 10 MAE and precision comparing algorithm's performance using original Netflix genres against IMDB genres (chronological partition)

Table 8 Sameness of the top 10 recommendations for Netflix dataset

	Collaborative	Content			
		All	Actors	Directors	Genre
		Top 10			
Collaborative		1,08	0,84	1,10	0,91
All		1,08		2,27	2,92	3,67
Actors		0,84	2,27		1,55	1,62
Directors		1,10	2,92	1,55		1,44
Genre		0,91	3,67	1,62	1,44	

another question arises. Do the approaches that have better performance, recommend the same programs? For example, given that content-based algorithm performs well both when using all the metadata elements and the directors individually, will they recommend the same programs?

Tables 8 and 9 compare the collaborative and content–based approaches by examining how many of the top-10 and top-20 recommended items are the same for one of the case studies presented in this paper (Netflix dataset, partitioned by the number of ratings per user–subdataset 50 %). The colours in the tables point out the greater similarities achieved for each

Table 9 Sameness of the top 20 recommendations for Netflix dataset

	Collaborative	Content			
		All	Actors	Directors	Genre
		Top 20			
Collaborative		2,68	2,16	3,08	2,34
All		2,68		4,99	6,04	8,70
Actors		2,16	4,99		3,37	3,46
Directors		3,08	6,04	3,37		3,26
Genre		2,34	8,70	3,46	3,26	

Table 10 Sameness of the top 10 recommendations for Movielens dataset

	Collaborative	Content			
		All	Actors	Directors	Genre
		Top 10			
Collaborative		0,31	0,35	0,24	0,25
Content · All	0,31		1,17	1,37	0,80
Content · Actors	0,35	1,17		0,57	0,26
Content · Directors	0,24	1,37	0,57		0,23
Content · Genre	0,25	0,80	0,26	0,23	

of the pairs of approaches (e.g. the greater similarity obtained for the collaborative method was obtained with the Directors' based approach–1.91; thus, the intersection of the collaborative line with the director's column is marked in blue). Tables 10 and 11 present the same analysis for the Movielens dataset.

From the tables, it is clear that different approaches recommend different programs. For the top-20 list, for example, recommendations based on all metadata only share 25 % of the items recommended based on the actors. These results may be relevant in the implementation of hybrid recommenders that list together the results obtained from two or more approaches.

Table 11 Sameness of the top 20 recommendations for Movielens dataset

	Collaborative	Content			
		All	Actors	Directors	Genre
		Top 20			
Collaborative		1,07	1,04	0,98	0,81
Content · All	1,07		3,25	4,02	2,31
Content · Actors	1,04	3,25		1,68	0,8
Content · Directors	0,98	4,02	1,68		0,23
Content · Genre	0,81	2,31	0,8	0,23	

It is also interesting to notice that although the approaches based on all the metadata and on the directors only showed similar performances, they share only 4.02 of the items recommended for the Top20 list of the Movielens dataset (Table 11).

7 Conclusions

The work presented in this paper provides a deep evaluation on how content-based recommendation algorithms can be influenced by the metadata information used in the domain of movies and television programs description. Different datasets and metrics were used in order to validate the results and to guarantee that they were not influenced by the dataset used rather than by the metadata itself.

The results presented in this paper demonstrate that although the collaborative algorithm usually performs better, improvements can be achieved in the content-based approach by using the adequate metadata information, making the results quite similar. This may contribute to make the content-based algorithm a good alternative when e.g. computational cost is too high to implement the collaborative method or the information available in the service is sparse and does not enable finding the best neighbours.

The combination of different metadata elements provides usually better results when compared with metadata used separately. In addition, the greater the number of metadata attributes used in combination, the better is the performance. However, an exception occurs in the case of the directors used individually and this finding may help decrease computation time while maintaining the same quality.

The better performance achieved by using information on the directors may be likely explained by the fact that users do not usually guide their interests by generic programs attributes (such as genre) but mainly by a quality perception that is not explicit in the descriptive content. This may be intuitively read from the dataset contents where, for example, films directed by James Cameran always have good ratings (above 7) while actors, even with a good reputation, may participate in movies with fairly inconsistent ratings. This leads to the conclusion that a film director can provide specific information on the potential quality of a movie that cannot be described with another set of metadata elements. The importance of this attribute is even clearer when looking at findings resulting from very sparse datasets: although all the other algorithms suffered a significant decrease in performance, using the directors metadata enable still guaranteeing good recommendations. This conclusion is relevant when dealing with recent systems with small history information.

Additionally, the results show that the granularity used (e.g. in the genre) has an impact in the quality of the recommendations. This observation can help media asset managers on choosing a more adequate content description schema.

Furthermore it was noticed that the list of items originated by each of the different approaches, have little in common. This fact illustrates the potential of using hybrid approaches to guarantee the diversity of the recommendations.

Future work includes the analysis of other perspectives on the evaluation of recommendation lists such as the novelty and diversity of the results.

References

1. Adomavicius G, Kwon Y (2007) New recommendation techniques for multicriteria rating systems. IEEE Intell Syst 22(3):48–55
2. Adomavicius G, Tuzhilin A (2005) Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans Knowl Data Eng 17(3):734–749
3. Ahn S, Shi CK (2008) Exploring movie recommendation system using cultural metadata. In: Proceedings of the 2008 International Conference on Cyberworlds, pp 431–438
4. Bar-Ilan J, Keenoy K, Yaari E, Levene M (2007) User rankings of search results. J Am Soc Inf Sci Technol 58(9):1254–1266
5. Bar-Ilan J, Mat-Hassan M, Levene M (2006) Methods for comparing rankings of search engine results. Comput Netw 50(10):1448–1463
6. Basu C, Hirsh H, Cohen W (1998) Recommendation as classification: using social and content-based information in recommendation. In: Proceedings of the Fifteenth National Conference on Artificial Intelligence, pp 714–720
7. Bellogin A, Castells P, Cantador I (2011) Precision-oriented evaluation of recommender systems: an algorithmic comparison. In: Proceedings of the fifth ACM conference on Recommender systems, pp 333–336
8. Bennett J, Lanning S (2007) The Netflix prize. In: Proceedings of KDD Cup and Workshop
9. Billsus D, Pazzani MJ (1998) Learning collaborative information filters. In: Proceedings of the 15th International Conference on Machine Learning, pp 46–54
10. Bobadill J, Serradilla F, Bernal J (2010) A new collaborative filtering metric that improves the behavior of recommender systems. Knowl-Based Syst 23(6):520–528. doi:10.1016/j.knosys.2010.03.009
11. Breese JS, Heckerman D, Kadie C (1998) Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence. Morgan Kaufmann, San Francisco, pp 43–52
12. Burke R (2007) Hybrid web recommender systems. In: The adaptive web. Springer, Berlin, pp 377–408
13. Castells P, Vargas S, Wang J (2011) Novelty and diversity metrics for recommender systems: choice, discovery and relevance. International Workshop on Diversity in Document Retrieval at the 33rd European Conference on Information Retrieval
14. Chen K, Chen T, Zheng G, Jin O (2012) Collaborative personalized tweet recommendation. In: Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval, pp 661–670
15. Cotter P, Smith B (2000) PTV: intelligent personalised TV guides. In: Proceedings of the 12th Innovative Applications of Artificial Intelligence Conference, pp 957–964
16. Cremonesi P, Koren Y, Turrin R (2010) Performance of recommender algorithms on top-n recommendation tasks. In: Proceedings of the fourth ACM conference on Recommender systems, pp. 39–46
17. Cremonesi P, Turrin R, Lentini E, Matteucci M (2008) An evaluation methodology for collaborative recommender systems. In: Proceedings of the 2008 International Conference on Automated solutions for Cross Media Content and Multi-channel Distribution, pp 224–231
18. Fouss F, Serens M (2008) Evaluating performance of recommender systems: an experimental comparison. In: Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, 1: pp 735–738
19. Gorgoglione M, Panniello U, Tuzhilin A (2011) The effect of context-aware recommendations on customer purchasing behavior and trust. In: Proceedings of the fifth ACM conference on Recommender systems, pp 85–92
20. Herlocker J, Konstan J, Terveen L, Riedl J (2004) Evaluating collaborative filtering recommender systems. ACM Trans Inf Syst 22(1):5–53
21. Hsu S H, Wen M, Lin H, Lee C (2007) AIMED-A personalized TV recommendation system. In: Proceedings of EuroITV, pp 166–174
22. Iaquinta L, Gemmis M, Lops P, Semeraro G, Filannino M, Molino P (2008) Introducing serendipity in a content-based recommender system. In: Proceedings of the 2008 8th International Conference on Hybrid Intelligent Systems, pp 168–173
23. Jancsary J, Neubarth F, Trost H (2010) Towards context-aware personalization and a broad perspective on the semantics of news articles. In: Proceedings of the fourth ACM conference on Recommender systems, pp 289–292

24. Jiang M, Cui P, Liu R, Yang Q, Wang F (2012) Social contextual recommendation. In: Proceedings of the 21st ACM international conference on Information and knowledge management, pp 45–54
25. Koren Y (2008) Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, pp 426–434
26. Koren Y (2010) Collaborative filtering with temporal dynamics. Commun ACM 53(4):89–97. doi:10.1145/1721654.1721677
27. Koren Y, Bell R, Volinsky C (2009) Matrix factorization techniques for recommender systems. Computer 42(8):30–37
28. Lampropoulos AS, Lampropoulos LS, TsihrintzisA GA (2011) Cascade-hybrid music recommender system for mobile services based on musical genre classification and personality diagnosis. In Multimedia Tools and Applications
29. Lang K (1995) NewsWeeder: learning to filter netnews. In: Proceedings of the 12th International Conference on Machine Learning, California, pp 331–339
30. Lipczak M, Milios E (2010) Learning in efficient tag recommendation. In: Proceedings of the fourth ACM conference on Recommender systems, pp 167–174
31. Lommatzsch A, Kille B, Albayrak S (2013) A framework for learning and analyzing hybrid recommenders based on heterogeneous semantic data. In: Proceedings of the 10th Conference on Open Research Areas in Information Retrieval, pp 137–140
32. Miller B N, Albert I, Lam SK, Konstan JA, Riedl J (2003) Movielens unplugged: experiences with an occasionally connected recommender system. In: Proceedings of the 8th international conference on Intelligent user interfaces, pp 263–266
33. Mitchell TM (1997) Machine learning. McGraw-Hill, Singapore
34. Moshfeghi Y, Piwowarski B, Jose J (2011) Handling data sparsity in collaborative filtering using emotion and semantic based features. In: Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval, pp 625–634
35. Pazzani M, Billsus D (1997) Learning and revising user profiles: the identification of interesting web sites. Mach Learn 27:313–331
36. Pilászy I, Tikk D (2009) Recommending new movies: even a few ratings are more valuable than metadata. In: Proceedings of the third ACM conference on Recommender systems, pp 93–100
37. Rendle S, Gantner Z (2011) Fast context-aware recommendations with factorization machines. In: Proceedings of the 34th International ACM SIGIR Conference on Research and Development, pp 635–644
38. Resnick P, Iacovou N, Suchak M, Bergstrom P, Riedl J (1994) GroupLens: an open architecture for collaborative filtering of netnews. In: Proceedings of the 1994 ACM conference on Computer supported cooperative work, pp 175–186
39. Resnick P, Varian HR (1997) Recommender systems. Commun ACM 40(3):56–58
40. Ricci F, Rokach L, Shapira B, Kantor P, Paul B (2010) Recommender systems handbook. Springer, London
41. Sarwar B, Karypis G, Konstan K, Riedl J (2000) Analysis of recommendation algorithms for e–commerce. In: Proceedings of the 2nd ACM conference on Electronic commerce, pp 158–167
42. Setten MV (2002) Experiments with a recommendation technique that learns category interests. ICWI, pp 2712–2718
43. Soares M, Viana P (2014) TV recommendation and personalization systems: Integrating broadcast and video on demand services. Advances in Electrical and Computer Engineering 14(1):115–120. doi:10.4316/AECE.2014.01018
44. Symeonidis P, Nanopoulos A, Manolopoulos Y (2007) Feature-weighted user model for recommender systems. User Modeling 2007, Lecture Notes in Computer Science, 4511:97–106, Springer
45. Tang J, Hu X, Gao H, Liu H (2013) Exploiting local and global social context for recommendation. In: Proceedings of the Twenty-Third international joint conference on Artificial Intelligence, pp 2712–2718
46. Vargas S, Castells P (2011) Rank and relevance in novelty and diversity metrics for recommender systems. In: Proceedings of the fifth ACM conference on Recommender systems, pp 109–116
47. Wei C, Hsu W, Lee M (2011) A unified framework for recommendations based on quaternary semantic analysis. In: Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval, pp 661–670
48. Yang B, Lei Y, Liu D, Liu J (2013) Social collaborative filtering by trust. In: Proceedings of the Twenty-Third international joint conference on Artificial Intelligence, pp 2747–2753
49. Yang X, Steck H, Guo Y, Liu Y (2012) On top-k recommendation using social networks. In: Proceedings of the sixth ACM conference on Recommender systems, pp 67–74
50. Yu Z, Zhou X, Hao Y, Gu J (2006) TV program recommendation for multiple viewers based on user profile merging. User Model User-Adap Inter 16(1):62–82

51. Zhong E, Fan W, Yang Q (2012) Contextual collaborative filtering via hierarchical matrix factorization. SDM, pp 744–755
52. Zhou K, Yang S, Zha H (2011) Functional matrix factorizations for cold-start recommendation. In: Proceedings of the 34th international ACMSIGIR conference on Research and development in Information Retrieval, pp 315–324
53. Zhuang Y, Chin WS, Juan Y-C, Lin C-J (2013) A fast parallel SGD for matrix factorization in shared memory systems. In: Proceedings of the 7th ACM conference on Recommender systems–RecSys 13, pp 249–256

Márcio Soares received his Masters degrees in Computer & Telecommunications Engineering from Polytechnic Institute of Porto. In 2011 he joined the Telecommunications and Multimedia unit of INESC TEC, where he is currently a researcher. His research activities include recommendation systems, recommendation algorithms, multimedia content and intelligent metadata.

Paula Viana is a Professor at the Polytechnic Institute of Porto, where she is responsible for courses in the area of multimedia and telecommunications. She is also a senior researcher at INESC TEC in the telecommunications and multimedia unit. She obtained her PhD (2008) from University of Porto in the area of multimedia management. She has been responsible for the participation of the INESC TEC team in several national and European projects. Author of several publications, she is also an active reviewer both for journals and conferences as well as an expert for the evaluation of European research proposals and projects. Her main research activities and interests are in the field of networked audiovisual systems, content management and semantic web.