User Modeling and User-Adapted Interaction manuscript No.
(will be inserted by the editor)

Movie Genome: Alleviating New Item Cold Start in Movie Recommendation

Yashar Deldjoo, Maurizio Ferrari Dacrema, Mihai Gabriel Constantin, Hamid Eghbal-zadeh, Stefano Cereda, Markus Schedl, Bogdan Ionescu, Paolo Cremonesi

Received: date / Accepted: date

Abstract As of today, most movie recommendation services base their recommendations on collaborative filtering (CF) and/or content-based filtering (CBF) models that use metadata (e.g., genre or cast). In most video-on-demand and streaming services, however, new movies and TV series are continuously added. CF models are unable to make predictions in such a scenario, since the newly added videos lack interactions – a problem technically known as new item cold start (CS). Currently, the most common approach to this problem is to switch to a purely CBF method, usually by exploiting textual metadata. This approach is known to have lower accuracy than CF because it ignores useful collaborative information and relies on human-generated textual metadata, which are expensive to collect and often prone to errors. User-generated content, such as tags, can also be rare or absent in CS situations. In this paper, we introduce a new movie recommender system that addresses the new item problem in the movie domain by (i) integrating state-of-the-art audio and visual descriptors, which can be automatically extracted from video content and constitute what we call the *movie genome*; (ii) exploiting an effective data fusion method named *canonical correlation analysis* (CCA), which was successfully tested in our previous works [28, 22], to better exploit complementary information between different modalities; (iii) proposing a two-step hybrid approach which trains a CF model on warm items (items with interactions) and leverages the learned model on the movie genome to recommend

Yashar Deldjoo · Maurizio Ferrari Dacrema · Stefano Cereda · Paolo Cremonesi
Politecnico di Milano, Italy
E-mail: {yashar.deldjoo, maurizio.ferrari, stefano.cereda, paolo.cremonesi}@polimi.it

Mihai Gabriel Constantin · Bogdan Ionescu
University Politehnica of Bucharest, Romania
E-mail: {mgconstantin, bionescu}@imag.pub.ro

Markus Schedl · Hamid Eghbal-zadeh
Johannes Kepler University Linz, Austria
Department of Computational Perception
E-mail: {markus.schedl, hamid.eghbal-zadeh}@jku.at

cold items (items without interactions). Experimental validation is carried out using a system-centric study on a large-scale, real-world movie recommendation dataset both in an absolute cold start and in a cold to warm transition; and a user-centric online experiment measuring different subjective aspects, such as satisfaction and diversity. Results show the benefits of this approach compared to existing approaches.

Keywords movie recommender systems, cold start, warm start, semi-cold start, new item, multimedia features, content-based, audio descriptors, visual descriptors, multimodal fusion, hybrid recommender system, feature weighting, collaborative-enriched content-based filtering, canonical correlations analysis

1 Introduction

A dramatic rise in the generation of video content has occurred in recent years. According to Cisco, the largest networking company across the globe, by 2020 more than 75% of the world's mobile data traffic will be video, or even 80% when video and audio data are considered together [1]. This rise has been fueled by online social network users who upload/post a staggering amount of user-generated video on a daily basis. For instance, as of 2018, YouTube[1] users upload over 400 hours of video every minute. This translates to about 3 years of non-stop watching in order to consume all videos uploaded to YouTube in a single hour. Similarly, Instagram[2] users post nearly 70 million photos and videos each day [109].

In this context, video recommender systems play an important role in helping users of online streaming services, as well as of social networks, cope with this rapidly increasing volume of videos and provide them with personalized experiences. Nevertheless, the growing availability of digital videos has not been fully accompanied by comfort in their accessibility via video recommender systems. The causes of this problem are two fold: (i) the type of recommendation models in service today, which are heavily dependent on usage data (in particular, implicit or explicit preference feedback) and/or metadata (e.g., genre and cast associated with the videos) (cf. Section 1.1), and (ii) the nature of video data, which are information intensive when compared to other media types, such as music or images (cf. Section 1.2). In the following article, we analyze each of these dimensions. Throughout this paper, we will use a number of abbreviations, which, for convenience are summarized in Table 1.

1.1 New item cold-start recommendation in the movie domain

To date, collaborative filtering (CF) methods [61] lie at the core of most real-word movie recommendation engines, due to their state-of-the-art accuracy [80,111]. In most video-streaming services, however, new movies and TV

[1] `https://www.youtube.com/`

[2] `https://www.instagram.com/`

Table 1: List of abbreviations used throughout the paper.

Abbreviation	**Term**
MM	Multimedia
RS	Recommender systems
VRS	Video recommender systems
MRS	Movie recommender systems
MMRS	Multimedia recommender systems
CS	Cold start
WS	Warm start
AVF	Aesthetic visual features
BLF	Block-level features
CBF	Content-based filtering
CF	Collaborative filtering
CF-MMRS	Collaborative filtering multimedia recommender system
CB-MMRS	Content-based multimedia recommender system
CA-MMRS	Context-aware multimedia recommender system
BPR	Bayesian personalized ranking
KNN	K-Nearest Neighbor
CFeCBF	Collaborative-filtering enriched content-based filtering

series are continuously added. CF models are not capable of providing meaningful recommendations when items in the catalogue contain few interactions, a problem commonly known as the cold start (CS) problem. The most severe case of CS is when new items are added that lack any interactions, technically known as the *new item* CS problem.[3] In such a situation, CF models are completely unable to make predictions. As such, these new items are not recommended, go unnoticed by a large part of the user community, and remain unrated, creating a vicious circle in which a set of items in the RS is left out of the vote/recommendation process [11]. Being able to provide high-quality recommendations for cold items has several advantages. Firstly, it will increase the novelty of the recommendations, which is a highly desirable property and inherent in the user-centric and business-centric goals of RS, i.e., the discovery of new content and the increase of revenues [5,69]. Secondly, providing good new movie recommendations will allow enough interactions/feedbacks to be collected in a brief amount of time enabling effective CF recommendation. Despite previous efforts, the new item CS problem remains far from being solved in the general case, and most existing approaches suffer from it [11,116,113].

Currently, the most common approach to counteracting the new item CS problem is to switch to a pure CBF [44,75] method by using additional attribute content for items, usually by resorting to metadata provided in textual form [71]. This approach is known to have lower accuracy than CF because

[3] Note that videos without interactions can also be old videos that have been never been watched by a user.

it ignores potentially useful collaborative information and typically relies on human-generated textual metadata, which are often noisy, expensive to collect, and sparse. More importantly, extra information for cold items is not always available on the web (especially in user-generated form), even if it is available in abundance for warm items [114]. In addition, given the unstructured or semi-structured nature of metadata, they often require complex natural language processing (NLP) techniques for pre-processing, e.g., syntactic and semantic analysisor topic modeling [4].

Many approaches have been proposed to address the new item CS issue, mainly based on hybrid CF and CBF models [68, 14, 98, 42]. Most recent work relies on machine learning to combine content and collaborative data. We focus on feature weighting rather than on other types of hybrids (e.g., joint matrix factorization) because we aim to build a hybridization strategy that can be easily applied to a CBF model. For instance, the authors in [43] proposed a method to map item features into the item embeddings learned in a matrix factorization algorithm, while the authors in [95] defined a probabilistic model trained via expectation minimization. Another example is [98], where the authors proposed a feature weighting model that learns feature weight by optimizing the ranking of the recommendations over the user interactions for warm items.

Addressing this issue, the main contribution of the present work is to improve the current state of the art by presenting a generalized, two-step machine learning approach to feature weighting and by testing its effectiveness on both editorial features and state-of-the-art multimedia (MM) descriptors. Hereafter, for simplicity, we refer to items without interactions as *cold items* and items containing interactions as *warm items*.

1.2 Video as an information-intensive multimodal media type

When we watch a movie, we can effortlessly register many details conveyed to us through different multimedia channels — in particular, the audio and visual channels. As a result, the perception of a film in the eyes of viewers is influenced by many factors related not only related to, e.g., the genre, cast, and plot, but also according to the overall film style [12]. These factors affect the viewer's experience. For example, two movies may be from the same genre and director, but they can be different based on the movie style. Consider as an example *Empire of the Sun* and *Schindler's List*, both dramatic movies directed by Steven Spielberg and both describing historical events. However, they are completely different in style, with *Schindler's List* shot like a documentary in black and white, while *Empire of the Sun* is shot using bright colors and makes heavy use of special effects. Although these two movies are similar with respect to traditional metadata (e.g., director, genre, year of production), their different styles are likely to affect the viewers' feelings and opinions differently [27]. In fact, the film story is first created by the author and the comprehension of the cinematographical language by the spectator reshapes the story [41]. The notion of story in a movie depends on semantic content (reflected better

in *metadata*) reshaped through stylistic cinematography elements (reflected better in *multimedia content*). These discernible characteristics of movie content meet users' different information needs.

The extent to which content-based approaches are used, and even the way "content" is interpreted, varies between domains. While extracting descriptive item features from text, audio, image, and video content is a well-established research domain in the multimedia community [65], the recommender system community has long considered metadata, such as the title, genre, tags, actors, or plot of a movie, as the single source for content-based recommendation models, thereby disregarding the wealth of information encoded in the actual content signals. In order for MRS to make progress in recommending the right movies to the right user(s), they need to be able to interpret such multimodal signals as an ensemble and utilize item models that take into account the maximum possible amount of this information. We refer to such a holistic description of a movie, taking into account all available modalities, as its *movie genome*, since it can be considered the footprint of both content and style [13, 50].[4]

In this paper, we specifically address the above-mentioned shortcomings of purely metadata-based MRS by proposing a practical solution for the new item CS challenge that exploits the *movie genome*. We set out to answer the following research questions:

RQ1: *Can the exploitation of movie genome describing rich item information as a whole, provide better recommendation quality compared with traditional approaches that use editorial metadata such as genre and cast in CS scenarios?*

RQ2: *Which visual and audio information better captures users' movie preferences in CS scenarios?*

RQ3: *Can we effectively leverage past user behavior data on warm items (items with interactions) to enrich the overall item representation and improve our ability to recommend cold items when interactions are not available?*

The remainder of this article is structured as follows. Section 2 positions our work in the context of the state of the art and highlights its novel contributions. Section 3 introduces the proposed general content-based recommendation framework. Sections 4 and 5 report on the experimental validation, namely the experimental setup and parameter tuning, offline experimentation, and a user study in a web survey, respectively. Section 6 concludes the article in the context of the research questions and discusses limitations and future perspectives.

[4] Similar to biological DNA composed of long sequences of four letters A, T, C, G referred to as nucleotides.

2 State of the art

One main contribution of this work is the introduction of a solution for the new item CS problem in the multimodal movie domain. In this section, we therefore review the existing, state-of-the-art approaches in content-based multimedia recommender systems (Section 2.1) and feature weighting for CS recommender systems (Section 2.2) and position our contribution (Section 2.3).

2.1 Content-based multimedia recommendation

A multimedia recommendation system is a system that recommends a particular media type, such as audio, image, video, and/or text, to the users [33, 34]. We therefore organize the state-of-the-art CB-MMRS based on the target media type, namely: (i) audio recommendation, (ii) image recommendation, and (iii) video recommendation. In the following subsections, we describe each of these systems.

2.1.1 Audio recommendation

The most common example of audio recommendation is *music recommendation* [94,105]. Over the past several years, a wealth of approaches, including CF, CBF, context-aware recommenders, and hybrid methods, have been proposed to address this task. An overview of popular approaches can be found in [93,94]. Perhaps more than in other MM domains, CB recommenders have attracted substantial interest from researchers in the music domain, not least due to their superior performance in CS scenarios.

Recent work has proposed deep learning-based CB approaches. For instance, the authors in [85] use a deep convolutional neural network (CNN) trained on audio features, more precisely, on the log-scaled Mel spectrograms extracted from 3-second-snippets of the audio, resulting in a latent factor representation for each song. The authors evaluate their approach for tag prediction and music recommendation using the Million Song Dataset [10]. In 10-fold cross-validation experiments using 50-dimensional latent factors, they show that the CNN outperforms both metric learning to rank and a multilayer perceptron trained on bag-of-words representations of vector-quantized Mel frequency cepstral coefficients (MFCC) [73] in both tasks.

In contrast to such automatic feature learning approaches, some systems use human-made annotations of music. Perhaps, the most notable and well-known is the proprietary *Music Genome Project* (MGP),[5] which is used by music streaming major Pandora.[6] MGP captures various attributes of music and uses them in a CBF recommender system. These attributes are created by musical experts who manually annotate songs. Pandora uses up to 450 specific

[5] `http://www.pandora.com/about/mgp`

[6] Pandora might also use automatically extracted content (and other) features in their system, but the MGP is arguable the approach for which Pandora is best known.

descriptors per song, such as "aggressive female vocalist", "prominent backup vocals", or "use of unusual harmonies".

*In our approach, we follow a strategy in between these two extremes (*i.e., *fully automated feature learning by deep learning and pure manual expert annotations). The proposed movie genome uses well-established, state-of-the-art audio descriptors that are semantically more meaningful than deep learned features, but at the same time do not require a massive number of human annotators.*

2.1.2 Image recommendation

Some interesting use-case scenarios of image recommendation can be mentioned in the *fashion* domain (e.g., recommending clothes) and the *cultural heritage* domain (e.g., recommending paintings in museums). For fashion, recommendation can be performed in two main manners: finding a piece of clothing that *matches* a given garment image shown to the system as a visual query (such as two pairs of jeans which are similar to each other considering their visual appearance) and finding the clothing, which *complements* the given query (such as recommending a pair of jeans that match a shirt). The authors in [78] propose a CB-MMRS which provides personalized fashion recommendations by considering the visual appearance of clothes. The main novelty, besides focusing on this novel fashion recommendation scenario, is examining the visual appearance of the items under investigation to overcome the CS problem.

The authors of [8] propose a multimedia (image—video—document) recommender platform to address the cultural heritage domain: in particular, a recommender system to provide personalized visiting paths to tourists visiting the Paestum ruins, one of the major Greco-Roman cities in the South of Italy. The proposed system is able to uniformly combine heterogeneous multimedia data and to provide context-aware recommendation techniques. This paper provides interesting insights for building context-aware multimedia systems using content information, with explicit focus on contextualization. The authors exploit high-level metadata extracted in an automatic or semi-automatic manner from low-level (signal-level) features and compare it with user preferences. The main shortcoming of this research is the lack of an experimental study on a larger multimedia dataset.

Visual descriptors have also been used in restaurant recommendation systems by the authors of [16], in which images collected from a restaurant-based social platform were first processed by an SVM-based image classification system that used both low-level and deep features and split the images into four classes, indoor, outdoor, food and drink images, based on the idea that these different categories of pictures may have different influences on restaurant recommendation. This content-based approach was used to successfully enhance the performance of matrix factorization, Bayesian personalized ranking matrix factorization and FM approaches.

In our approach, we follow a strategy that also recognizes the importance of low-level content (visual and audio) for movie recommendation and leverages it for new item CS movie recommendation.

2.1.3 Video recommendation

As one of the earliest approaches to the problem of video recommendations, the authors of [110, 82, 81] propose a video recommender system named "Video Reach". Given an online video and related information (query, title, tags and surrounding text), the system recommends relevant videos in terms of multimodal relevance and user feedback. Two types of user feedback are leveraged: browsing behavior and playback on different portions of the video (the latter is specific to [81]). These approaches are interesting from the perspective of using multimodal video content (audio, visual, and textual) and a fusion scheme based on user behavior. However, they have some limitations as well. Firstly, according to the properties required by the attention fusion function, the proposed Video Reach system filters out videos with low textual similarity to ensure that all videos are more or less relevant and then only calculates the visual similarity of the filtered videos; this may result in losing important information. Secondly, it uses only one type of visual feature, namely the basic color histogram. Thirdly, an empirical set of weights is chosen to serve as importance weights in a linear feature/modality fusion; for example, the textual keywords are given a much higher weight than the visual and aural keywords, without investigating the opposite arrangement. Although the authors show that this assumption is sufficient to make recommendation via adjusting weights, it is not clear what effect such an empirical assumption has.

*In our approach, we introduce a video recommendation system that leverages all video properties (*i.e., *audio, visual, and textual) and an effective fusion method based on canonical correlation analysis (CCA) to exploit the complementary information between modalities in order to produce more powerful combined descriptors. More importantly, we propose an approach for new item recommendation that leverages the collaborative knowledge about warm items for the CBF of cold items, using the combined descriptors.*

2.2 Feature weighting for cold-start recommender systems

Relying on CBF algorithms to address cold items has two main drawbacks: firstly, it is limited by the availability and quality of item features, and secondly, it is difficult to connect the content and collaborative information. One way to build a hybrid of content and collaborative information is via feature weighting. We focus on feature weighting rather than on other types of hybrids because we aim to build a hybridization strategy that can be easily applied to a CBF model. Feature weighting algorithms can be either embedded methods, which learn feature weights as part of the model training, or wrapper methods, which learn weights in a second phase on top of an already available model. Examples of embedded methods are user-specific feature-based similarity models (UFSM) [7] and factorized bilinear similarity models (FBSM) [98]. Among embedded methods, the main drawbacks are the complex training phase and a sensitivity to noise due to the strong coupling of features and interactions. UFSM learns a personalized linear combination of similarity func-

tions, known as global similarity functions for cold-start top-N item recommendations. UFSM can be considered a special case of FM [7,90]. FBSM was proposed as an evolution of UFSM that aims to discover relations among item features instead of building user-specific item similarities. The model builds an item-item similarity matrix which models how how well a feature of an item interacts with all the features of the second item.

Wrapper methods, meanwhile, rely instead on a two-step approach by learning feature weights on top of an already available model. An example of this is least-square feature weights (LFW) [14], which learns feature weights from a SLIM item-item similarity matrix using a simpler model than FBSM:

$$sim(i,j) = \mathbf{f}_i^T \mathbf{D} \mathbf{f}_j \tag{1}$$

where f is the feature vector of an item and D is a diagonal matrix having as dimension the number of features. Another example of a wrapper method is *HP3* [9], which builds a hybrid recommender on top of a graph-based collaborative model. A generalization of LFW has recently been published by the authors in [42]. They demonstrate the effectiveness of wrapper methods in learning from a wider variety of collaborative models and present a comparative study of some state-of-the-art algorithms. Their paper further shows that wrapper methods with no latent factor component (i.e., matrix $\mathbf{V}$, as in FBSM) tend to outperform others. *In our approach, we therefore choose to adopt this simpler model, as it combines good recommendation quality with fast training time.*

Similar strategies are available for matrix factorization models. Collective matrix factorization [100] allows the joint factorization of both collaborative and content data, which is applied in [92] to propose local collective embedding, a joint matrix factorization that enforces the manifold structure exhibited by the collective embedding in the content data as well as allowing collaborative interactions to be mapped to topics. An example of a wrapper method is attribute to feature mapping [43], an attribute-aware matrix factorization model which maps item features to its latent factors via a two-step approach. All previous approaches rely on the availability of some descriptors for each item, which in some cases can be an issue.

Other proposals to address the CS problem make use of other relations between users or items, i.e., social networks. For example, the authors in [115] use social tags to enrich the descriptions of items in a user-tag-object tripartite graph model; while the authors in [76] instead use a social trust network to enrich the user profile. Another example is [107], where authors analyze the impact of the connections on the quality of recommendations. While this group of techniques shows promising results, it is still limited by the fact that obtaining fine-grained and accurate features is a complex and time-consuming task. Moreover, those other existing relationships might not always be available or meaningful for the target domain. See [38] for a good and general introduction to recommendation complicating scenarios (e.g., the CS problem).

In this work, we adopt feature weighting techniques because they have shown promising results in recent years to the point of becoming the current state of the art.

2.3 Contributions of this work

The work at hand builds on foundations and results realized in our previous work, but considerably extends it. We therefore present in the following our novel contributions, and connect them to previous work.

In [29,31,26,27,39,17], we proposed a CB-MRS that implements a movie filter according to *average shot length* (measure of camera motion), *color variation*, *lighting key* (measure of contrast), and *motion* (measure of object and camera motion). The proposed features were originally used in the field of multimedia retrieval for movie genre classification [89] and have a stylistic nature which is believed to be in accordance with applied media aesthetics [112] for conveying communication effects and simulating different feelings in the viewers. For this reason, these features were named mise-en-scène features.

Since full movies can be unavailable, costly or difficult to obtain, in [27] it was studied whether movie trailers can be used to extract mise-en-scène visual features. The results indicated that they are indeed correlated with the corresponding features extracted from full-length movies and that feeding the features extracted from movie trailers and full movies into a similar CB-MRS results in a comparable quality of recommendations (both superior to the genre baseline). The main shortcoming of this work is that it used a small dataset for evaluation (containing only 167 movies and the corresponding trailers). Additionally, the number of visual features was limited (only five features, cf. [89]). Due to these restrictions, the generalizability of our findings in [27] may be limited; also see Section 6.3 for a discussion of limitations.

In [28,30,25] we specifically addressed the under-researched problem of combining visual features extracted from movies with available semantic information embedded in metadata or collaborative data available in users' interaction patterns in order to improve offline recommendation quality. To this end, for multimodal fusion (i.e., fusing features from different modalities) in [28], for the first time, we investigated adoption of an effective data fusion technique named *canonical correlation analysis* (CCA) to fuse visual and textual features extracted from movie trailers. A detailed discussion about CCA can be found in Section 3.2. Although a small number of visual features were used to represent the trailer content (similar to [27]), the results of offline recommendation using 14K trailers suggested the merits of the proposed fusion approach for the recommendation task. In [30] we extended [28] and used both low-level visual features (color- and texture-based) using the MPEG-7 standard together with deep learning features in a hybrid CF and CBF approach. The aggregated and fused features were ultimately used as input for a collective sparse linear method (SLIM) [83] method, generating an enhancement for the CF method. While the results for each of these two features improved the genre and tag baselines, the best results were achieved with the CCA fusion approach. Although [30] significantly extended the previous works [28,28] both in terms of the content and the core recommendation model, it ignored the role of the audio modality in the entire item modeling.

Finally, in [25], we used factorization machines (FM) [90] as the core recommendation technique. FM is a general predictor working with any real valued

feature vector and has the power of capturing all interactions between variables using factorized parameters. FM was used specifically with the goal of encoding the interactions between mise-en-scène visual features and metadata features for the recommendation task. Please note that in the present work, we neither use FM nor SILM, specifically because one of the main contributions of the work at hand is to propose and simulate a novel technique for new item recommendation for which FM or SLIM are not applicable.

In a different research line, in [39], we designed an online movie recommender system which incorporates mise-en-scène visual features for the evaluation of recommendations by real users. We performed an offline performance assessment by implementing a pure CB-MRS with three different versions of the same algorithm, respectively based on (i) conventional movie attributes, (ii) mise-en-scène visual features, and (iii) a hybrid method that interleaves recommendations based on the previously noted features. As a second contribution, we designed an empirical study and collected data regarding the quality perceived by the users. Results from both studies showed that the introduction of mise-en-scène, together with traditional movie attributes, improves the quality of both offline and online recommendations. However, the main limitation of [39] is that we used basic late fusion by interleaving the recommendations to combine recommendations generated by different CBF systems.

In summary, although we achieved relevant progress, some limitations of our previous work remain unsolved: (i) solely visual and/or text modalities were considered, forgetting the rich audio information (e.g., conversations or music); (ii) better fusion techniques are required to fully exploit the complementary information from (several) modalities; (iii) visual content can be represented with richer descriptors; and (iv) the recommendation model used was either a CBF model based on KNN or a CBF+CF model based on SLIM, both of which are not capable to deal with new item CS scenarios.

In this paper, we enhance these previous achievements and go beyond the state of the art in the following directions:

1. We propose a *multimodal* movie recommendation system which exploits established multimedia *aesthetic-visual features*; *block-level audio features*; state-of-the-art *deep visual features*; and *i-vectors audio features*. Apart from the use of automated content descriptors, the system uses as input movie trailers instead of complete movies, which makes it more versatile, as trailers are more readily available than full movies. We show that the proposed CB-MRS outperforms the traditional use of metadata. To the best of our knowledge, this has not previously been achieved, existing systems being limited to the use of either visual and/or textual modalities [27, 24, 25] or basic low-level descriptors [110, 81];
2. We propose a practical solution to the CS new-item problem where user behavior data are unavailable, and therefore neither CF nor CBF using user-generated content are applicable. Our solution consists of a two-step approach named *collaborative-filtering-enriched content-based filtering* (CFeCBF) to leverage the collaborative knowledge about warm items and exploit it for CBF on cold items.

3. To achieve multimodal MRS, we adopt an early fusion approach using *canonical correlation analysis* (CCA), which was successfully tested in our previous works [28,30] for combining heterogeneous features extraced from different modalities (audio, visual and textual). CCA is often used when two types of data (feature vectors in training) are assumed to correlate. We hypothesize that this is relevant in the movie domain and that combining audio, visual, and textual data enriches the recommendations.
4. We evaluate the quality of the proposed *movie genome* descriptors by two comprehensive wide and articulated empirical studies: (i) a *system-centric* experiment to measure the offline quality of recommendations in terms of accuracy-related metrics, i.e., mean average precision (MAP) and normalized discounted cumulative gain (NDCG); and beyond-accuracy metrics [53], i.e., list diversity, distributional diversity, and coverage; (ii) a *user-centric* online experiment involving 101 users, computing different subjective metrics, including relevance, satisfaction, and diversity.
5. We publicly release the resources of this work to allow researchers to test their own recommendation models. The dataset was already released partly in [23] while the code is now available on Github.[7]

3 Proposed recommendation framework

The main processing stages involved in our proposed CFeCBF-MRS are presented in Figure 1. As previously mentioned, the only input information, apart from the collaborative one, is the movie trailers. First, we perform pre-processing that consists of decomposing the visual and audio channels into smaller and semantically more meaningful units. We use frame-level and block-level segmentation for the audio channel. For video, we use the frames captured at 1 fps. The next step consists of computing meaningful content descriptors (cf. Section 3.1), namely: (i) *multimedia* — *audio* and *visual* features; and (ii) *metadata* — movie *genres*. Features are aggregated temporally using different video-level aggregation techniques, such as statistical summarization, Gaussian mixture models (GMM), and vectors of locally aggregated descriptors (VLAD) [52]. Features are fused by using the early fusion method CCA (cf. Section 3.2). At this stage, each video is represented by a feature vector of fixed length, which is referred to as the item profile. A collaborative recommender is trained on all available user-item interactions in order to model the correlations encoded in users' interaction patterns, using the similarity of ratings as an indicator of similar preference. As the last step, the CFeCBF weighting scheme is trained on the given item profile and collaborative model to discover the hybrid feature weights. The learned feature weights are then applied to a CBF recommender able to provide recommendations for cold items. Each of these steps is detailed in the following sections.

[7] `https://github.com/MaurizioFD/CFeCBF`

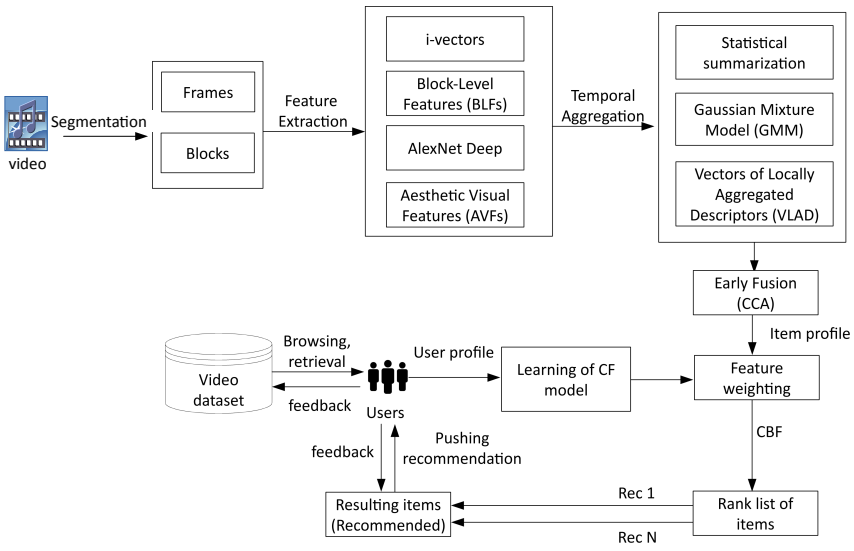

Fig. 1: The proposed collaborative-filtering-enriched content-based filtering (CFeCBF) movie recommender system framework.

3.1 Rich item descriptions to model the movie genome

Similar to biological DNA, which represents a living being, multimedia content information can be seen as the genome of video recommendation, i.e., the footprint of both content and style. In this section, we present the rich content descriptors integrated into the proposed movie recommendation system to boost its performance. These features were selected based on their effectiveness in representing multimedia content in various domains and comprise both audio and visual features [22,23].

3.1.1 Audio features

The exploited audio features are inspired by the fields of speech processing and music information retrieval (MIR) and by their successful application in MIR-related tasks, including music retrieval, music classification, and music recommendation [58]. We investigate two kinds of audio features: (i) *block-level features* [96] which consider chunks of the audio signal known as *blocks* and are therefore capable of exploiting temporal aspects of the signal; and (ii) *i-vector features* [36] which are extracted at the level of audio segments using audio frames. Both approaches eventually model the feature at the level of the entire audio piece; by aggregating the individual feature vectors across time.

Block-level features: We extract block-level features (BLF) from larger audio segments (several seconds long) as proposed in [97]. They can capture

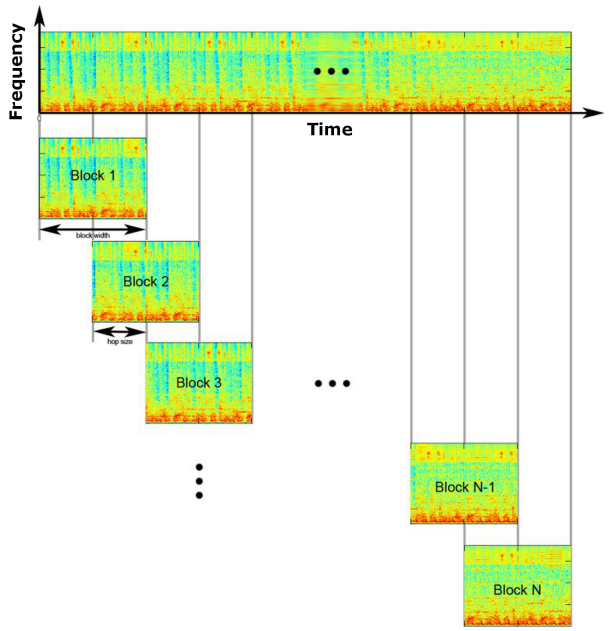

Fig. 2: Overview of the feature extraction process in the block-level features (BLF), according to [97].

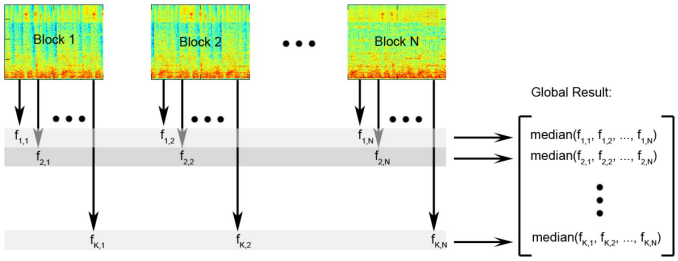

Fig. 3: Obtaining a global feature representation from individual blocks in the block-level framework, according to [97].

temporal aspects of an audio recording and have been shown to perform very well in audio and music retrieval and similarity tasks [96] and can be considered state of the art in this domain.

The BLF framework [97] defines six features. These capture *spectral aspects* (spectral pattern, delta spectral pattern, variance delta spectral pattern), *harmonic aspects* (correlation pattern), *rhythmic aspects* (logarithmic fluctuation pattern), and *tonal aspects* (spectral contrast pattern). The feature extraction process in the block-level framework is illustrated in Figure 2. Based on the spectrogram, blocks of fixed length are extracted and processed one at a time.

The block width defines how many temporally ordered feature vectors comprise a block. The hop size is used to account for possible information loss due to windowing. After having computed the feature vectors for each block, a global representation is created by aggregating the feature values along each dimension of the individual feature vectors via a summarization function, which is usually expressed as a percentile, as illustrated in Figure 3. A more technical and algorithmic discussion can be found in [97]. The extraction process results in a $9,948$-dimensional feature vector per video.

I-vector features: I-vector is a fixed-length and low-dimensional representation containing rich acoustic information, which is usually extracted from short segments (typically from 10 seconds to 5 minutes) of acoustic signals such as speech, music, and acoustic scene. The i-vector features are computed using frame-level features such as mel-frequency cepstral coefficients (MFCCs). In a movie recommendation system, we define total variability as the deviation of a video clip representation from the average representation of all video clips. I-vectors are latent variables that capture total variability to represent how much an audio excerpt is shifted from the average clip. The main idea is to first learn a universal background model (UBM) to capture the average distribution of all the clips in the acoustic feature space using a dataset containing a sufficient amount of data consisting of different movie clips. The UBM is usually a Gaussian Mixture Model (GMM) and serves as a reference to measure the amount of shift for each segment where the i-vector is the estimated shift.

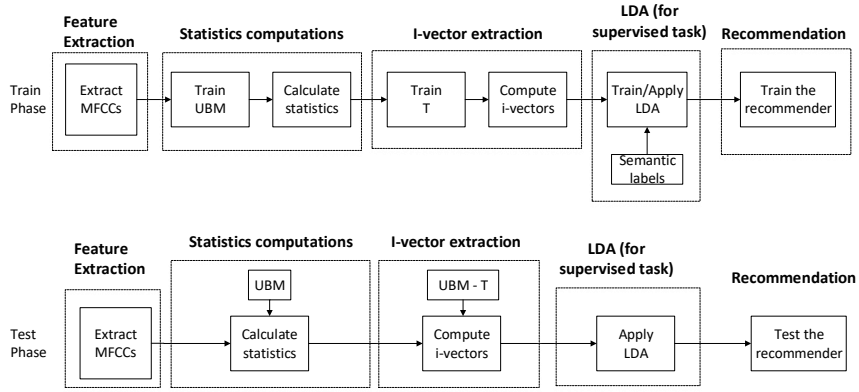

Fig. 4: Block diagram of i-vector FA pipeline for, both, supervised and unsupervised approaches.

The block-diagram of the i-vector pipeline, from frame-level feature extraction to i-vector extraction and finally to recommendation, is shown in Figure 4. The framework can be decomposed into several stages: (i) *Frame-level feature extraction:* MFCCs have proven to be useful features for many audio and mu-

sic processing tasks [74,40,36]. They provide a compact representation of the spectral envelope are also a musically meaningful representation [36], and are used to capture acoustic scenes [35]. Even though it is possible to use other features [103], we avoid the challenges involved in feature engineering and instead focus on the timbral modeling technique. We used a 20-dimensional MFCCs feature; (ii) *Computation of Baum-Welch statistics:* In this step, we collect sufficient statistics by adapting UBM to a specific segment. This is a process in which a sequence of MFCC feature is represented by the Baum-Welch (BW) statistics (0-th and 1-st order Baum-Welch statistics) [64,56] using a GMM as prior; (iii) *I-vector extraction:* I-vector extraction refers to the extraction of total factors from BW statistics. This step reduces the dimensionality of the movie clip representations and improves the representation for a recommendation task; (iv) *Recommendation:* Recommendation is effected by integrating the extracted i-vector features in a CBRS.

During the training phase, the UBM is trained on the items in the training dataset and is used as an external knowledge source for the test dataset. In the testing step, test i-vectors are extracted using the models from the training step and the MFCCs of the test set. In the supervised approach, these i-vectors are projected by LDA in the training step. For the i-vector extraction, we used 20-dimensional MFCCs. For the items in the training set (in each fold), we trained a UBM with either 256 or 512 Gaussian components and a different dimensionality of latent factors (40, 100, 200, 400). We performed a *hyper-parameter search* and reported the best results obtained over 5-fold cross-validation for each evaluation metric.

3.1.2 Visual features

The visual features we selected for our experiments were previously used in other domains, including image aesthetics, media interestingness, object recognition, and affect classification. We selected two types of visual features: (i) *aesthetic visual features*, a set of features mostly associated with media aesthetics, and (ii) *deep learning features* extracted from the fc7 layer of the AlexNet deep neural network, initially developed for visual object recognition, but extended and used in numerous other domains. Several aggregation methods were also performed with these features, with the goal of obtaining video-level descriptors from the frame-level set of extracted features.

Aesthetic-visual features: the three groups of features and their early fusion combinations were aggregated in a standard statistical aggregation scheme based on mean, median, variance, and median absolute deviation. In a work discussing the measurement of coral reef aesthetics, the authors in [45] propose a set of features inspired by the aesthetic analysis of artwork [66] and photographic aesthetics [19,54]. This collection of features is derived from related domains, such as photographic style, composition, and the human perception of images, and was grouped into three general features types: color-related, texture-related and object-related.

The color-related features have 8 main components. The first elements consist of the average channel values extracted from the *HSL and HSV color spaces*. A *colorfulness* measure was created by calculating the Earth Mover's Distance, Quadratic Distance and standard deviation between two distributions: the color frequency in each of the 64 divisions of the RGB spectrum and an equal reference distribution. The *hue descriptors* contained statistical calculations for pixel hues: number of hues present, number of significant hues for the image etc. The *hue models* are based on the distance between the current picture and a set of nine hue models considered appealing for humans inspired by the models presented in [77]. The *brightness* descriptor calculates statistics regarding image brightness, including average brightness values and brightness/contrast across the image. Finally, average *HSV and HSL* values were calculated while taking into account the main focus region and rule of thirds compositional guideline [84].

The texture-related features have 6 components. The *edge* component calculates statistics based on edge distribution and energy, while the *texture* component calculates statistics based on texture range and deviation. Also *entropy* measures were calculated on each channel of the RGB color space, generating a measure of randomness. A three-level Daubechies *wavelet* transform [20] was calculated for each channel of the HSV space along with the values for the *average wavelet*. A final texture component was based on the *low depth-of-field* photographic composition rule, according to the method described by [19].

The object-related features have 11 components. These components are mostly based on the largest segments in an image obtained through the method proposed in [19], which is based on the k-means clustering algorithm. The *area*, *centroids*, values for the *hue*, *saturation*, and *value* channels, *average brightness values*, horizontal and vertical *coordinates*, *mass variance* and *skewness* for the largest, and therefore most salient, segments each constitutes a component of this feature type. *Color* spread and complementarity also represented a component, while the last component calculates hue, saturation, and brightness *contrast* between the resulting segments.

As previously mentioned, this set of features is highly correlated to the human observer, some components being heavily based on psychological or aesthetic aspects of visual communication. For example, the hue model component calculates the distance between the hue model of a certain image and models considered appealing to humans, inspired by the work of [77]. Also, some general rules of photographic style were used, rules previously shown to have a high impact on human aesthetic perception, therefore generating more pleasant images and videos [62]. For example, the authors in [70] modify images in order to achieve a better aesthetic score, one of the rules applied for this optimization being the rule of thirds.

We used these features in our experiments, both separated into the three main feature types (color, texture and object) and in an early fusion concatenated descriptor for each image in the video. Regarding the aggregation method, we used four standard statistical aggregation schemes based on mean, median, variance and median absolute deviation.

Deep-learning features: Deep neural networks have become an important part of the computer vision community, gathering interest and gaining importance as their results started performing better than more traditional approaches in different domains. The ImageNet Large Scale Visual Recognition Competition (ILSVRC) gives the opportunity to test different object recognition algorithms on the same dataset, consisting of a subset of 1.2 million images and $1,000$ different classes taken from the ImageNet[8] database. The AlexNet [63] deep neural network was the winner of the competition in 2012, achieving a top-5 error rate of 15.3% — a significant improvement over the second – best entry – that year. The authors also ran experiments on the ILSVRC 2010 dataset, concluding that the top-1 and top-5 error rates of 37.5% and 17% were again improvements on previous state-of-the-art approaches. One of the novelties introduced by this network was the ReLU (Rectified Linear Units) nonlinearity output function, which was able to achieve faster training times than networks working with more standard functions like $f(x) = tanh(x)$ or $f(x) = (1 + e^{-x})^{-1}$, instead using $f(x) = max(0, x)$.

AlexNet consists of 5 convolutional layers and 3 fully connected layers, ending with a final, $1,000$-dimensional softmax layer. The input of the network consists of a $224 \times 224 \times 3$ image, therefore requiring the original image to be resized if the resolution is different. The five convolutional layers have the following structure: the first layer has 96 kernels of size $11 \times 11 \times 3$; the second, 256 kernels of size $5 \times 5 \times 48$; the third, 384 kernels of size $3 \times 3 \times 256$; the fourth, 384 kernels of size $3 \times 3 \times 192$; and the final, fifth convolutional layer, 256 kernels of size $3 \times 3 \times 192$. The fully connected layers all have $4,096$ neurons, and the output of the final one is fed into a softmax layer that creates a distribution for the $1,000$ labeled classes. This generates a network with 60 million parameters and $650,000$ neurons; thus, in order to reduce overfitting on the original dataset, some data augmentation solutions were employed, including image translations, horizontal reflections, and the alteration of the intensity of the RGB channels and a dropout technique [49].

Given the good performance of the fc7 layer in tasks related to human preference, we chose to extract the outputs of this layer for each frame of our videos, thus obtaining a $4,096$-dimensional descriptor for each image. We then obtain a video-level descriptor through two types of aggregation methods: standard statistical aggregation, where we calculate the mean, median, variance, and median absolute deviation, and VLAD [52] aggregation followed by PCA for dimensional reduction, with three different sizes for the visual word codebook: $k \in \{32, 64, 128\}$.

3.1.3 Metadata features

We also use two types of editorial metadata features to serve as baselines: movie genre and cast/crew features.

Genre features: For every movie, genre features are used to serve as metadata baselines. *Genre Features* (18 categories): *Action*, *Adventure*, *Animation*,

[8] `http://www.image-net.org/`

Children's, *Comedy*, *Crime*, *Documentary*, *Drama*, *Fantasy*, *Film Noir*, *Horror*, *Musical*, *Mystery*, *Romance*, *Sci-Fi*, *Thriller*, *War*, and *Western*. The final genre feature vector is a binary, 18-dimensional vector.[9]

Cast/crew features: For every movie, the corresponding *cast* and *crew* have been downloaded from TMDB[10] using the available API and movie ID mapping provided by Movielens20M. The feature vector contains 162K Boolean features. Each movie is associated, on average, with 25 features.

3.2 Multimodal fusion

Two main paradigms of fusion exist in the literature of multimedia processing [102]: (i) late fusion which generates separate candidate results created by different systems and fuses them into a final set of results; the main limitation of late fusion methods is that they do not consider the correlation among features and are computationally more expensive during training; (ii) early fusion which tries to map multiple feature spaces to a unified space, in which conventional similarity-based evaluation can be conducted.

Motivated by the above, in current work we exploit a *multimodal early fusion* method based on *canonical correlation analysis* (CCA) that was successfully tested in our previous works [28,30]. CCA is a technique for joint fusion and dimensionality reduction across two or more (heterogeneous) feature spaces, which is often used when two set of data are believed to have some underlying correlation. We hypothesize that this is relevant in movie domain and combining audio, visual and textual data enriches the recommendations and training. Additionally, since the focus of the recommendation model in our work is on a CF-enriched CBF model (see Section 3.3), we have realized that currently the proposed method functions better with a lower size of the feature vectors. As CCA reduces the dimensionality of the final descriptor, it is is leveraged greatly in the proposed recommendation framework. Finally, CCA can be pre-computed and used in an off-the-shelf manner making it a convenient descriptor in offline experiments (as opposed to late fusion methods [22]).

We review the concept of CCA here for our methodology. Let $X \in \mathbb{R}^{p \times n}$ and $Y \in \mathbb{R}^{q \times n}$ be two sets of features in which p and q are the dimensions of features extracted from the n items. Let $S_{xx} = \text{cov}(x) \in \mathbb{R}^{p \times p}$ and $S_{yy} = \text{cov}(y) \in \mathbb{R}^{q \times q}$ be the *within-set* and $S_{xy} = \text{cov}(x, y) \in R^{p \times q}$ be the *between-set* covariance matrix. Let us further define $S \in \mathbb{R}^{(p+q) \times (p+q)}$ as the *overall covariance matrix* — a complete matrix which contains information about

[9] While presenting the results, we will use the genre metadata as the baseline for evaluation, as it is prevalent in the domain. Furthermore, we refrain from using user-generated metadata such as tag features in this work, since in a new item CS situation these features cannot exist.

[10] `https://www.themoviedb.org/`

associations between pairs of features — represented as follows:

$$S = \begin{pmatrix} S_{xx} & S_{xy} \\ S_{yx} & S_{yy} \end{pmatrix} = \begin{pmatrix} \text{cov}(x) & \text{cov}(x,y) \\ \text{cov}(y,x) & \text{cov}(y) \end{pmatrix} \tag{2}$$

The aim of CCA is to identify a pair of linear transformations, represented by $X^* = W_x^T X$ and $Y^* = W_y^T Y$, that maximizes the pairwise correlation across two feature sets given by

$$\arg \max_{W_x, W_y} \text{corr}(X^*, Y^*) = \frac{\text{cov}(X^*, Y^*)}{\text{var}(X^*) \cdot \text{var}(Y^*)} \tag{3}$$

where $\text{cov}(X^*, Y^*) = W_x^T S_{xy} W_y$ and $\text{var}(X^*) = W_x^T S_{xx} W_x$ and $\text{var}(Y^*) = W_y^T S_{yy} W_y$.

In order to solve the above optimization problem, we use the maximization procedure described in [46]. The CCA model parameters W_x and W_x are learned on trained items (warm items) and leveraged both in the training and test phases. We investigate two ways to perform fusion: (i) via concatenation (abbreviated by 'ccat') and (ii) via summation (abbreviated by 'sum') of the transformed features.

3.3 The cold-start recommendation model

The core recommendation model in our system is a standard pure CBF system using Eq. 4 to compute similarities between different pair of videos:

$$sim(i,j) = \frac{\mathbf{f}_i \, D \, \mathbf{f}_j}{\|\mathbf{f}_i\|_F^2 \, \|\mathbf{f}_j\|_F^2} \tag{4}$$

where $\mathbf{f}_i \in \mathbb{R}^{n_F}$ is the feature vector for video i, $\|\|_F^2$ is the Frobenius norm and n_F is the number of features. We are interested in finding the diagonal weight matrix $D \in \mathbb{R}^{n_F \times n_F}$, which represents the importance of each feature.

An underlying assumption is that a CF model will achieve much higher recommendation quality than CBF and will be better able to capture the user's point-of-view. We use a CF model to learn D, cast into the following optimization problem:

$$\underset{\mathbf{D}}{\text{argmin}} \quad \left\| \mathbf{S}^{(\mathbf{CF})} - \mathbf{S}^{(\mathbf{D})} \right\|_F^2 + \alpha \, \|\mathbf{D}\|_F^2 + \beta \, \|\mathbf{D}\| \tag{5}$$

where $\mathbf{S}^{(\mathbf{CF})}$ is the item-item collaborative similarity matrix from which we want to learn, $\mathbf{S}^{(\mathbf{D})}$ is the item-item hybrid similarity metric presented via Eq. (4), $\mathbf{D}$ is the feature weight matrix, α and β are the weights of the regularization terms.

We call this model *collaborative-filtering-enriched content-based filtering* (CFeCBF). The optimal $\mathbf{D}$ is learned via machine learning, applying stochastic gradient descent with Adam [57], which is well suited for sparse and noisy

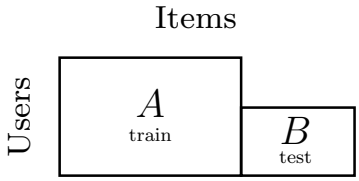

Fig. 5: User interactions' item-wise split; A contains warm items while B contains s and refers to a subset of the users.

gradients. The code is available on Github[11]. CFeCBF is a wrapper method for feature weighting; therefore, it does not learn weights while building the model but rather relies on a previously trained model and then learns feature weights as a subsequent step. Since the model we rely on is collaborative, we can only learn weights associated with features that occur in warm items. This affects how well the algorithm can perform in scenarios where the available features are too sparse; in this case, the number of features appearing in s but not in warm items will tend to increase, reducing the number of parameters in the model.

It is important to point out that while it will be possible to learn a zero collaborative similarity for items having a common feature, it will not be possible to learn anything for items with no common features. Therefore, content-based similarity poses a hard constraint on the extent to which collaborative information can be learned. As content-based similarity is a function of the item features, the sparser this matrix is the less information will be learnable from a collaborative model. This could be a challenge when using Boolean features that tend to be sparse, but much less of one when using real-valued attributes like the multimedia descriptors, which result in dense feature vectors. A consequence of this is that the success of applying CFeCBF on a given dataset depends not only on how accurate the collaborative model is, but also on whether its similarity structure, resulting from the items having common features, is sufficiently compatible with that of the content-based model[12]. CFeCBF requires a two-step training procedure. In the first step, we aim to find the optimal hyper-parameters for a collaborative model by training it on warm items and selecting the optimal hyper-parameters via cross-validation. Since we want a single hyper-parameters set, not one for each fold, we chose those with the best average recommendation quality across all the training folds.

Once the collaborative model is available, the second step is to learn weights by solving the minimization problem described in Eq. 5. As the purpose of this method is to learn $\mathbf{D}$, or feature weights, the optimal hyper-parameters for the machine learning phase are chosen via a cold item split to improve the CBF on new items. Figure 5 shows how a cold item split is performed: split A

[11] `https://github.com/MaurizioFD/CFeCBF`

[12] As our experiments have showed, the best collaborative similarity will not necessarily yield the best weights.

represents the warm items, that is, items for which we have interactions and that we can use to train the collaborative model, and split B represents cold items that we use only for testing the weights. All reported results for pure CBF and CFeCBF are reported on split B.

4 Experimental study A: Offline experiment

In this experiment, we investigate offline recommendation in cold- and warm-start scenarios. The specific experimental setup is presented in the following section.

Table 2: Characteristics of the evaluation dataset used in the offline study: $|\mathcal{U}|$ is the number of users, $|\mathcal{I}|$ the number of items, $|\mathcal{R}|$ the number of ratings.

ML-20M	$\lvert\mathcal{U}\rvert$	$\lvert\mathcal{I}\rvert$	$\lvert\mathcal{R}\rvert$	$\frac{\lvert\mathcal{R}\rvert}{\lvert\mathcal{U}\rvert}$	$\frac{\lvert\mathcal{R}\rvert}{\lvert\mathcal{I}\rvert}$	$\frac{\lvert\mathcal{R}\rvert}{\lvert\mathcal{I}\rvert \cdot \lvert\mathcal{U}\rvert}$ (density)
Train (A)	138 K	12,6 K	10 M	72.46	793.65	0.0057
Test (B)	3 K	4,8 K	212 K	70.67	44.16	0.0144

4.1 Data

We evaluated the performance of the proposed MRS on the MovieLens-20M (ML-20M) dataset [47], which contains user-item interactions between users and an up-and-running movie recommender system. We employ 5-fold cross-validation (CV) in our experiments by partitioning the items in our dataset into 5 non-overlapping subsets (item-wise splitting of the user-rating matrix). Different folds will have different cold items. Similar to [3], we built the test split by randomly selecting $3,000$ users, each having a minimum of 50 ratings in their rating profile, in order to speed up the experiments on the many feature sets. The items those users interacted with will be considered *cold items*; see split B in Fig. 5. The remaining items and interactions will be part of the training set. The reported results are referred to split B. Meanwhile, split A is used to perform parameter tuning. The characteristics of the data split are shown in Table 2. The significantly higher number of ratings per item in the training set (A) is due to the fact that it contains more users, and hence more interactions, than the test set (B).

4.2 Objective evaluation metrics

For assessing performance in the offline experiments, we compute the two categories of metrics, accuracy metrics (cf. Section 4.2.1) and beyond-accuracy metrics (cf. Section 4.2.2). The name and definition of the specific metrics computed is provided in the corresponding sections.

4.2.1 Accuracy metrics

Mean average precision (MAP) is a metric that computes the overall precision of a recommender system, based on precision at different recall levels [67]. It is computed as the arithmetic mean of the average precision (AP) over the entire set of users in the test set, where AP is defined as follows:

$$AP = \frac{1}{\min(M, N)} \sum_{k=1}^{N} P@k \cdot rel(k) \tag{6}$$

where $rel(k)$ is an indicator signaling if the k^{th} recommended item is relevant, i.e., $rel(k) = 1$, or not, i.e., $rel(k) = 0$; M is the number of relevant items; and N is the number of recommended items in the top N recommendation list. Note that AP implicitly incorporates recall, because it considers relevant items not in the recommendation list. Finally, given the AP equation, MAP will be defined as follows:

$$MAP = \frac{1}{|U|} \sum_{u \in |U|} AP_u \tag{7}$$

Normalized discounted cumulative gain (NDCG) is a measure for the ranking quality of the recommendations. This metric was originally proposed to evaluate the effectiveness of information retrieval systems [51]. It is nowadays also frequently used for evaluating music recommender systems [72,86,108]. Assuming that the recommendations for user u are sorted according to the predicted rating values in descending order, DCG_u is defined as follows:

$$DCG_u = \sum_{i=1}^{N} \frac{r_{u,i}}{log_2(i+1)} \tag{8}$$

where $r_{u,i}$ is the true rating (as found in test set T) for the item ranked at position i for user u, and N is the length of the recommendation list. Since the rating distribution depends on users' behavior, the DCG values for different users are not directly comparable. Therefore, the cumulative gain for each user should be normalized. This is done by computing the ideal DCG for user u, denoted as $IDCG_u$, which is the DCG_u value that provides the best possible ranking, obtained by ordering the items by true ratings in descending order. Normalized discounted cumulative gain for user u is then computed as follows:

$$NDCG_u = \frac{DCG_u}{IDCG_u} \tag{9}$$

Finally, the overall normalized discounted cumulative gain $NDCG$ is computed by averaging $NDCG_u$ over the entire set of users.

4.2.2 Beyond-accuracy metrics

The purpose of a recommender system is not only to recommend relevant items to the user based on their past behavior but also to facilitate exploration of the catalogue, helping to discover new items that the user might find interesting. Beyond-accuracy metrics try to assess if the recommender is able to diversify its recommendations for different users and leverage the whole catalogue or if it is focused on just a few highly popular items. In this study, we focus on the following measures:

Coverage of a recommender system is defined as the proportion of items which have been recommended to at least one user [48]:

$$coverage = \frac{|\hat{I}|}{|I|} \tag{10}$$

where $|I|$ is the cardinality of the test item set and $|\hat{I}|$ is the number of items in I which have been recommended at least once. Recommender systems with lower coverage are limited in the number of items they recommend.

Intra-list Diversity is another important beyond-accuracy measure. It gauges the extent to which recommended items are different from each other, where difference can relate to various aspects, e.g., genre, style or composition. Diversity can be defined in several ways. One of the most common is to compute the pairwise distance between all items in the recommendation set, either averaged [118] or summed [101]. In the former case, the diversity of a recommendation list L is calculated as follows:

$$IntraL(L) = \frac{\sum_{i \in L} \sum_{j \in L \setminus i} dist_{i,j}}{|L| \cdot (|L| - 1)} \tag{11}$$

where $dist_{i,j}$ is some distance function defined between items i and j. Common choices are inverse cosine similarity [91], inverse Pearson correlation [106], or Hamming distance [55]. In our experiments we report a diversity computed using the genre of the movies and cosine similarity.

Inter-list diversity or *inter-user diversity* measures the uniqueness of different users recommendation lists [117]. Given two users i and j, and their recommendation list L, the inter-list distance can be calculated by:

$$InterL(L_i, L_j) = 1 - \frac{q(L_i, L_j)}{|L|} \tag{12}$$

where $q(L_i, L_j)$ is the number of common items in recommendation lists of length $|L|$. $InterL(L_i, L_j) = 0$ indicates identical lists and $InterL(L_i, L_j) = 1$, completely different ones. The mean distance is obtained by averaging $InterL(L_i, L_j)$ over all pairs of users such that $i \neq j$.

A model which tends to frequently recommend the same set of items will result in similar recommendation lists and low diversity, whereas a recommender better able to tailor its recommendations to each user will exhibit

higher diversity [117]. In this respect, *inter-list diversity* and *intra-list diversity* are complementary. Consider a Top Popular recommender (i.e., one that recommends the most popular items). Its recommendations might have high *intra-list diversity* if they involve movies with different characteristics; therefore, a user will perceive them as diverse. However, all users will receive the same recommendations and both *item coverage* and *inter-list diversity* will be very low.

While an increase in diversity can indicate that the recommender is better able to offer personalized recommendations, it should be taken into account that the lowest diversity, and item coverage, will be obtained by always recommending the same items, whereas the highest will be obtained by a random recommender. This is another example of the accuracy-diversity trade-off.

In order to better understand how much the proposed techniques truly contribute towards more diverse and idiosyncratic recommendations across all users, in addition to the above beyond-accuracy metric, we also computed the metrics entropy, Gini coefficient, and Herfindahl (HHI) index [2]. These metrics provide different means for measuring *distributional dispersion* of recommended items across all users, and are therefore referred to as *aggregate diversity*. If recommendations are concentrated on a few popular items, the recommender will have low coverage and low diversity in terms of entropy and HHI but high Gini Index. If recommendations are more equally spread out across all candidate items, the recommender will exhibit high diversity and coverage but low Gini Index [2]. These metrics provide an overview of the recommender system from a system-wide point of view and are useful for assessing its behavior when deployed on a real, business-oriented system.

The distributional dispersion metrics are defined as follows:

$$Entropy = -\sum_{i \in I} \frac{rec(i)}{rec_t} \cdot \ln \frac{rec(i)}{rec_t} \tag{13}$$

$$Gini - index = \sum_{i=1}^{|I|} \frac{2i - |I| - 1}{|I|} \cdot \frac{rec(i)}{rec_t} \tag{14}$$

$$Herfindahl - index = 1 - \frac{1}{rec_t^2} \sum_{i \in I} rec(i)^2 \tag{15}$$

where $rec(i)$ refers to the number of times item i has been recommended over all users, rec_t the total number of recommendations (i.e., cutoff value times the number of test users), I the cold items set, and $|I|$ its cardinality. Note that while the Gini index and Herfindahl index have a value range between 0 and 1, Shannon entropy is not bounded by 1.

4.3 Collaborative filtering model

Following the results of [42] we chose as collaborative model RP3beta [87] which demonstrated a very competitive recommendation quality at a very

small computational cost, since it does not require ML. RP3beta is a graph-based algorithm which models a random walk between two sets of nodes, users and items. Each user is connected to the items he/she interacted with and each item is similarly connected to the users. The model consists of an item-item similarity matrix which represents the transition probability between the two items, computed directly via the graph adjacency matrix, easily obtainable from the URM. The similarity values are are elevated at a coefficient *alpha* and divided by each item's popularity elevated to a coefficient *beta*, the latter acting as a reranking phase which takes the popularity bias into account.

4.4 Hyper-parameter tuning

The proposed approach requires two types of parameter tuning. Firstly, it is necessary to train and tune the CF model. Since we want a single optimal hyper-parameter set we train the CF recommender on all the train folds separately and then select the hyper-parameters corresponding to the best average recommendation quality on all folds, measured with MAP. This constitutes a robust validation and testing methodology, and reduces the risk to overfit. Each fold will be associated with its own collaborative model since different folds will correspond to different cold items split. Secondly, the tuning of the hyper-parameters of the feature weighting machine learning is performed in a similar way, again optimizing MAP. We searched the optimal hyper-parameters via a Bayesian search [6] using the implementation of Scickit-optimize[13]. As for different aggregation methods designed for the audio and visual features, we chose the best performing ones with regards to the metric under study.

4.5 Overall computational time and complexity

In this section, we provide general information regarding runtimes and overall computational complexity of the subsystems in the proposed framework.

Regarding the extraction of the visual features, this process performs above the real-time frame rate of the movies (25 or 30 frames per second). We have performed feature extraction on a computer with Intel Xeon E5-1680 processor with 8 cores, 16 threads and a base frequency of 3.00 GHz, 192 GB RAM and an NVIDIA 1080TI GPU card with 3584 CUDA cores. While the extraction of AlexNet features was handled by the GPU, with an average speed of 62.8 processed frames per second, the extraction of the aesthetic visual features was done on the CPU, in parallel, using 7 of the 8 available cores and recording an average speed of 38.3 processed frames per second.

The feature weighting phase has a low computational complexity as it requires, for each epoch, to compute the gradient for each collaborative similarity value and compute the prediction error by using the item features. It is therefore linear in terms of both the number of descriptors and in terms of the number of similarities which in turn grows quadratically on the items.

[13] `https://scikit-optimize.github.io/`

In terms of runtime, on an Intel Xeon E3-1246 3.50GHz with 32GB RAM, learning the weights on the descriptors of length 200 takes 15 minutes on a single core, including the time required to perform the validations needed by early stopping.

4.6 Performance analysis: accuracy metrics

The experiments performed in Study A can be divided into four different categories, as presented in Table 3: *baseline* experiments using the genre and cast/crew metadata features, both editorially created (cf. Section 3.1.3);[14] *unimodal* experiments using traditional and state-of-the-art (SoA) audio and visual features (cf. Section 3.1); *content-based multimodal* experiments, where the proposed canonical correlation analysis (CCA) is used as an early fusion method (cf. Section 3.2); and finally, *collaborative-filtering enhanced multimodal* experiments, where the systems from the previous multimodal experiments are enhanced through the use of collaborative filtering (cf. Section 3.3). In the latter two, multimodal, categories, we report and analyze the performance of all combinations from the proposed unimodal features and the genre baseline[15].

As a general observation, we see that the *unimodal* visual and audio features constantly outperform the baseline metadata systems. The best performance is obtained by Deep visual features, improving the genre baseline by 53.0% in terms of NDCG and by 42.8% in terms of MAP. Even the lowest performing unimodal feature, i.e., i-vector, still achieves a 14.4% increase for NDCG and a 7.1% increase for MAP over the baseline. We further observe that the Deep feature outperforms the traditional AVF feature in the visual category, while in the audio category, the reverse pattern occurs, i.e., the traditional BLF feature has a better performance than the i-vector audio feature for both metrics.

As presented in Section 3.2, our *multimodal* approaches use CCA as a fusion method. We compared the CCA approach with a simple concatenation method, as well as with a weighted late fusion Borda count method, as described in [22]. We chose CCA as our early fusion method because all results were better for the CCA approach. For example, in the case of the i-vec + genre multimodal combination, CCA achieved a 9.5% MAP increase and a 20.2% NDCG increase over the simple concatenation method in the pure CBF approach, while in the CFeCBF approach, the CCA fusion method achieved a 151.6% increase in terms of MAP and a 181.8% increase in terms of NDCG. These results confirm not only that CCA fusion produces good results on its

[14] Note that we could not use tags as a feature in Study A, since the tags available in this dataset are user-generated. For cold items, no interactions with users have occurred yet, so no tags could be provided. While it could be possible the users added tags without providing a rating, this does not solve the underlying problem as it presumes a kind of interaction. Therefore the available tags for each items will be related to its popularity, some items will acquire tags easily while others may have none for quite some time.

[15] Note that we use the genre features as the main baseline due to their widespread usage and the fact that genre and cast had similar performance in almost all reported metrics.

own but also that it increases the power of collaborative filtering approaches by heavily reducing the size of the feature vector. Furthermore, the use of an early fusion method such as CCA allows us to easily create systems that outperform the late fusion method mentioned in [22], in both accuracy metrics.

For the *multimodal CBF approach*, we observe that the CCA fusion of the best performing unimodal audio and visual features (i.e., Deep and BLF) leads to the best multimodal results. More precisely, Deep + BLF achieves a 22.8% improvement over the baseline (0.0102 vs. 0.0083) in terms of NDCG and a 26.1% increase in terms of MAP (0.0053 vs. 0.0042). Similarly, the combination i-vec + genre performed strongly, improving on the baseline by 21.6% for NDCG (0.0101 vs. 0.0083) and 9.5% for MAP (0.0046 vs. 0.0042). This result was surprising, since both individual features, genre and i-vec, had a weaker performance in the unimodal experiment. In fact, in all genre combinations, such as AVF + genre, BLF + genre, and i-vec + genre, we can see an improvement in performance. This suggests that the genre feature has an information-complementary nature with other modalities, which can be leveraged using the CCA fusion. However, the combination of Deep + genre is an exception, as one can observe a decrease in performance. This may be due to the correlation between the two.

The *multimodal CFeCBF approach* aims to enable the recommendation of cold items by leveraging collaborative knowledge of warm items. The proposed method was applied on CCA multimodal approaches, as presented in the CBF multimedia approach. Looking at the performance globally, one can observe that the CFeCBF multimodal approach *improves the pure CBF multimodal systems* in all 10 combinations along NDCG and in 8 combinations along MAP; the few non-improved feature combinations, i.e., AVF + BLF and Deep + BLF, already performed well in pure CBF experiments. For NDCG, the average growth factor is 67%, with the minimum equal to 7% for Deep + BLF and the maximum equal to 123% for AVF + Genre. For MAP, the average growth factor is 68%, with the minimum equal to -7% for AVF + BLF and the maximum equal to 148% for AVF + Genre. When compared with *the genre baseline*, the proposed CFeCBF method improves the features, on average, by 79.75% for MAP and 72.6% for NDCG.

One final step was taken for the validation of these results, namely performing the significance tests as pairwise comparisons between the best performing systems and the best performing baseline genre. For both NDCG and MAP metrics, we performed statistical significance tests using the multiple comparison test provided by the statistical and machine learning toolbox in MATLAB[16] (function *multcompare*()), in which we adopted Fisher's least significant difference to compensate for multiple tests when performing all pairwise comparisons. Detailed information about the test can be found in [99]. The three best performing systems, *i-vec + genre*, *AVF + genre*, and *AVF + Deep*, show significant improvements over the baseline with $p < 0.05$, where the improvement along NDCG is 124.1%, 131.33%, and 130.12%, respectively, and that along MAP equal to 85.7%, 130.12%, and 130.12%, respectively. These results indicate the effectiveness of the proposed approach in dealing

[16] `https://www.mathworks.com/help/stats/multiple-comparisons.html`

with very different kinds of features and its ability to embed collaborative knowledge in a CBF recommender. In particular, the systems showing significant improvements have lower dimensionality for the descriptors than the others. This suggests that learning feature weights becomes harder as the number of dimensions increases. Applying dimensionality reduction techniques is therefore beneficial when dealing with very long descriptors.

4.7 Performance analysis: beyond-accuracy metrics

In this section, we report the results for beyond-accuracy metrics. The results are summarized in Table 4 (reports diversity metrics computed on the various recommendation lists: *inter-list diversity* and *intra-list diversity*) and Table 5 (reports all the *aggregate diversity* metrics, which are instead computed on the overall number of times each item was recommended to any user: *Item coverage*, *Shannon entropy*, *Gini index*, and *Herfindahl Index*).

From Table 4, we can observe that *intra-list diversity* (intraL) exhibits similar values across all cases. As previously mentioned, this diversity is computed with respect to the genre of movies, so a higher diversity would mean recommendations of heterogeneous genres, while a lower diversity would mean recommendations of the same genre. Following this definition, we expect that a recommender based only on genre as a feature will exhibit the lowest intraL diversity, which is in fact what we do observe. If we consider that as baseline value, we can see that all other features — metadata, unimodal or multimodal — achieve slightly higher diversity while not penalizing recommendation accuracy; this increase is significant in all cases. In terms of *inter-list diversity* (interL), results are more varied. We can see that multimodal recommenders, both pure CBF and hybrid CFeCBF, yield higher diversity in most cases, meaning that given any two users, the average number of items they have in common in their recommendation lists is going to be lower. The increased InterL diversity for CFeCBF is statistically significant in almost all cases. This suggests that multimodal recommenders will be less prone to concentrate their recommendations on a small subset of items.

From Table 5, we can see the results for *aggregate diversity* metrics. Note that while greater diversity will result in higher values for *Item coverage*, *Shannon entropy*, and *Herfindahl Index*, it will drive *Gini index* closer to zero. These metrics allow us to look at the recommender from the point of view of the whole system instead of that of the user, which is important when deploying recommenders as a part of a business model. We first focus on *Item coverage*, which tells us the portion of cold items the system was able to recommend. We can immediately see that the baseline recommenders using metadata have poor coverage: only half of the available items were recommended at least once. Most models based on multimodal features, instead, exhibit significantly higher coverage — up to more than 90%, meaning they are able to explore the catalogue much better without sacrificing recommendation quality. The other metrics measure the number of times each item has been recommended. Compared to the coverage, they provide the additional information about the number of

occurrences. Within a certain coverage value, the distribution of items can be very different. For example, in the case of a Top Popular recommender in a warm item scenario, the final coverage will be higher than the length of the recommendation list because some users will already have already interacted with those items and therefore other, less popular, items will be recommended to them. Distribution diversity metrics allow us to determine the extent to which the recommender is trying to diversify its recommendations. As an example, consider the 4 cases having coverage between 94.5% and 96.5%, with an interval of just 2% of all items. These cases exhibit a Gini index varying between .65 to .78, meaning that there is a difference in the number of times those items were recommended. In particular, the increase in coverage was accompanied in this case by more unbalanced, and therefore less diverse, item occurrence.

We can see how there is a significant difference between Multimodal and Base recommenders in terms of Gini index, meaning that the multimodal recommenders, both pure and hybrid, have more balanced item distribution. The combination of very high item coverage and improved distributional diversity metrics suggest that the collaborative machine learning step does not add a popularity bias to the feature weights, on the contrary CFeCBF is less subject to it than the Base recommenders. Moreover, we see that Shannon Entropy increases, meaning that the recommender is getting less "predictable" in the recommendations it will provide. This confirms what was observed in terms of interL diversity. The Herfindahl index is known to have a small value range when applied to recommender systems, as we can see in our experiments where its value ranges from .96 to .99. Compared to the other indices, it is less sensitive to items being recommended only a few times, due to its quadratic nature, but more sensitive to items being recommended a high number of times. Its values confirm the increased diversity achievable by Multimodal recommenders in almost all cases for pure CBF and in all cases for hybrid CFeCBF.

4.8 Cold to warm item transition

While the core of our experimental study is aimed at cold start items, in a real case scenario we expect some interactions to become available over time as the users interact with the cold items. For practical use it is interesting to assess when it is appropriate to change the recommendation model from a content based, either pure or CFeCBF, to a collaborative model. To this end we design a brief study, aiming to assess at which interaction density an item transition from cold to warm, allowing the use of CF methods.

It is already well known that, depending on the dataset, even a few interactions may be sufficient to outperform CBF approaches [88].

4.8.1 Experimental protocol

To simulate a realistic cold to warm transition we add some interactions to the cold items. Those interactions are taken from the original test set of that

fold. Since this study requires to create a new data split, with a denser train and a sparser test set, the results here reported are not comparable to the ones reported in the previous study.

We report two different experimental settings, one preserves the popularity distribution of the items, the other does not. The reader should notice that, being sampled in different, ways, the test set of the two experiments are different and the results are not directly comparable.

Random sampling In order to preserve the statistical distribution of the interactions and the impact of the item's popularity, the new train interactions for the cold items are randomly sampled, with no constraints applied. This will result in a mixture of popular items having a few interactions and unpopular items having none. This experiment allows to assess what happens in a realistic case in which some cold items will be popular and therefore collect interactions much faster, while others will not. This is motivated by the fact that CF algorithms, which CFeCBF is learning from, are sensitive to the popularity distribution and altering such distribution will result in biased CF models. The original test data is sampled so that 2% of its interactions become new train data and 98% constitute the new test data. To show the behaviour at different densities, the train data is further divided in a smaller set only containing 0.5% of the original test interactions.

Fixed number of interactions While the previous experiment models a real case scenario more accurately, it leaves open the question of how significant is the effect of the popularity bias on the results. To this end also build a different split which contains a fixed number of train interactions for the cold items. This creates an artificial popularity distribution which will change the behaviour of the CF model. The number of interactions we chose is 1 and 5. This will result in a perfectly balanced train set. In this case the test data is composed by the original test data minus 4 interactions for each item.

This new train data is therefore composed by the original train data plus the interactions sampled from the test set and is used to train all algorithms: CBF, CF and CFeCBF. The optimal parameters remain those selected in the previous phase when no interactions were available. In a real case scenario it would be impractical to run a new tuning of the model's parameters after each few interactions are added. It is instead more realistic for this tuning phase to be executed again only once a sufficient amount of new data is available.

4.8.2 Result discussion

The results for the random split are reported in Table 6 for both accuracy metrics and Item Coverage.[17] As it is possible to see, in terms of accuracy metrics the recommendation quality of pure CBF remains constant as the transition progresses. CFeCBF, instead, changes its recommendation quality, in some cases improving over the cold item case, in others not. This is due to the evolving CF model it is learning from.

[17] For brevity we did not report all beyond accuracy metrics.

The most important thing to observe is that the pure collaborative algorithm, RP3beta, is immediately able to outperform all CBF and CFeCBF models in terms of accuracy metrics. It should be noted that Movielens, the dataset from which the interactions are taken, tends to exhibit high recommendation quality for collaborative algorithms which makes this cold to warm transition very fast. Consider that *Warm 0.5%* corresponds to an average of $1 \cdot 10^{-}1$ interactions per item and *Warm 2.0%* of $4 \cdot 10^{-}1$ interactions per item. Looking at the recommendation quality alone is however misleading. In terms of diversity it is possible to see that CF has a remarkably low item coverage. This means that the CF algorithm is still not able to explore the catalogue, being confined to a marginal 6% of the available items. The result can be explained by the significant popularity bias of the dataset, hence a few items account for a sizable quota of the interactions, while many others have much fewer. This behaviour means that the CF model is recommending only a few popular items, being unable to recommend the vast majority of them. CF fails completely to allow the user a broad exploration of the catalogue and offers very little personalization. Moreover, if the items are not seen by the users, it will be very difficult to collect the interactions needed for them to become warm items, the risk being to keep them in a cold state for very long. CFeCBF, on the other hand, has a very high Item Coverage, which allows a broader exploration of the catalogue, yielding to a higher probability cold items will be rated and a more effective CF model could be applied at a later stage.

If we look at the results for the fixed number of interactions experiment in Table 7, we can observe a different behaviour. The CBF and CFeCBF models maintain their almost stable recommendation quality while CF increases. However, as opposed to the previous case, we can see that the CF advantage grows less steeply with respect to CFeCBF even though the train data is much denser, 1 and 4 as opposed to $4 \cdot 10^{-}1$. Moreover, the CF Item Coverage is comparable or higher than CFeCBF. This allows to state that the behaviour of the CF algorithm in the random sampling experiment is strongly influenced by the significant popularity bias of the dataset.

To summarize, in terms of accuracy metrics CF algorithms are able to outperform CBF and CFeCBF when even just a few interactions are available, more so if the dataset has a strong popularity bias. However CBF and CFeCBF maintain a sizable advantage in terms of diversity metrics and Item Coverage. Depending on the specific use-case or application, and therefore the desired balance between accuracy and catalogue exploration, a different strategy may be adopted. If the main focus is on accuracy, then as soon as the item has an interaction it can be considered as warm. The reader should note that, while Movielens has a high popularity bias, other datasets with a less pronounced bias will exhibit a less steep CF quality improvement. If the focus is on improving catalogue exploration to reduce the popularity bias effect then the target number of interactions per item may be pushed further.

5 Experimental study B: Insights from a preliminary user study about perceived quality

In this section, we describe an empirical study whose goal is not to recommend new movies, as in the experimental study A, but to understand to what extent the proposed movie genome is *perceived* as useful when deployed in a real MRS. The developed system uses a pure CBF recommender based on the KNN algorithm and measures the utility of the recommendation as perceived by the user in terms of *accuracy*, *novelty*, *diversity*, *level of personalization*, and overall *satisfaction*. In this study, we intentionally avoid the discussion of hybridization and focus instead only on six unimodal recommendation approaches, classifiable in 3 categories: (i) metadata: *genre* and *tag*, (ii) audio: *i-vectors* and *BLF*, and (iii) visual: *Deep features* and *AVF*. We use only the unimodal recommendation schemes presented in the experimental study A. The reason for this is to avoid overloading users with too many recommendation choices, and thus to be able to obtain more reliable responses from users collectively. Note that in this study, the tags feature is considered because, as stated, the study's focus is no longer on new movie recommendation (as in study A) and tags serve as a rich semantic baseline.

Our preliminary studies in a similar direction have been published in [39] which focused on a single visual modality [39], and in [22], which used a lower number of participants (74 vs. 101). In addition, compared to [22], we performed better sanity checks and removed unreliable user input. Further information is provided in the following sections.

5.1 Perceived quality metrics

The goal of the current study is to measure how the user perceives the quality of the proposed recommender system. Perceived quality is as an indirect indicator of a recommenders potential for persuasion [18]. It is defined as the degree to which the users judge recommendations positively and appreciate the overall experience of the recommender system. We operationalize the notion of perceived quality in terms of five metrics [37]: *Perceived accuracy* (also called *Relevance*) — measures how much the recommendations match users' interests, preferences, and tastes; *Satisfaction* — measures global users' feelings about their experience with the recommender system; *Understands me* — relates to perceived personalization or the user's perception that the recommender understands their tastes and can effectively adapt to them; *Novelty*[18] — measures the extent to which users receive new (unknown) recommendations; *Diversity* — measures how much users perceive recommendations as different from each other, e.g., movies from different genres.

[18] Note that we could not use Novelty as an evaluation criterion in study A because Novelty is defined in terms of item popularity, which is available for warm items but not for cold items.

5.2 Evaluation protocol

To measure the user's perception of the recommendation lists according to the five quality metrics explained above, we adopt the questionnaire proposed in [60]. This instrument contains 22 questions to assess various aspects of the recommendation lists. For convenience, these questions are shown in Table 8. As suggested by the authors from [37], the questions are asked in a comparative mode instead of seeking absolute values.

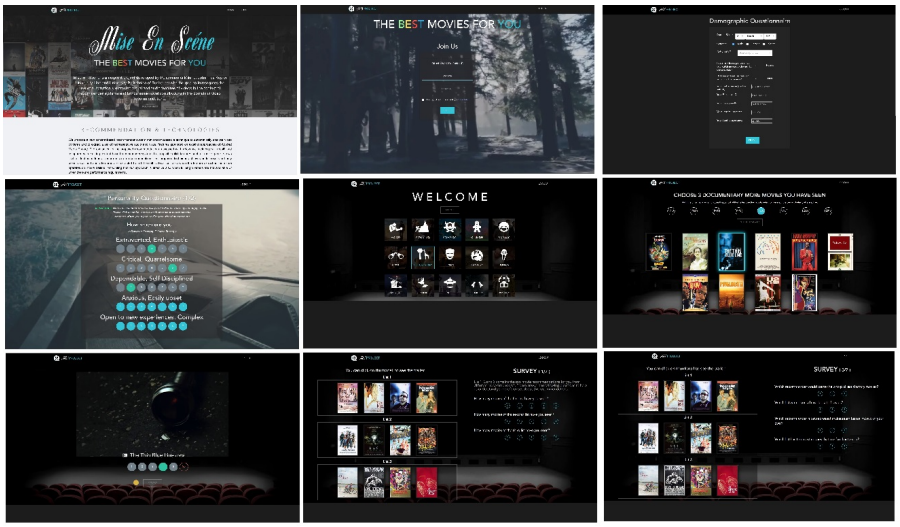

Fig. 6: Screenshots of the *MISRec* web application, designed for movie recommendation and empirical studies. The user needs to register, answer demographic and personality questionnaires, select his/her favorite genre, and rate some movies by looking at their trailers. Then, he/she is presented with 3 recommendation lists and a list of questions about perceived quality.

We developed *MISRec* (Mise-en-Scène Movie Recommender), a web-based testing framework for the movie search and recommendation domain, which can easily be configured to facilitate the execution of controlled empirical studies. Some screenshots of the system are presented in Figure 6. MISRec is powered by a pure CBF algorithm based on KNN and supports users with a wide range of functionalities common in online video-streaming services such as Netflix[19]. MISRec contains the same catalog of movies used in the first study (see Section 4). Users can browse the catalog of movies, retrieve detailed descriptions of each, rate them, and receive recommendations. MISRec also embeds an online questionnaire system that allows researchers to easily collect quantitative and qualitative information from the user. The first prototype of *MISRec* was used for conducting an empirical study on the con-

[19] `https://www.netflix.com`

tribution of stylistic visual features to movie recommendation, and the results were published in [39]. A more recent development of MISRec powered by the proposed movie genome features was published in [22]. An extension of the system was also developed in [32] to use the system in an interactive manner e.g., for kid movie recommendation using cover photos of the movies as the system activator.

Our main target audience is users aged between 19 and 54 who have some familiarity with the use of the web but have never used MISRec before the study (to control for the potentially confounding factor of biases or misconceptions derived from previous uses of the system). The total number of recruited subjects who also completed the task was 101 (73 male, 28 female, mean age 25.64 years, std. 6.61 years, min. 19 years, max. 54 years). Data collection were carried out mostly from master students at three universities: Politecnico Di Milano Italy, JKU Linz Austria and Politehnica di Bucharest, Romania attending the course of Recommender Systems or similarly related courses. They were trained to perform the study, were given written instructions on the evaluation procedure, and were regularly supervised by Ph.D. students and a PostDoc researcher during their activities. The interaction begins with a sign-up process, where each participant (user) is asked to specify his/her e-mail address, user name, and password (see Figure 6 top-middle). For users who wish to remain anonymous, we provide the option to conceal their true email address. Afterwards, the user is asked to provide basic demographics (age, gender, education, nationality, and number of movies watched per month, consumption channels, some optional social media IDs, such as Facebook, Twitter, and Instagram). After the user has registered for the system and provided his/her basic demographic information, he/she is asked to fill out the Ten-Item Personality Inventory (TIPI) questionnaire (see Figure 6 middle-left) so that the system can assess his/her Big Five personality traits (openness, conscientiousness, extroversion, agreeableness, and neuroticism) [79]. Then, for *preference elicitation* [15], the user is invited to browse the movie catalog from his/her favorite genre and to scroll through productions from different years in a user-friendly manner (see Figure 6 center and middle-right). The user initially selects four movies as his/her favorites.

The user can watch the trailers for the selected movies and provide ratings for them using a 5-level Likert scale (1 = low interest in/appreciation for the movie to 5 = high interest in/appreciation for the movie). The user can also report a movie (if the trailer is not correctly displayed) and the movie will be skipped (see Figure 6 bottom-left). After that, on the basis of these ratings and the content features described in Section 3.1, three categories of recommendation lists are presented to the user: (i) audio-based recommendation using BLF or i-vectors as features, (ii) visual-based recommendation using AVF or Deep as features, (iii) metadata-based recommendation using genre or tag as features. In each of the three recommendation categories, the recommendations are created using one of the two recommendation approaches (e.g., BLF or i-vectors for (i), and so on), chosen randomly. Since watching trailers is a time-consuming process, we decided to show only four recommendations in each of the three lists.

It is important to note that since we do not wish to overload the user with too much information, we avoid presenting him/her with six recommendation lists using all of the features. This would be the case in a *within-subject design*, where each subject uses all variants of the factorial designs simultaneously, i.e., six recommendation approaches in this case. Instead, we decided to use a *between-subject design*, where factorial designs are randomized for a given subject. Since our final goal is to have the user compare the **three** recommendation classes (i.e., audio *vs.* visual *vs.* metadata) at the same time, the way we implemented the between-subject design randomizes each of the two instances of each category for a given user. Therefore, each user compares one out of eight possible combinations: (BLF, AVF, genre), (i-vector, AVF, genre), (BLF, AVF, tag), (i-vector, AVF, tag) and so forth[20]. This gives us more flexibility in handling all this information and obtaining reliable responses. Finally, to avoid possible biases or learning effects, the positions of the recommendation lists are randomized for each user.

5.3 Results

In this section, we present the user-perceived accuracy, satisfaction, personalization, diversity, and novelty. Before analyzing the survey responses, we cleaned the data by removing users who did not complete the questionnaire. We also removed users who were too fast in giving answers (less than 15% of the median time of all users) since we do not consider these users reliable. As the results of these filtering steps, 21 users are filtered out. Furthermore, users were asked to specify how many of the movies in each recommendation list they have seen. A list is included in the analysis only if the user has seen at least one movie from it. For example, if a user chooses a list as the recommendation most accurately matching his/her taste but has previously specified that he/she has not seen any movie from that list, we discard that list from his/her responses.

We compute a score for each recommender/feature with respect to the five performance measures. When recommendation lists are presented to the user, he/she has to choose one list out of the three as an answer to each question (cf. Table 8). Each selected list counts for a vote for the respective recommender that has created the list. Note that answers/scores given to questions marked with a + contribute positively to the final score, whereas scores to questions marked with a - contribute negatively. Finally, all votes given to each recommender are summed along each dimension (performance measure) and expressed as percentages, i.e., the relative frequency with which each recommender has been selected as the best one. The final results for the five dimensions are presented in Table 9 and discussed below.

[20] Note that we could not use tag as a feature in Study A since tags are user-generated content. In cold items, no interactions with users have occurred yet; therefore, no tag could have been provided as a feature. Tags could be obtained via cross-domain techniques, but those are a vast research area and outside the scope of this paper. Tags could also be obtained by manual/editorial tagging, but that would be time-consuming and expensive, and therefore not suitable for a high rate of new items, which is the scenario of main interest for this paper.

Perceived Accuracy/Relevance: the following algorithms are perceived as the most accurate (relevant) by the subjects: tag, genre, and the SoA visual deep feature, with 26%, 25%, and 24% of the votes, respectively. User-generated tags are rich semantic descriptors and, as expected, the respective feature is evaluated the best by the subjects; however, the difference from genre and deep features remains very small (1 to 2 %). Meanwhile, the lowest performance is obtained by the traditional audio and visual features BLF and AVF with 3% and 8% of the votes, respectively. I-vector aggregates 13% of the votes. These results are in agreement with our expectations in that, as a standalone feature, the proposed *SoA feature, deep, and i-vector show the most promising results compared with traditional multimedia features*; e.g., Deep achieves a result of 24% in comparison with 8% for AVF, which represents an improvement of about 300%.

Understands Me and Satisfaction: the results of users' perceived personalization (captured by the questions in the "Understands Me" category) and the overall feeling of the experience with the recommender system (captured by the questions in the "Satisfaction" category) show superior performance for Deep and tag features, with 32% and 31% of the votes, while genre is ranked lower, with 24% of the votes. For user satisfaction, the best performance is perceived for tag, deep, and genre features, with 25%, 24%, and 24% of the user votes, respectively. The lowest performance is obtained by the traditional audio and visual features (between 7% and 10%). We can also note that the results along the above perceived quality metrics are highly correlated (Pearson's correlation coefficient is 0.9735). The only exception is audio, in which we can find a difference in two dimensions between the performance obtained by SoA i-vectors (compare 3.6% vs. 11%) and by traditional BLFs (compare 1.2% vs. 6%). The results of "Understands me" and "Satisfaction" are also highly correlated with perceived accuracy (Pearson's correlation coefficients are 0.9390 and 0.9897, respectively.). This can indicate that the users' perception of personalization and satisfaction is the same as accuracy and that users respond to the questions belonging to these categories in a similar way.

Diversity: the results for the perceived diversity indicate that the best performance is achieved by genre (29%) - substantially higher than i-vector, Deep, and tag, with 19%, 18%, and 16% of the votes, respectively. On the other hand, both traditional visual and audio features, AVF and BLF, show the lowest perceived diversity, attracting only 13% and 6% of the votes, respectively. The results for diversification are slightly different than those gained in our original user study [22] and show that users perceived recommendation by genre the most diverse (while perceived highly relevant too). Perhaps this is because users do not mentally compute list diversification based on genre diversity but also consider other attributes (e.g., the appearance of the DVD cover) when they are asked to indicate the most diversified recommendation list. Another reason could be that one of the questions explicitly asks for diversity of mood, and the same genre can have movies with very different moods (e.g., in sci-fi).

Novelty: results for novelty are surprising in several ways. Firstly, it is the traditional visual features, AVF, which have the highest amount of perceived novelty, gaining as much as 31% of votes, followed by the SoA audio and

visual features i-vector and deep with 21% and 19% of the votes, respectively. Meanwhile, the tag feature has attracted a very small amount, i.e., only 5%, of the scores for perceived novelty. Since tags are user-assigned, they have a high semantic content and capture something specific about the user perception of the movie. Therefore, similar tags may yield to recommendations not perceived as novel.

Globally, the results of our study on perceived recommendation quality indicate that perceived quality of recommendations is high for the SoA visual and audio features (Deep and i-vector) along most investigated performance measures. The exception is the user's perceived personalization ("Understands Me") for which i-vector performs poorly (but Deep visual performs best). For the remaining dimensions, these SoA features are ranked in the top 3 of all investigated features. Especially when it comes to novelty, SoA audio and visual features by far outperform metadata features. Overall, each feature has its merits, which again support our proposal for multimodal recommendation approaches.

6 Conclusions and future perspectives

In this work, we presented a framework for *new movie recommendation* by exploiting *rich item descriptors* and a *novel recommendation model*. We compared our system to some standard metadata-based methods that use genres and casts (editorial metadata). Specifically, the proposed system integrates multimedia *aesthetic visual features* and *audio block-level features*, as well as novel, state-of-the-art *deep visual features* and *i-vector audio features*, together with genre and cast features, all of which are referred to as the *movie genome*. For exploiting the complementary information of different modalities, we proposed CCA to fuse movie genome descriptors into shorter and stronger descriptors. Lastly, we presented a novel recommendation model that leverages a two-step approach named collaborative-filtering-enriched content-based filtering (CFeCBF). It exploits the collaborative knowledge of warm items (videos with interactions) to weight content information for cold items (videos without interactions) and improve the ability to recommend cold videos, for which interactions and user-generated content are rare or unavailable. The proposed system represents a practical solution for alleviating the CS problem, in particular, the extreme CS new item problem, where newly added items lack any interaction and/or user-generated content.

6.1 Discussion of the results

For evaluation, we conducted two empirical studies: (i) a system-centric study to measure the offline quality of recommendations in terms of *accuracy* (NDCG and MAP) and *beyond accuracy* (list diversity, distributional diversity, and item coverage) (cf. Section 4); (ii) a preliminary, user-centric online experiment to measure different subjective metrics, including relevance, satisfaction, and diversity (cf. Section 5). In both studies, we used a dataset of more than 4,000

movie trailers, which makes our approach more versatile, because trailers are more readily available than full movies.

In the first study, visual and audio features generally outperform the metadata features with respect to the two tested accuracy measures, with an average growth factor of 32% along NDCG (min 14% and max 53%) and 23% along MAP (min 7% and max 42%). The real improvement, however, is in the final system performance, in which *the proposed system outperforms the baseline by a substantial margin of 80% along NDCG and 73% along MAP* and also outperforms the simpler multimodal recommender model using CCA in a pure CBF system by 67% for NDCG and 68% for MAP. These results are promising and indicate the capability of our recommendation model to improve the utility of new item recommendation by leveraging rich CF data for existing warm items and utilizing them as feature weights to improve the content information in pure CBF.

Moreover, in terms of beyond-accuracy measures, we can see that the genre-based recommender exhibits the lowest diversity, as could be expected. In addition, our results show that the multimodal recommender is able to provide substantially higher coverage and improved distributional diversity on all reported metrics. This means that a multimodal recommender is less prone to popularity bias; in particular, *multimodal recommendations generated by our CFeCBF model show a significant improvement along (almost) all reported beyond-accuracy metrics, while not penalizing the accuracy and even improving it substantially.*

When an item transition from cold to warm we can see that CF is able to outperform CFeCBF very soon in terms of accuracy metrics on a dataset with significant popularity bias, while CFeCBF still exhibit much better ability to leverage all the available items. The strength of the two algorithms may be combined allowing to exploit the superior recommendation quality of CF for warm items and the much greater coverage of CFeCBF to recommend cold items, whose low popularity renders the transition to warm slower.

In the user study, results show that the perceived recommendation for state-of-the-art visual (Deep) and audio (i-vector) features are meaningful. With the exception of the user's perceived personalization, in which i-vector performed poorly, these audio and visual features are ranked in the top 3 of all investigated features. In some cases, such as for the perceived novelty, the improvement of these features over metadata was significantly high. *Overall, the results of the user study show that each feature has advantages and supports our proposal for multimodal recommendation approaches.*

6.2 Answers to research questions

RQ1: Can the exploitation of movie genome describing rich item information as a whole, provide better recommendation quality compared with traditional approaches that use editorial metadata such as genre and cast in CS scenarios? As the experiments have shown, multimedia features can provide a good alternative to editorial metadata such as genre and cast in terms of both accuracy and beyond-accuracy measures. The use of multimedia features can

allow to increase the recommendation quality in terms of accuracy while also improving the ability of the recommender to leverage the whole catalogue of items.

RQ2: Which visual and audio information better captures users' movie preferences in CS scenarios? The most important improvement for the accuracy metric was achieved by exploiting the state-of-the-art deep features for the visual modality but *traditional* block-level features for the audio modality.

RQ3: Could we leverage user interaction to enrich cold item information? We proved that it is possible to effectively leverage user interactions and enrich the item descriptors by learning a set of feature weights associated with the descriptors. This would result in improving the recommendation quality of cold items over current editorial baselines (genre and cast).

6.3 Limitations

Recommendation model. The proposed recommender model has a few limitations. Firstly, since it leverages item features, the quality and noisiness of item features have an impact on the ability to learn good feature weights. If an item has too few features, the resulting recommendations will exhibit limited diversity and the weights might embed some popularity bias. This is visible in Table 4 for AVF + Genre, which, while having good recommendation quality, exhibits lower InterL diversity with respect to the other cases. On the other hand, if the number of features is too high, the number of collaborative similarities might not be enough to ensure good weighs are learned.

Secondly, as the model leverages a collaborative model, this feature weighting scheme will not be applicable to any scenario. If the user-item interactions are too few, it is well known that the collaborative model will perform poorly in comparison to a pure CBF recommender. If this is the case, the learned weights will be approximating a poor collaborative model and therefore the resulting recommendations will not improve. Even so, however, it may still be possible to leverage a collaborative model on a smaller and denser portion of the dataset to learn only some of the weights. This is an aspect that can be studied more in detail.

Thirdly, in the case of Boolean features, CFeCBF is sensitive to items with very sparse features due to the fact that it can learn weights only for features available for cold items. Feature sparsity has the dual effect of both increasing the probability of new items having many new features, previously unobserved, and reducing the degree of freedom of the model.

Finally, in our previous study [27], we concluded concluded that trailers and movies share similar characteristics in the recommendation scenario. However, the dataset used in [27] was rather small (167 full movies and corresponding trailers were used for comparison). Also, the number of visual features was limited (only five features, cf. [89]). Due to these restrictions, the generalizability of our findings in [27] might be restricted. Nevertheless, we argue that

using trailers instead of full-length movies serves as a good proxy and has several advantages: trailers are accessible, are sensibly shorter that the entire movie, and preserve the main idea of the movie since they are designed to trigger the viewers interest in watching the entire movie. Results in the paper at hand show that the performance recommendation system that exploits movie genome is better in comparison with editorial metadata (using genre or cast). We believe this can be seen as a breakthrough to demonstrate that they can effectively replace the full movies. Lastly, depending on the strength of the video descriptors with respect to the CF information, the items may transition from cold to warm after even a single interaction. In popularity biased datasets a premature switch from CFeCBF to CF may result in poor catalogue exploration and therefore limited overall recommender effectiveness. This effect can be minimized by adopting strategies to allow a gradual switch between the two allowing the less popular items more time to collect the interactions they need to become warm, while benefitting from the higher recommendation quality of a CF for warm items. The choice of an optimal point where to switch between CFeCBF and CF remains challenging.

User study limitations. The reported user study results should be considered preliminary. In fact, given the relatively low number of participants, the results may not be statistically significant. Given the complexity of the questionnaire, which takes more than half an hour to complete, as well as due to the specificity of the movie dataset used, i.e., the movies tend to be classic ones not easily available to the younger generation, it is very difficult to find reliable users and motivate them to participate in the study, even when considering a paying platform such as crowdsourcing.

6.4 Future perspectives

We believe our proposed movie recommendation framework can pave the way for a new paradigm in new product recommendation by exploiting CFeCBF models built on top of rich item descriptors extracted from content. Examples of such products include fashion (images), music (audio), and tourism (both images and audio) and generic videos. As a related future research line, we would like to understand in what ways affective metadata (metadata that describe the user's emotions) can be used for CBF of videos/movies, similar to the research [104] carried out for images.

Regarding the carried out user study, currently it involves 101 subjects. This is while according to [59], approximately 73 subjects are necessary in every configuration to ensure statistical significance of results (i.e., about 600 subjects in total). This is an important limitation of our current work, which we plan to overcome in the future by hiring a larger number of *reliable subjects*. Furthermore, we plan to validate the generalization power of our new movie recommender model on video datasets of a different nature, such as full-length movies, movie clips and user-generated videos. An initial attempt at the former was published in our work [22] and at the latter in [21], whose authors plan to

release a publicly available dataset of movie clips. Part of these data is used in the MediaEval 2018 task "Recommending Movies Using Content".[21]

Last but not least, a feature analysis will be conducted to better understand how movie genome features contribute to the success of the combined features as part of future work.

Acknowledgement

This work was supported by the Austrian Ministry for Transport, Innovation and Technology, and the Ministry of Science, Research and Economy, and the Province of Upper Austria in the frame of the COMET center SCCH. The work of Mihai Gabriel Constantin and Bogdan Ionescu was partially supported by the Ministry of Innovation and Research, UEFISCDI, project SPIA-VA, agreement 2SOL/2017, grant PN-III-P2-2.1-SOL-2016-02-0002.

References

1. Cisco visual networking index: Global mobile data traffic forecast update, 2015-2020 white paper. http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html. Accessed: 2016-12-1
2. Adomavicius, G., Kwon, Y.: Improving aggregate recommendation diversity using ranking-based techniques. IEEE Transactions on Knowledge and Data Engineering **24**(5), 896–911 (2012)
3. Adomavicius, G., Zhang, J.: Stability of recommendation algorithms. ACM Transactions on Information Systems (TOIS) **30**(4), 23 (2012)
4. Aggarwal, C.C.: Content-based recommender systems. In: Recommender Systems, pp. 139–166. Springer (2016)
5. Aggarwal, C.C.: Evaluating recommender systems. In: Recommender Systems, pp. 225–254. Springer (2016)
6. Antenucci, S., Boglio, S., Chioso, E., Dervishaj, E., Shuwen, K., Scarlatti, T., Ferrari Dacrema, M.: Artist-driven layering and user's behaviour impact on recommendations in a playlist continuation scenario. In: Proceedings of the ACM Recommender Systems Challenge 2018 (RecSys Challenge '18) (2018)
7. Asmaa Elbadrawy, G.K.: User-specific feature-based similarity models for top-n recommendation of new items. In: ACM Transactions on Intelligent Systems, 2015, vol. 6 (2015). DOI 10.1145/2700495
8. Bartolini, I., Moscato, V., Pensa, R.G., Penta, A., Picariello, A., Sansone, C., Sapino, M.L.: Recommending multimedia objects in cultural heritage applications. In: International Conference on Image Analysis and Processing, pp. 257–267. Springer (2013)
9. Bernardis, C., Ferrari Dacrema, M., Cremonesi, P.: A novel graph-based model for hybrid recommendations in cold-start scenarios. In: Proceedings of the Late-Breaking Results track part of the Twelfth ACM Conference on Recommender Systems. ACM (2018). URL https://arxiv.org/abs/1808.10664
10. Bertin-Mahieux, T., Ellis, D.P., Whitman, B., Lamere, P.: The million song dataset. In: Proceedings of the 12th International Society for Music Information Retrieval Conference, pp. 591–596. Miami, USA (2011)
11. Bobadilla, J., Ortega, F., Hernando, A., Bernal, J.: A collaborative filtering approach to mitigate the new user cold start problem. Knowledge-Based Systems **26**, 225–238 (2012)

[21] http://www.multimediaeval.org/mediaeval2018/content4recsys

12. Bordwell, D., Thompson, K., Smith, J.: Film art: An introduction, vol. 7. McGraw-Hill New York (1997)
13. Bronstein, A.M., Bronstein, M.M., Kimmel, R.: The video genome. arXiv preprint arXiv:1003.5320 (2010)
14. Cella, L., Cereda, S., Quadrana, M., Cremonesi, P.: Deriving item features relevance from past user interactions. In: Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization, pp. 275–279. ACM (2017)
15. Chen, L., Pu, P.: Survey of preference elicitation methods. Tech. rep. (2004)
16. Chu, W.T., Tsai, Y.L.: A hybrid recommendation system considering visual information for predicting favorite restaurants. World Wide Web **20**(6), 1313–1331 (2017)
17. Cremonesi, P., Elahi, M., Deldjoo, Y.: Enhanced content-based multimedia recommendation method (2018). US Patent App. 15/277,490
18. Cremonesi, P., Garzotto, F., Turrin, R.: Investigating the persuasion potential of recommender systems from a quality perspective: An empirical study. ACM Transactions on Interactive Intelligent Systems (TiiS) **2**(2), 11 (2012)
19. Datta, R., Joshi, D., Li, J., Wang, J.Z.: Studying aesthetics in photographic images using a computational approach. In: European Conference on Computer Vision, pp. 288–301. Springer (2006)
20. Daubechies, I.: Ten lectures on wavelets. SIAM (1992)
21. Deldjoo, Y., Constantin, M.G., Dritsas, T., Schedl, M., Ionescu, B.: The mediaeval 2018 movie recommendation task: Recommending movies using content. In: MediaEval 2018 Workshop (2018)
22. Deldjoo, Y., Constantin, M.G., Eghbal-Zadeh, H., Schedl, M., Ionescu, B., Cremonesi, P.: Audio-visual encoding of multimedia content to enhance movie recommendations. In: Proceedings of the Twelfth ACM Conference on Recommender Systems. ACM (2018). DOI https://doi.org/10.1145/3240323.3240407
23. Deldjoo, Y., Constantin, M.G., Ionescu, B., Schedl, M., Cremonesi, P.: Mmtf-14k: A multifaceted movie trailer dataset for recommendation and retrieval. In: Proceedings of the 9th ACM Multimedia Systems Conference, MMSys, pp. 450–455 (2018)
24. Deldjoo, Y., Cremonesi, P., Schedl, M., Quadrana, M.: The effect of different video summarization models on the quality of video recommendation based on low-level visual features. In: Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing, p. 20. ACM (2017)
25. Deldjoo, Y., Elahi, M., Cremonesi, P.: Using visual features and latent factors for movie recommendation. CEUR-WS (2016)
26. Deldjoo, Y., Elahi, M., Cremonesi, P., Garzotto, F., Piazzolla, P.: Recommending movies based on mise-en-scene design. In: Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems, pp. 1540–1547. ACM (2016)
27. Deldjoo, Y., Elahi, M., Cremonesi, P., Garzotto, F., Piazzolla, P., Quadrana, M.: Content-based video recommendation system based on stylistic visual features. Journal on Data Semantics **5**(2), 99–113 (2016)
28. Deldjoo, Y., Elahi, M., Cremonesi, P., Moghaddam, F.B., Caielli, A.L.E.: How to combine visual features with tags to improve movie recommendation accuracy? In: International Conference on Electronic Commerce and Web Technologies, pp. 34–45. Springer (2016)
29. Deldjoo, Y., Elahi, M., Quadrana, M., Cremonesi, P.: Toward building a content-based video recommendation system based on low-level features. In: International Conference on Electronic Commerce and Web Technologies, pp. 45–56. Springer (2015)
30. Deldjoo, Y., Elahi, M., Quadrana, M., Cremonesi, P.: Using visual features based on mpeg-7 and deep learning for movie recommendation. International Journal of Multimedia Information Retrieval (2018)
31. Deldjoo, Y., Elahi, M., Quadrana, M., Cremonesi, P., Garzotto, F.: Toward effective movie recommendations based on mise-en-scène film styles. In: Proceedings of the 11th Biannual Conference on Italian SIGCHI Chapter, pp. 162–165. ACM (2015)
32. Deldjoo, Y., Frà, C., Valla, M., Paladini, A., Anghileri, D., Tuncil, M.A., Garzotta, F., Cremonesi, P., et al.: Enhancing childrens experience with recommendation systems. In: Workshop on Children and Recommender Systems (KidRec'17)-11th ACM Conference of Recommender Systems, pp. N–A (2017)

33. Deldjoo, Y., Schedl, M., Cremonesi, P., Pasi, G.: Content-Based Multimedia Recommendation Systems: Definition and Application Domains. In: Proceedings of the 9th Italian Information Retrieval Workshop (IIR 2018). Rome, Italy (2018)
34. Deldjoo, Y., Schedl, M., Hidasi, B., Kness, P.: Multimedia recommender systems. In: Proceedings of the 12th ACM Conference on Recommender Systems. ACM (2018). DOI 10.1145/3240323.3241620
35. Eghbal-Zadeh, H., Lehner, B., Dorfer, M., Widmer, G.: CP-JKU submissions for DCASE-2016: a hybrid approach using binaural i-vectors and deep cnns. Tech. rep., DCASE2016 Challenge (2016)
36. Eghbal-Zadeh, H., Schedl, M., Widmer, G.: Timbral modeling for music artist recognition using i-vectors. In: Signal Processing Conference (EUSIPCO), 2015 23rd European, pp. 1286–1290. IEEE (2015)
37. Ekstrand, M.D., Harper, F.M., Willemsen, M.C., Konstan, J.A.: User perception of differences in recommender algorithms. In: Proceedings of the 8th ACM Conference on Recommender Systems, RecSys '14, pp. 161–168. ACM, New York, NY, USA (2014). DOI 10.1145/2645710.2645737. URL http://doi.acm.org/10.1145/2645710.2645737
38. Elahi, M., Braunhofer, M., Gurbanov, T., Ricci, F.: User Preference Elicitation, Rating Sparsity and Cold Start, chap. Chapter 8, pp. 253–294. DOI 10.1142/9789813275355_0008. URL https://www.worldscientific.com/doi/abs/10.1142/9789813275355_0008
39. Elahi, M., Deldjoo, Y., Bakhshandegan Moghaddam, F., Cella, L., Cereda, S., Cremonesi, P.: Exploring the semantic gap for movie recommendations. In: Proceedings of the Eleventh ACM conf. on Recommender Systems, pp. 326–330. ACM (2017)
40. Ellis, D.P.: Classifying music audio with timbral and chroma features. In: ISMIR, vol. 7, pp. 339–340 (2007)
41. Fatemi, N., Mulhem, P.: A conceptual graph approach for video data representation and retrieval. In: International Symposium on Intelligent Data Analysis, pp. 525–536. Springer (1999)
42. Ferrari Dacrema, M., Gasparin, A., Cremonesi, P.: Deriving item features relevance from collaborative domain knowledge. In: Proceedings of KaRS 2018 Workshop on Knowledge-aware and Conversational Recommender Systems. ACM (2018)
43. Gantner, Z., Drumond, L., Freudenthaler, C., Rendle, S., Schmidt-Thieme, L.: Learning attribute-to-feature mappings for cold-start recommendations. In: Data Mining (ICDM), 2010 IEEE 10th International Conference on, pp. 176–185. IEEE (2010)
44. de Gemmis, M., Lops, P., Musto, C., Narducci, F., Semeraro, G.: Semantics-aware content-based recommender systems. In: Recommender Systems Handbook, pp. 119–159. Springer (2015)
45. Haas, A.F., Guibert, M., Foerschner, A., Calhoun, S., George, E., Hatay, M., Dinsdale, E., Sandin, S.A., Smith, J.E., Vermeij, M.J., et al.: Can we measure beauty? computational evaluation of coral reef aesthetics. PeerJ **3**, e1390 (2015)
46. Haghighat, M., Abdel-Mottaleb, M., Alhalabi, W.: Fully automatic face normalization and single sample face recognition in unconstrained environments. Expert Systems with Applications **47**, 23–34 (2016)
47. Harper, F.M., Konstan, J.A.: The movielens datasets: History and context. ACM Transactions on Interactive Intelligent Systems (TiiS) **5**(4), 19 (2016)
48. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. ACM Trans. Inf. Syst. **22**(1), 5–53 (2004). DOI 10.1145/963770.963772
49. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580 (2012)
50. Jalili, V., Matteucci, M., Goecks, J., Deldjoo, Y., Ceri, S.: Next generation indexing for genomic intervals. IEEE Transactions on Knowledge and Data Engineering (2018)
51. Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of ir techniques. ACM Trans. Inf. Syst. **20**(4), 422–446 (2002). DOI 10.1145/582415.582418
52. Jégou, H., Douze, M., Schmid, C., Pérez, P.: Aggregating local descriptors into a compact image representation. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pp. 3304–3311. IEEE (2010)
53. Kaminskas, M., Bridge, D.: Diversity, serendipity, novelty, and coverage: A survey and empirical analysis of beyond-accuracy objectives in recommender systems. ACM

Trans. Interact. Intell. Syst. **7**(1), 2:1–2:42 (2016). DOI 10.1145/2926720. URL http://doi.acm.org/10.1145/2926720
54. Ke, Y., Tang, X., Jing, F.: The design of high-level features for photo quality assessment. In: Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, vol. 1, pp. 419–426. IEEE (2006)
55. Kelly, J.P., Bridge, D.: Enhancing the diversity of conversational collaborative recommendations: a comparison. Artificial Intelligence Review **25**(1), 79–95 (2006). DOI 10.1007/s10462-007-9023-8. URL https://doi.org/10.1007/s10462-007-9023-8
56. Kenny, P.: A small footprint i-vector extractor. In: Odyssey, vol. 2012, pp. 1–6 (2012)
57. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
58. Knees, P., Schedl, M.: Music Similarity and Retrieval: An Introduction to Audio-and Web-based Strategies, vol. 36. Springer (2016)
59. Knijnenburg, B.P., Willemsen, M.C.: Evaluating recommender systems with user experiments. In: Recommender Systems Handbook, pp. 309–352. Springer (2015)
60. Knijnenburg, B.P., Willemsen, M.C., Gantner, Z., Soncu, H., Newell, C.: Explaining the user experience of recommender systems. User Modeling and User-Adapted Interaction **22**(4-5), 441–504 (2012)
61. Koren, Y., Bell, R.: Advances in collaborative filtering. In: Recommender systems handbook, pp. 77–118. Springer (2015)
62. Krages, B.: Photography: the art of composition. Skyhorse Publishing, Inc. (2012)
63. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp. 1097–1105 (2012)
64. Lei, Y., Scheffer, N., Ferrer, L., McLaren, M.: A novel scheme for speaker recognition using a phonetically-aware deep neural network. In: Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on, pp. 1695–1699. IEEE (2014)
65. Lew, M.S., Sebe, N., Djeraba, C., Jain, R.: Content-based multimedia information retrieval: State of the art and challenges. ACM Trans. Multimedia Comput. Commun. Appl. **2**(1), 1–19 (2006). DOI 10.1145/1126004.1126005. URL http://doi.acm.org/10.1145/1126004.1126005
66. Li, C., Chen, T.: Aesthetic visual quality assessment of paintings. IEEE Journal of selected topics in Signal Processing **3**(2), 236–252 (2009)
67. Li, Y., Hu, J., Zhai, C., Chen, Y.: Improving one-class collaborative filtering by incorporating rich user information. In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, pp. 959–968. ACM (2010)
68. Lika, B., Kolomvatsos, K., Hadjiefthymiades, S.: Facing the cold start problem in recommender systems. Expert Systems with Applications **41**(4), 2065–2073 (2014)
69. Liu, J.H., Zhou, T., Zhang, Z.K., Yang, Z., Liu, C., Li, W.M.: Promoting cold-start items in recommender systems. PloS one **9**(12), e113,457 (2014)
70. Liu, L., Chen, R., Wolf, L., Cohen-Or, D.: Optimizing photo composition. In: Computer Graphics Forum, vol. 29, pp. 469–478. Wiley Online Library (2010)
71. Liu, N.N., Meng, X., Liu, C., Yang, Q.: Wisdom of the better few: cold start recommendation via representative based rating elicitation. In: Proceedings of the fifth ACM conference on Recommender systems, pp. 37–44. ACM (2011)
72. Liu, N.N., Yang, Q.: Eigenrank: a ranking-oriented approach to collaborative filtering. In: SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, pp. 83–90. ACM, New York, NY, USA (2008). DOI 10.1145/1390334.1390351. URL http://dx.doi.org/10.1145/1390334.1390351
73. Logan, B.: Mel Frequency Cepstral Coefficients for Music Modeling. In: Proceedings of the International Symposium on Music Information Retrieval (ISMIR). Plymouth, MA, USA (2000)
74. Logan, B., et al.: Mel frequency cepstral coefficients for music modeling. In: ISMIR (2000)
75. Lops, P., De Gemmis, M., Semeraro, G.: Content-based recommender systems: State of the art and trends. In: Recommender systems handbook, pp. 73–105. Springer (2011)
76. Ma, H., King, I., Lyu, M.R.: Learning to recommend with explicit and implicit social relations. ACM Transactions on Intelligent Systems and Technology (TIST) **2**(3), 29 (2011)

77. Matsuda, Y.: Color design. Asakura Shoten **2**(4), 10 (1995)
78. McAuley, J., Targett, C., Shi, Q., Van Den Hengel, A.: Image-based recommendations on styles and substitutes. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 43–52. ACM (2015)
79. McCrae, R.R., John, O.P.: An Introduction to the Five-Factor Model and its Applications. Journal of Personality **60**(2), 175–215 (1992)
80. McFee, B., Barrington, L., Lanckriet, G.: Learning content similarity for music recommendation. IEEE transactions on audio, speech, and language processing **20**(8), 2207–2218 (2012)
81. Mei, T., Yang, B., Hua, X.S., Li, S.: Contextual video recommendation by multimodal relevance and user feedback. ACM Transactions on Information Systems (TOIS) **29**(2), 10 (2011)
82. Mei, T., Yang, B., Hua, X.S., Yang, L., Yang, S.Q., Li, S.: Videoreach: an online video recommendation system. In: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 767–768. ACM (2007)
83. Ning, X., Karypis, G.: Slim: Sparse linear methods for top-n recommender systems. In: 2011 11th IEEE International Conference on Data Mining, pp. 497–506. IEEE (2011)
84. Obrador, P., Schmidt-Hackenberg, L., Oliver, N.: The role of image composition in image aesthetics. In: Image Processing (ICIP), 2010 17th IEEE International Conference on, pp. 3185–3188. IEEE (2010)
85. van den Oord, A., Dieleman, S., Schrauwen, B.: Deep content-based music recommendation. In: C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, K.Q. Weinberger (eds.) Advances in Neural Information Processing Systems 26, pp. 2643–2651. Curran Associates, Inc. (2013). URL `http://papers.nips.cc/paper/5004-deep-content-based-music-recommendation.pdf`
86. Park, S.T., Chu, W.: Pairwise preference regression for cold-start recommendation. In: Proceedings of the third ACM conference on Recommender systems, RecSys '09, pp. 21–28. ACM, New York, NY, USA (2009). DOI 10.1145/1639714.1639720
87. Paudel, B., Christoffel, F., Newell, C., Bernstein, A.: Updatable, accurate, diverse, and scalable recommendations for interactive applications. ACM Transactions on Interactive Intelligent Systems (TiiS) **7**(1), 1 (2017)
88. Pilászy, I., Tikk, D.: 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. ACM (2009)
89. Rasheed, Z., Sheikh, Y., Shah, M.: On the use of computable features for film classification. IEEE Transactions on Circuits and Systems for Video Technology **15**(1), 52–64 (2005)
90. Rendle, S.: Factorization machines with libFM. ACM Transactions on Intelligent Systems and Technology (TIST) **3**(3), 57:1–57:22 (2012)
91. Ribeiro, M.T., Lacerda, A., Veloso, A., Ziviani, N.: Pareto-efficient hybridization for multi-objective recommender systems. In: Proceedings of the Sixth ACM Conference on Recommender Systems, RecSys '12, pp. 19–26. ACM, New York, NY, USA (2012). DOI 10.1145/2365952.2365962. URL `http://doi.acm.org/10.1145/2365952.2365962`
92. Saveski, M., Mantrach, A.: Item cold-start recommendations: learning local collective embeddings. In: Proceedings of the 8th ACM Conference on Recommender systems, pp. 89–96. ACM (2014)
93. Schedl, M., Knees, P., McFee, B., Bogdanov, D., Kaminskas, M.: Recommender Systems Handbook, 2nd edn., chap. Music Recommender Systems. Springer (2015)
94. Schedl, M., Zamani, H., Chen, C., Deldjoo, Y., Elahi, M.: Current challenges and visions in music recommender systems research. IJMIR **7**(2), 95–116 (2018). DOI 10.1007/s13735-018-0154-2. URL `https://doi.org/10.1007/s13735-018-0154-2`
95. Schein, A.I., Popescul, A., Ungar, L.H., Pennock, D.M.: Methods and metrics for cold-start recommendations. In: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 253–260. ACM (2002)
96. Seyerlehner, K., Schedl, M., Knees, P., Sonnleitner, R.: A Refined Block-level Feature Set for Classification, Similarity and Tag Prediction. In: 7th Annual Music Information Retrieval Evaluation eXchange (MIREX 2011). Miami, FL, USA (2011)

97. Seyerlehner, K., Widmer, G., Schedl, M., Knees, P.: Automatic Music Tag Classification based on Block-Level Features. In: Proceedings of the 7th Sound and Music Computing Conference (SMC). Barcelona, Spain (2010)
98. Sharma, M., Zhou, J., Hu, J., Karypis, G.: Feature-based factorized bilinear similarity model for cold-start top-n item recommendation. In: Proceedings of the 2015 SIAM International Conference on Data Mining, pp. 190–198. SIAM (2015)
99. Sheskin, D.J.: Handbook of parametric and nonparametric statistical procedures. crc Press (2003)
100. Singh, A.P., Gordon, G.J.: Relational learning via collective matrix factorization. In: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 650–658. ACM (2008)
101. Smyth, B., McClave, P.: Similarity vs. diversity. In: Proceedings of the 4th International Conference on Case-Based Reasoning: Case-Based Reasoning Research and Development, ICCBR '01, pp. 347–361. SpringerVerlag, London, UK (2001). URL `http://dl.acm.org/citation.cfm?id=646268.758890`
102. Snoek, C.G., Worring, M., Smeulders, A.W.: Early versus late fusion in semantic video analysis. In: Proceedings of the 13th annual ACM international conference on Multimedia, pp. 399–402. ACM (2005)
103. Suh, J.W., Sadjadi, S.O., Liu, G., Hasan, T., Godin, K.W., Hansen, J.H.: Exploring hilbert envelope based acoustic features in i-vector speaker verification using ht-plda. In: Proc. of NIST 2011 Speaker Recognition Evaluation Workshop (2011)
104. Tkalčič, M., Burnik, U., Košir, A.: Using affective parameters in a content-based recommender system for images. User Modeling and User-Adapted Interaction **20**(4), 279–311 (2010)
105. Vall, A., Dorfer, M., Eghbal-zadeh, H., Schedl, M., Burjorjee, K., Widmer, G.: Feature-combination hybrid recommender systems for automated music playlist continuation. User Modeling and User-Adapted Interaction (2019)
106. Vargas, S., Castells, P.: Rank and Relevance in Novelty and Diversity Metrics for Recommender Systems. In: Proceedings of the 5^{th} ACM Conference on Recommender Systems (RecSys). Chicago, IL, USA (2011)
107. Victor, P., Cornelis, C., Teredesai, A.M., De Cock, M.: Whom should i trust?: the impact of key figures on cold start recommendations. In: Proceedings of the 2008 ACM symposium on Applied computing, pp. 2014–2018. ACM (2008)
108. Weimer, M., Karatzoglou, A., Smola, A.: Adaptive collaborative filtering. In: RecSys '08: Proceedings of the 2008 ACM conference on Recommender systems, pp. 275–282. ACM, New York, NY, USA (2008). DOI 10.1145/1454008.1454050
109. Xu, Y., Monrose, F., Frahm, J.M., et al.: Caught red-handed: Toward practical video-based subsequences matching in the presence of real-world transformations. In: Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on, pp. 1397–1406. IEEE (2017)
110. Yang, B., Mei, T., Hua, X.S., Yang, L., Yang, S.Q., Li, M.: Online video recommendation based on multimodal fusion and relevance feedback. In: Proceedings of the 6th ACM international conference on Image and video retrieval, pp. 73–80. ACM (2007)
111. Yuan, J., Shalaby, W., Korayem, M., Lin, D., AlJadda, K., Luo, J.: Solving cold-start problem in large-scale recommendation engines: A deep learning approach. In: Big Data (Big Data), 2016 IEEE International Conference on, pp. 1901–1910. IEEE (2016)
112. Zettl, H.: Sight, sound, motion: Applied media aesthetics. Cengage Learning (2013)
113. Zhang, L., Agarwal, D., Chen, B.C.: Generalizing matrix factorization through flexible regression priors. In: Proceedings of the fifth ACM conference on Recommender systems, pp. 13–20. ACM (2011)
114. Zhang, X., Cheng, J., Qiu, S., Zhu, G., Lu, H.: Dualds: A dual discriminative rating elicitation framework for cold start recommendation. Knowledge-Based Systems **73**, 161–172 (2015)
115. Zhang, Z.K., Liu, C., Zhang, Y.C., Zhou, T.: Solving the cold-start problem in recommender systems with social tags. EPL (Europhysics Letters) **92**(2), 28,002 (2010)
116. Zhou, K., Yang, S.H., Zha, H.: Functional matrix factorizations for cold-start recommendation. In: Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval, pp. 315–324. ACM (2011)

117. Zhou, T., Kuscsik, Z., Liu, J.G., Medo, M., Wakeling, J.R., Zhang, Y.C.: Solving the apparent diversity-accuracy dilemma of recommender systems. Proceedings of the National Academy of Sciences **107**(10), 4511–4515 (2010)
118. Ziegler, C.N., McNee, S.M., Konstan, J.A., Lausen, G.: Improving recommendation lists through topic diversification. In: Proceedings of the 14th International Conference on the World Wide Web, pp. 22–32. ACM (2005)

Authors' Biographies

Yashar Deldjoo is a research assistant at the Information Retrieval Laboratory of University of Milano Bicocca, Italy. He obtained his Masters degree in Electrical Engineering from Chalmers University of Technology, Sweden in 2010. He earned his Ph.D. with Distinction in Computer Science from Politecnico di Milano, Italy in 2018. His main research interests include recommender systems, user modeling, multimedia recommendation and multimedia computing. Despite being a very young researcher, Yashar has (co-)authored quite many refereed articles at internationally recognized conferences and peer-reviewed journals (among others, published in ACM RecSys, MMSys, CHI, IEEE Transactions on Knowledge and Data Engineering, and User Modeling and User-Adapted Interaction, Journal of Data Semantics and Springer journal of multimedia information retrieval). He is also holder of a US Patent. Yashar has served as a PC member or reviewer for top-tier conferences including SIGIR, CIKM, AAAI, ACM MM, RecSys, ECML-PKDD, ECIR, MMSys, IDC, IPMU as well as top-tier journals, including Expert Systems with Applications and IEEE Access.

Maurizio Ferrari Dacrema is a Ph.D. candidate at the Department of Electronics, Information and Bioengineering (DEIB) at Politecnico di Milano, Italy, where he graduated in Computer Science and Engineering in 2016. His research is now focused on building hybrid recommender systems via feature weighting, as well as on offline model evaluation. He has been a member of a regular panel for Pilot 3 of the Horizon2020 TeSLa Project.

Mihai Gabriel Constantin graduated from the Faculty of Automatic Control and Computers, University Politehnica of Bucharest (UPB) and is currently a PhD candidate with the Faculty of Electronics, Telecommunications and Information Technology and a researcher with the Multimedia Lab, CAMPUS Research Center, UPB, Romania. His research focused on the study of methods for analyzing the visual impact of multimedia data. He has authored over 11 scientific publications and was involved in several Romanian/EU funded research projects. He was a member of the organizing team for several conferences (e.g., IEEE/ACM CBMI 2016, ACM ICMR 2017) and benchmarking tasks (e.g., 2018 MediaEval Recommending Movies Using Content task).

Hamid Eghbal-zadeh is a research assistant at the Johannes Kepler University (JKU) Linz, Austria. He received his Masters degree in Computer Engineering from the Shiraz State University, Iran in 2012. From 2012 to 2014, he worked as a researcher in the area of speech processing at Ozyegin University in Istanbul, Turkey. In 2014, he joined the Institute of Computational Perception at the Johannes Kepler University of Linz. During the following years, he worked as a researcher on representation learning from signals and sequences.

He is currently pursuing his PhD and his focus of research is on learning from data with incomplete label information.

Stefano Cereda is a PhD student at Politecnico di Milano, Italy, in the department of Electronics, Information and Bioengineering, where he received his Masters degree in Computer Science and Engineering in 2017. After his Masters degree he worked as a researcher in the topic of recommender system. His research is now focused on automatic hyperparameter optimization for performance tuning.

Markus Schedl is associate professor at the Johannes Kepler University (JKU) Linz, Austria, affiliated with the Institute of Computational Perception. He graduated in Computer Science from the Vienna University of Technology and earned his Ph.D. in Computer Science from the JKU. His main research interests include web and social media mining, data analytics, information retrieval, recommender systems, multimedia, and music information research. Markus (co-)authored more than 200 refereed conference papers and journal articles (among others, published in ACM Multimedia, RecSys, ICMR, SIGIR, ECIR, ISMIR, WEB/WWW, IEEE Visualization; Journal of Machine Learning Research, ACM Transactions on Information Systems, IEEE Transactions on Affective Computing, IEEE Multimedia, User Modeling and User-Adapted Interaction, PLOS ONE). Furthermore, he is associate editor of the Springer International Journal of Multimedia Information Retrieval and the Transactions of the International Society for Music Information Retrieval. He serves on the program committee of top-tier conferences, including WWW, RecSys, ICMR, and ACM Multimedia, and as reviewer for top-tier journals, including ACM Computing Surveys, Elsevier Pattern Recognition Letters, IEEE Transactions on Multimedia, Journal of the Association for Information Science and Technology, and ACM Transactions on Intelligent Systems and Technology.

Bogdan Ionescu coordinates the CAMPUS Research Center at University Politehnica of Bucharest (UPB). He holds a double PhD degree in image/video processing from UPB and University of Savoie, France. He is currently a tenured Professor with ETTI-UPB. His main research interests cover: multimedia/video/image processing and analysis, multimedia content-based retrieval and machine learning for multimedia. He has authored over 160 scientific publications. He serves/served as guest co-editor for Image and Vision Computing, Multimedia Tools and Applications, and International Journal of Computer Vision; conference committee chair for various conferences, e.g., ACM Multimedia 2013 (proceedings co-chair), ACM Multimedia 2019 (publicity chair), IEEE/ACM CBMI 2016 (general chair), ACM ICMR 2017 (general chair), CLEF 2021 (general chair); lead organizer/co-organizer for several benchmark campaigns: MediaEval Retrieving Diverse Social Images, Violent Scenes Detection, Affective Impact of Movies Task, Predicting Media Interestingness, and Predicting Media Memorability, ImageCLEF 2017-2019, ChaLearn ICPR Multimedia Information Processing for Personality & Social Networks Analysis Challenge. He contributed to over 21 Romanian/EU funded research and strategic programmes, as principal investigator or as part of the research team.

Paolo Cremonesi is professor of Recommender Systems at the Computer Science Department of Politecnico di Milano. Paolo is also the co-founder of

Moviri and ContentWise, the first and most successful start-ups from Politecnico di Milano. Paolo holds an MSc in Aerospace Engineering and a PhD in Computer Science. His research interests include high-performance computing and recommender systems. He has published almost 200 papers in refereed journals, conferences and book chapters, and holds 5 patents.

Table 3: Performance of various features: i-vector (Audio), BLF (Audio), Deep (Visual), and AVF (Visual), editorial-metadata, in terms of *accuracy metrics* NDCG and MAP. For fusion, we report the results for the CCA fusion variation (either ccat or sum) that lead to the best performance (cf. Section 3.2). The features (or feature combination) which *outperform* genre *significantly* are shown in bold ($p < 0.05$). Abbreviations: A - Audio, V - Visual, EM - Editorial Metadata.

	Feature name	CCA Fusion	Accuracy Metrics NDCG	MAP
Base	cast (EM)	-	0.0088	0.0041
	genre (EM)	-	0.0083	0.0042
Unimodal	i-vec (A: SoA)	-	0.0095	0.0045
	BLF (A: traditional)	-	0.0111	0.0055
	Deep (V: SoA)	-	0.0127	0.0060
	AVF (A: traditional)	-	0.0108	0.0047
Multimodal (pure CBF)	i-vec + Genre (A + EM)	sum	0.0101	0.0046
	i-vec + Deep (A + V)	sum	0.0089	0.0042
	i-vec + AVF (A + V)	ccat	0.0077	0.0038
	i-vec + BLF (A)	sum	0.0094	0.0046
	AVF + Genre (V + EM)	sum	0.0086	0.0039
	AVF + BLF (V + A)	sum	0.0091	0.0052
	AVF + Deep (V)	sum	0.0086	0.0041
	Deep + Genre (V + EM)	sum	0.0078	0.0038
	Deep + BLF (V + A)	sum	0.0102	0.0053
	BLF + Genre (A + EM)	ccat	0.0090	0.0046
Multimodal (CFeCBF)	**i-vec+Genre (A + EM)**	**sum**	**0.0186**	0.0078
	i-vec + Deep (A + V)	sum	0.0176	0.0083
	i-vec + AVF (A + V)	sum	0.0121	0.0062
	i-vec + BLF (A)	ccat	0.0177	0.0083
	AVF+Genre (V + EM)	**sum**	**0.0192**	**0.0097**
	AVF + BLF (V + A)	sum	0.0102	0.0048
	AVF + Deep (A + V)	**sum**	**0.0191**	**0.0097**
	Deep + Genre (V + EM)	sum	0.0117	0.0059
	Deep + BLF (A)	sum	0.0111	0.0052
	BLF + Genre (A + EM)	ccat	0.0120	0.0059

Table 4: Performance of various features in terms of *beyond-accuracy metrics* for *list diversity*. For fusion, we report the results for the CCA fusion variation (either ccat or sum) that lead to the the best performance (cf. Section 3.2). Results in *bold* show the features (or feature combinations) that *outperform* genre *significantly* ($p < 0.05$) along the respective metric. Abbreviations: A - Audio, V - Visual, EM - Editorial Metadata.

	Feature name	CCA Fusion	List Diversity IntraL	InterL
Base	cast (EM)	-	**0.8990**	0.8794
	genre (EM)	-	0.8886	0.9035
Unimodal	i-vec (A: SoA)	-	**0.8994**	**0.9322**
	BLF (A: traditional)	-	**0.8994**	**0.9522**
	Deep (V: SoA)	-	**0.8992**	0.8641
	AVF (A: traditional)	-	**0.8994**	**0.9528**
Multimodal (pure CBF)	i-vec + Genre (A + EM)	sum	**0.8965**	**0.9577**
	i-vec + Deep (A + V)	sum	**0.8994**	**0.9602**
	i-vec + AVF (A + V)	ccat	**0.8994**	0.7682
	i-vec + BLF (A)	ccat	**0.8995**	0.8772
	AVF + Genre (V + EM)	(sum, ccat)	**0.8927**	**0.9536**
	AVF + BLF (V + A)	(sum, ccat)	**0.8995**	0.8724
	AVF + Deep (V)	ccat	**0.8994**	0.6890
	Deep + Genre (V + EM)	ccat	**0.8964**	**0.9633**
	Deep + BLF (V + A)	ccat	**0.8995**	**0.9548**
	BLF + Genre (A + EM)	ccat	**0.8969**	**0.9616**
Multimodal (CFeCBF)	i-vec + Genre (A + EM)	sum	**0.8981**	**0.9304**
	i-vec + Deep (A + V)	sum	**0.8995**	**0.9535**
	i-vec + AVF (A + V)	(ccat, ssum)	**0.8995**	**0.9584**
	i-vec + BLF (A)	sum	**0.8995**	**0.9533**
	AVF + Genre (V + EM)	(sum, ccat)	**0.8992**	0.7532
	AVF + BLF (V + A)	sum	**0.8995**	**0.9373**
	AVF + Deep (V)	ccat	**0.8995**	**0.9549**
	Deep + Genre (V + EM)	(ccat, sum)	**0.8983**	**0.9361**
	Deep + BLF (V + A)	ccat	**0.8995**	**0.9599**
	BLF + Genre (A + EM)	ccat	**0.8986**	**0.9396**

Table 5: Performance of various features in terms of *beyond-accuracy metrics* for *aggregate diversity*. Results in bold show the features (or feature combinations) that *outperform* genre *significantly* ($p < 0.05$). For each feature combination, we only report the results for the CCA method that has the best performance (either ccat or sum). Abbreviations: A - Audio, V - Visual, EM - Editorial Metadata, Entropy - Shannon Entropy, HHI - Herfindahl, Item Cov - Item Coverage.

	Feature name	CCA Fusion	Distributional Diversity Gini	Entropy	HHI	Item Cov
Base	cast (EM)	-	0.7652	7.2672	0.9879	0.5348
	genre (EM)	-	0.7424	7.4525	0.9903	0.5435
Unimodal	i-vec (A: SoA)	-	**0.7055**	**8.2934**	**0.9932**	**0.9276**
	BLF (A: traditional)	-	**0.6614**	**8.5983**	**0.9952**	**0.9412**
	Deep (V: SoA)	-	0.7992	7.2659	0.9864	**0.6615**
	AVF (A: traditional)	-	**0.6583**	**8.6165**	**0.9952**	**0.9336**
Multimodal (pure CBF)	i-vec + Genre (A + EM)	sum	**0.6510**	**8.6752**	**0.9957**	**0.9431**
	i-vec + Deep (A + V)	sum	**0.6283**	**8.7951**	**0.9960**	**0.9960**
	i-vec + AVF (A + V)	ccat	0.7794	6.1634	0.9768	0.2569
	i-vec + BLF (A)	ccat	0.8022	7.2558	0.9877	**0.6412**
	AVF + Genre (V + EM)	ccat	**0.6754**	**8.4935**	**0.9953**	**0.8811**
	AVF + BLF (V + A)	(ccat, *: sum)	**0.6595***	6.7429	0.9872	0.3014
	AVF + Deep (V)	(ccat)	0.8037	5.7369	0.9689	0.2147
	Deep + Genre (V + EM)	(ccat, *:sum)	**0.6361**	**8.7644**	**0.9963**	**0.9388***
	Deep + BLF (V + A)	(ccat)	**0.6402**	**8.7184**	**0.9954**	**0.9655**
	BLF + Genre (A + EM)	(ccat)	**0.6381**	**8.7520**	**0.9961**	**0.9459**
Multimodal (CFeCBF)	i-vec + Genre (A + EM)	sum	**0.7232**	**8.0490**	**0.9930**	**0.7769**
	i-vec + Deep (A + V)	sum	**0.6719**	**8.5653**	**0.9953**	**0.9275***
	i-vec + AVF (A + V)	sum	**0.6342**	**8.6499**	**0.9958**	**0.8736**
	i-vec + BLF (A)	sum	**0.6667**	**8.5831**	**0.9953**	**0.9299**
	AVF + Genre (V + EM)	ccat	0.7742	6.0510	0.9753	0.2345
	AVF + BLF (V + A)	(sum, *:ccat)	**0.7150**	**8.0789**	**0.9937**	**0.7664***
	AVF + Deep (V)	ccat	**0.6504**	**8.5740**	**0.9955**	**0.8691**
	Deep + Genre (V + EM)	sum	**0.7170**	**8.1647**	**0.9936**	**0.8258**
	Deep + BLF (V + A)	ccat	**0.6439**	**8.7395**	**0.9960**	**0.9595**
	BLF + Genre (A + EM)	ccat	**0.7007**	**8.2965**	**0.9939**	**0.8619**

Table 6: Results for the cold to warm transition scenario for accuracy metrics and Item Coverage. In evaluation scenario *Cold* the test items are cold. In *Warm 0.5%* the 0.5% of existing interactions have been added to the cold items, while in *Warm 2.0%* its the 2.0%.

	Feature name	Evaluation scenario								
		Cold			Warm 0.5 %			Warm 2.0 %		
		NDCG	MAP	Item Cov	NDCG	MAP	Item Cov	NDCG	MAP	Item Cov
Base	Cast (EM)	0.0087	0.0040	0.5327	0.0087	0.0040	0.5327	0.0087	0.0040	0.5333
	Genre (EM)	0.0114	0.0051	0.5987	0.0114	0.0051	0.5987	0.0114	0.0051	0.5991
Multimodal (CFeCBF)	Cast (EM)	0.0095	0.0046	0.7130	0.0095	0.0046	0.7134	0.0095	0.0045	0.7126
	Genre (EM)	0.0082	0.0039	0.4867	0.0088	0.0042	0.5064	0.0071	0.0033	0.4215
	i-vec + Genre (A + EM)	0.0094	0.0040	0.7754	0.0080	0.0039	0.8128	0.0113	0.0054	0.8407
	i-vec + Deep (A + V)	0.0082	0.0035	0.9263	0.0105	0.0048	0.9237	0.0118	0.0052	0.9183
	i-vec + AVF (A + V)	0.0086	0.0045	0.7665	0.0109	0.0051	0.8492	0.0088	0.0041	0.8285
	i-vec + BLF (A)	0.0115	0.0055	0.7227	0.0110	0.0053	0.7213	0.0114	0.0058	0.7598
	AVF + Genre (V + EM)	0.0139	0.0067	0.0522	0.0136	0.0062	0.1563	0.0216	0.0109	0.2999
	AVF + BLF (V + A)	0.0075	0.0036	0.6423	0.0081	0.0040	0.5446	0.0108	0.0053	0.6187
	AVF + Deep (A + V)	0.0093	0.0042	0.7905	0.0098	0.0045	0.8575	0.0105	0.0049	0.8488
	Deep + Genre (V + EM)	0.0087	0.0042	0.7447	0.0108	0.0052	0.6887	0.0097	0.0044	0.7029
	Deep + BLF (A)	0.0094	0.0042	0.8329	0.0095	0.0043	0.8742	0.0086	0.0041	0.9346
	BLF + Genre (A + EM)	0.0072	0.0034	0.7912	0.0078	0.0038	0.8071	0.0093	0.0041	0.7959
CF	RP3beta	0.0000	0.0000	0.0000	0.0900	0.0494	0.0613	0.1185	0.0884	0.2030

Table 7: Results for the cold to warm transition scenario for accuracy metrics and Item Coverage. In evaluation scenario *Cold* the test items are cold, in *Warm 1* each test item has exactly 1 interaction in the train set, in *Warm 5* each test items has 5 interactions in the train set.

	Feature name	Evaluation scenario								
		Cold			Warm 1			Warm 5		
		NDCG	MAP	Item Cov	NDCG	MAP	Item Cov	NDCG	MAP	Item Cov
Base	Cast (EM)	0.0059	0.0031	0.5256	0.0059	0.0031	0.5329	0.0047	0.0025	0.5416
	Genre (EM)	0.0058	0.0031	0.5925	0.0058	0.0031	0.5963	0.0053	0.0027	0.5999
CFeCBF	Cast (EM)	0.0060	0.0031	0.7038	0.0061	0.0031	0.7166	0.0047	0.0024	0.7308
	Genre (EM)	0.0056	0.0029	0.3641	0.0078	0.0041	0.5230	0.0053	0.0027	0.4293
	i-vec + Genre (A + EM)	0.0053	0.0027	0.7973	0.0060	0.0032	0.7790	0.0046	0.0023	0.7976
	i-vec + Deep (A + V)	0.0064	0.0033	0.9154	0.0054	0.0027	0.9168	0.0048	0.0026	0.9408
	i-vec + AVF (A + V)	0.0054	0.0026	0.6817	0.0063	0.0035	0.8447	0.0044	0.0021	0.8358
	i-vec + BLF (A)	0.0055	0.0031	0.7157	0.0061	0.0030	0.7416	0.0036	0.0017	0.7552
	AVF + Genre (V + EM)	0.0050	0.0025	0.2887	0.0070	0.0033	0.1348	0.0065	0.0036	0.2179
	AVF + BLF (V + A)	0.0057	0.0029	0.4980	0.0057	0.0031	0.5549	0.0048	0.0028	0.5670
	AVF + Deep (A + V)	0.0062	0.0032	0.8689	0.0054	0.0028	0.8003	0.0047	0.0023	0.8527
	Deep + Genre (V + EM)	0.0055	0.0032	0.7747	0.0058	0.0032	0.6469	0.0047	0.0025	0.7211
	Deep + BLF (A)	0.0053	0.0027	0.9397	0.0046	0.0022	0.9318	0.0048	0.0025	0.9416
	BLF + Genre (A + EM)	0.0051	0.0027	0.8037	0.0052	0.0027	0.7722	0.0044	0.0021	0.8222
CF	RP3beta	0.0000	0.0000	0.0000	0.0327	0.0180	0.8783	0.0507	0.0269	0.9012

Table 8: The list of questions [37,60] used to measure the perceived quality of recommendations. Note that answers/scores given to questions marked with a + contribute positively to the final score, whereas scores to questions marked with a **-** are subtracted.

Factor / Question (W. l. = Which list, W. r. = Which recommender)
Percieved Accuracy
W. l. has more movies that you find appealing? **(Q17 +)**
W. l. has more movies that might be among the best movies you see in the next year? **(Q19 +)**
W. l. has more obviously bad movie recommendations for you? **(Q6 -)**
W. r. does a better job of putting better movies on the left? **(Q9 +)**
Diversity
W. l. has more movies that are similar to each other? **(Q22 -)**
W. l. has a more varied selection of movies? **(Q7 +)**
W. l. has movies that match a wider variety of moods? **(Q13 +)**
W. l. would suit a broader set of tastes? **(Q2 +)**
Understands Me
W. r. better understands your taste in movies? **(Q12 +)**
W. r. would you trust more to provide you with recommendations? **(Q18 +)**
W. r. seems more personalized to your movie taste? **(Q14 +)**
W. r. more represents mainstream tastes instead of your own? **(Q3 -)**
Satisfaction
W. r. would better help you find movies to watch? **(Q8 +)**
W. r. would you be more likely to recommend to your friends? **(Q16 +)**
W. l. of recommendations do you find more valuable? **(Q11 +)**
W. r. would you rather have as an app on your mobile phone? **(Q20 +)**
W. r. would better help to pick satisfactory movies? **(Q1 +)**
Novelty
W. l. has more movies you do not expect? **(Q21 +)**
W. l. has more movies that are familiar to you? **(Q4 -)**
W. l. has more pleasantly surprising movies? **(Q5 +)**
W. l. has more movies you would not have thought to consider to watch? **(Q10 +)**
W. l. provides most new suggestions? **(Q15 +)**

Table 9: Results of the user study with respect to the five tested perceived quality criteria in a real movie recommender system.

feature name	feature type	Relevance	Diversity	Understands me	Satisfaction	Novelty
tag	metadata	0.2632	0.1625	0.3133	0.2514	0.0577
genre	metadata	0.2526	0.2857	0.2410	0.2404	0.1538
i-vector	audio	0.1263	0.1875	0.0361	0.1093	0.2115
BLF	audio	0.0316	0.1250	0.0120	0.0601	0.0769
deep	visual	0.2421	0.1750	0.3253	0.2459	0.1923
AVF	visual	0.0842	0.0625	0.0723	0.0929	0.3077