International Journal of Multimedia Information Retrieval
https://doi.org/10.1007/s13735-018-0155-1

REGULAR PAPER

Using visual features based on MPEG-7 and deep learning for movie recommendation

Yashar Deldjoo[1] · Mehdi Elahi[2] · Massimo Quadrana[1] · Paolo Cremonesi[1]

Received: 19 January 2018 / Revised: 24 April 2018 / Accepted: 2 June 2018

Abstract
Item features play an important role in movie recommender systems, where recommendations can be generated by using explicit or implicit preferences of users on traditional features (attributes) such as tag, genre, and cast. Typically, movie features are human-generated, either editorially (e.g., genre and cast) or by leveraging the wisdom of the crowd (e.g., tag), and as such, they are prone to noise and are expensive to collect. Moreover, these features are often rare or absent for new items, making it difficult or even impossible to provide good quality recommendations. In this paper, we show that users' preferences on movies can be well or even better described in terms of the *mise-en-scène* features, i.e., the visual aspects of a movie that characterize design, aesthetics and style (e.g., colors, textures). We use both *MPEG-7* visual descriptors and *Deep Learning* hidden layers as examples of mise-en-scène features that can visually describe movies. These features can be computed automatically from any video file, offering the flexibility in handling new items, avoiding the need for costly and error-prone human-based tagging, and providing good scalability. We have conducted a set of experiments on a large catalog of 4K movies. Results show that recommendations based on mise-en-scène features consistently outperform traditional metadata attributes (e.g., genre and tag).

Keywords Multimedia recommendation · Video analysis · Deep learning · Cold start · Visual features

1 Introduction

Multimedia recommender systems typically base their recommendations on human-generated content attributes which are either crowd-sourced (e.g., tags, reviews, and comments) or editorial-generated (e.g., genre, director, and cast). The typical approach is to recommend items sharing properties with the other items the user liked in the past.

In the movie domain, information about movies can be exploited by either use content-based filtering (CBF) or to boost collaborative filtering (CF) with rich *side information* [49]. A necessary prerequisite for both CBF and CF with side information is the availability of a rich set of descriptive *attributes* about movies.

An open problem with multimedia recommender systems is to enable generation of recommendations when users' ratings or items' "traditional" (human-generated) attributes are missing or incomplete. This problem is known the *New Item* problem [1,24–26,38,45,46,48] and it often happens in video-on-demand application scenarios, when a new media content is added to the catalog of available items with no metadata. For instance, the users of YouTube upload 500 h of movies per minute [52] and some of uploaded videos may not contain any metadata and hence cause new item problem to occur.

In order to better describe our proposed solution, for this problem, we provide a brief explanation of the features that can represent any form of multimedia content (i.e., movies, videos, and films). Such features can be classified into three hierarchical levels [55]:

1. At the highest level, there exist *semantic* features that deal with concepts or events in a movie. An example

✉ Mehdi Elahi
meelahi@unibz.it

Yashar Deldjoo
deldjooy@acm.org

Massimo Quadrana
massimo.quadrana@polimi.it

Paolo Cremonesi
paolo.cremonesi@polimi.it

[1] Politecnico di Milano, Via Ponzio 34/5, 20133 Milan, Italy

[2] Free University of Bozen - Bolzano, Piazza Domenicani 3, 39100 Bolzano, Italy

Published online: 14 June 2018

of semantic feature is the plot of the movie *The Good, the Bad and the Ugly*, which revolves around three gunslingers competing to find a buried cache of gold during the American Civil War;
2. At the intermediate level, there exist *syntactic* features that deal with the objects in a movie and their interactions. As an example, in the same noted movie, there are Clint Eastwood, Lee Van Cleef, Eli Wallach, plus several horses and guns;
3. At the lowest level, there exist *stylistic* features, related to the *mise-en-scène* form of a movie, i.e., the design aspects that characterizes aesthetic and style of a movie (e.g., colors or textures); as an example, in the same movie noted, predominant colors are yellow and brown, and camera shots use extreme close-up on actors' eyes.

The same plot (semantic level) can be acted by different actors (syntactic level) and directed in different ways (stylistic level). In general, there is no direct link between the high-level concepts and the low-level features. Each combination of features convey different communication effects and stimulate different feelings in the viewers.

Recommender systems in the movie domain mainly focus on the high-level or the intermediate-level features—usually provided by a group of domain experts or by a large community of users—such as movie genres (semantic features, high level), actors (syntactic features, intermediate level) or tags (semantic and syntactic features, high and intermediate levels) [27,34,51]. We refer to these features as *attributes*.

Movie genre is a form of these attributes normally assigned by movie experts, and tags are another form assigned by communities of users [53]. Human-generated attributes present a number of disadvantages:

1. These attributes are prone to user biases and errors, therefore not fully reflecting the characteristics of a movie;
2. New items might lack these attributes as well as user ratings;
3. Unstructured attributes such as tags require complex natural language processing (NLP) in order to account for stemming, stop words removal, synonyms detection and other semantic analysis tasks;
4. Not all attributes of an item have the same importance related to the task at hand; for instance, a background actor does not have the same importance as a guest star in defining the characteristics of a movie.

In contrast to human-generated attributes, the content of movie streams is itself a rich source of information about low-level stylistic features that can be used to provide movie recommendations. Low-level visual features have been shown to be very representative of the users feelings, according to the theory of *Applied Media Aesthetics* [58]. By analyzing a movie stream content and extracting a set of low-level features, a recommender system can make personalized recommendations, tailored to a user's taste. This is particularly beneficial in the new item scenario, i.e., when movies without ratings and without user-generated attributes are added to the catalog.

Moreover, while low-level visual features can be extracted from full-length movies, they can also be extracted from shorter version of the movies (i.e., trailers) in order to have a scalable recommender system [17].

In previous works, we have shown that mise-en-scène visual features extracted from trailers can be used to accurately make prediction of the genre of movies [15,17–19,23]. In this paper, we extend these papers and show how to use low-level visual features extracted automatically from movie files as input to a pure CBF and a hybrid CF+CBF (based on *cSLIM* [41]) algorithm. We have extracted the low-level visual features by using two different approaches:

– MPEG-7 visual descriptors [36]
– Pre-trained deep-learning neural networks (DNN) [50]

Based on the discussion above, we articulate the following research hypothesis: "*a recommender system using low-level visual features (mise-en-scène) provides better accuracy compared to the same recommender system using traditional content attributes (genre and tag)*".

We articulate the research hypothesis along the following research questions:

RQ1: Do visual low-level features extracted from any of MPEG-7 descriptors or pre-trained deep-learning networks provide better top-N recommendations than genre and tag attributes?
RQ2: Do visual low-level features extracted from MPEG-7 descriptor in conjunction with pre-trained deep-learning networks provide better top-N recommendations than genre and tag attributes?

We have performed an exhaustive evaluation by comparing low-level visual features with respect to the traditional attributes (i.e., genre and tag). For each set of features, we have used two algorithms, i.e., (a) CBF using the similarity of the items, and (b) hybrid CBF+CF algorithm that includes item features as side information, where item similarity is learned with a Sparse LInear Method (cSLIM) [41].

We have used visual features and traditional content attributes either individually or in combination, in order to obtain a clear picture of the real ability of visual features in learning the preferences of users and effectively generating relevant recommendations.

We have computed different evaluation metrics, i.e., precision, recall, mean average precision (MAP) and F1, over

a dataset of more than 8M ratings provided by 242K users to about 4K movies. In our experiments, recommendations based on mise-en-scène visual features consistently provide the best performance.

Overall, this work provides a number of contributions to the RSs field in the movie domain:

- we propose a novel RS that automatically analyzes the content of the movies and extracts visual features in order to generate personalize recommendations for users;
- we evaluate recommendations by using a dataset of 4K movies and compare the results with the state-of-the-art recommendation algorithms, i.e., CBF and hybrid CF+CBF;
- we have extracted mise-en-scène visual features adopting two different approaches (i.e., DNN and MPEG-7) and fed them to the recommendation algorithm, either individually or in combination, in order to better study the power of these types of features;
- for the fusion, we employ a data fusion method called Canonical Correlation Analysis (CCA) [17] that has shown promising performance for fusing different feature set including the ones based on visual and textual features.
- the dataset, together with the user ratings and the visual features extracted from the videos (both MPEG-7 and DNN features), is available for download.[1]

The rest of the paper is organized as follows. Section 2 reviews the relevant state of the art, related to content-based recommender systems and video recommender systems. Section 3 introduces some theoretical background on media aesthetics that helps us to motivate our approach and interpret the results of our study. It describes the possible relation between the visual features adopted in our work and the aesthetic variables that are well known for artists in the domain of movie making. In Sect. 4 we describe our method for extracting and representing mise-en-scène visual features of the movies and provide the details of our recommendation algorithms. Section 5 introduces the evaluation method and presents the results of the study and Sect. 6 discusses them. Section 7 draws the conclusions and identifies open issues and directions for future work.

2 Related work

2.1 Multimedia recommender systems

Multimedia recommender systems typically exploit *high-level* or *intermediate-level* features in order to generate movie recommendation [8,39,40]. This type of features express semantic and syntactic properties of media content that are obtained from structured sources of metadata such as databases, lexicons and ontologies, or from less structured data such as reviews, news articles, item descriptions and social tags.

In contrast, in this paper, we propose exploiting *low-level* features to provide recommendations. Such features express stylistic properties of the media content and are extracted directly from the multimedia content files [17].

While this approach has been already investigated in the music recommendation domain [4,47], it has received marginal attention for movie recommendations. The very few approaches only consider low-level features to improve the quality of recommendations based on other type of features. The work in [57] proposes a video recommender system, called *VideoReach*, which incorporates a combination of high-level and low-level video features (such as textual, visual and auditory) in order to improve the click-through-rate metric. The work in [61] proposes a multi-task learning algorithm to integrate multiple ranking lists, generated by using different sources of data, including visual content.

While low-level features have been marginally explored in the community of recommender systems, they have been studied in other fields such as computer vision and content-based video retrieval. The works in [6,33] discuss a large body of low-level features (visual, auditory or textual) that can be considered for video content analysis. The work in [43] proposes a practical movie genre classification scheme based on computable visual cues. Rasheed and Shah [42] discusses a similar approach by considering also the audio features. Finally, the work in [62] proposes a framework for automatic classification of videos using visual features, based on the intermediate level of scene representation.

We note that, while the scenario of using low-level features, as an additional side information, to hybridize the existing recommender systems is an interesting approach, however, this paper addresses a different scenario, i.e., when the only available information is the low-level visual features and the recommender system has to use it effectively for recommendation generation. Indeed, this is a severe case of the new item cold start problem [45], where traditional recommender systems fail in properly doing their job and novel techniques are required to cope with the severe problem [5,22,35,48,60]. It is worthwhile to note that although the present work has a focus on exploiting computer vision techniques on item description of products (i.e., the *item-centric* aspect), these techniques are also exploited in studying users' (interaction) behaviors e.g., through studying their eye, gaze, head movement (and other forms of interaction) while navigating with the system (i.e., the *user-centric* aspect) [2,13,20,56].

[1] recsys.deib.polimi.it.

Fig. 1 Examples of visual effects (e.g., color and light tuning) adopted by movie makers in order to evoke certain types of emotional effects in the audience: **a** the Revenant—2015, **b** the Wolf of Wall Street—2013, **c** Vanilla Sky—2001, **d** Mission: Impossible III—2006, **e** Alice in Wonderland—2010, and **f** Black Mass—2015

2.2 Aesthetic view

The relation of mise-en-scène elements with the reactions they are able to evoke in viewers, is one of the main concerns of applied media aesthetic [58]. Examples of mise-en-scène elements that are addressed in the literature on movie design are *lighting* and *color* [21].

Lighting is the deliberate manipulation of light for a certain communication purpose, and it is used to create viewers' perception of the environment, and establish an aesthetic context for their experiences. The two main lighting alternatives are usually addressed to as *chiaroscuro* and *flat lighting* [59]. Figure 1a, b exemplifies these two alternatives.

Colors can strongly affect our perceptions and emotions in unsuspected ways. For instance, red light gives the feeling of warmth, but also the feeling that time moves slowly, while blue light gives the feeling of cold, but also that time moves faster. The expressive quality of colors strongly depends on the lighting, since colors are a property of light [59]. Figure 1 presents examples of using colors in movies to evoke certain emotions.

Interestingly, most mise-en-scène elements can be computed from the video data stream as statistical values [7,43]. We call these computable aspects as *visual low-level features* [16].

3 Artistic motivation

In this section, we describe the artistic background to the idea of stylistic visual features for movie recommendation. We do this by describing the stylistic visual features from an artistic point of view and explaining the relation between these visual features and the corresponding aesthetic variables in movie-making domain.

The study on aesthetic elements and how their combination contributes to establish the meaning conveyed by an artistic work is the subject of different disciplines such as semiotics, and traditional aesthetic studies. The shared notion is that humans respond to certain stimuli in ways that are predictable, up to a given extent. A consequence of the above notion is that similar stimuli are expected to provoke similar reactions, and this as a result, may allow to group similar works of art together by the reaction they are expected to provoke.

Among these disciplines, applied media aesthetic [58], particularly, is concerned with the relation among media elements, such as light, camera movements, and colors, with the perceptual reactions they are able to evoke in consumers of media communication, mainly videos and films. Such media elements, that together build the visual images composing the media, are investigated following a rather formalistic approach that suits the purposes of this paper. By an analysis of cameras, lenses, lighting, etc., as production tools and their aesthetic characteristics and uses, applied media aesthetic tries to identify patterns in how such elements operate to produce the desired effect in communicating emotions and meanings.

The image elements that are usually addressed as fundamental in the literature e.g., in [21], even if with slight differences due to the specific context, are lights and shadows, colors, space representation, motion. It has been proved

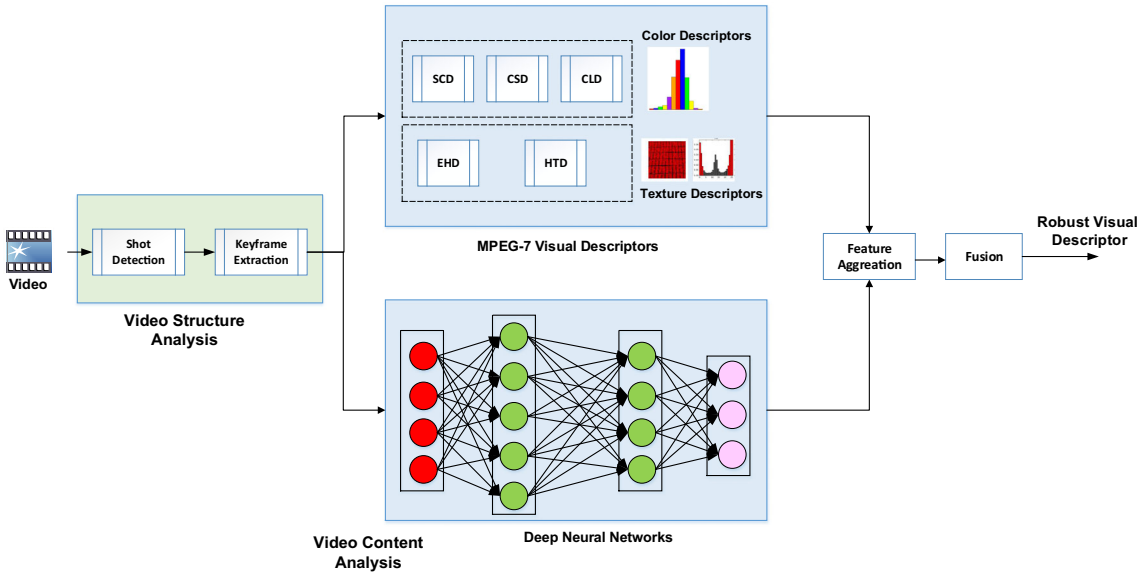

Fig. 2 Flowchart of the methodology designed and developed to extract the visual features, based on MPEG-7 descriptors and pre-trained deep-learning networks, from the movies

e.g., in [7,43], that some aspects concerning these elements can be computed from the video data stream as statistical values. We call these computable aspects as *features*.

The terms mise-en-scène and *aesthetic visual* are used interchangeably throughout the article and as authors in [37, 59] suggest, color, light and texture MPEG-7 descriptors can be used as elements of the applied media aesthetic (referred as mise-en-scène in the movie domain).

4 Methodology

The methodology adopted to provide recommendations based on visual features is composed of five main steps:

1. *Video segmentation* the goal is to segment each video into *shots* and to select a representative key frame (typically middle frame) from each shot;
2. *Feature extraction* the goal is to extract visual feature vectors from each key frame. We have considered two different types of visual features for this purpose: (i) vectors extracted from MPEG-7 visual descriptors, and (ii) vectors extracted from pre-trained deep-learning networks;
3. *Feature aggregation* feature vectors extracted from the key frame of a video are aggregated over time to obtain a feature vector descriptive of the whole video.
4. *Feature fusion* in this step, features extracted from the same video but with different methods (e.g., MPEG-7 descriptors and deep-learning networks) are combined into a fixed-length descriptor;
5. *Recommendation* the (eventually aggregated and fused) vectors describing low-level visual features of videos are used to feed a recommender algorithm. For this purpose, we have considered the method *Collective SLIM* as a feature-enhanced collaborative filtering (CF) [41].

The flowchart of the methodology is shown in Fig. 2, and the steps are elaborated in more details in the following subsections.

4.1 Video segmentation

Shots are sequences of consecutive frames captured without interruption by a single camera. The transition between two successive shots of the video can be abrupt, where one frame belongs to a shot and the following frame belongs to the next shot, or gradual, where, two shots are combined using chromatic, spatial or spatial-chromatic video production effects (e.g., fade in/out, dissolve, or wipe), which gradually replace one shot by another.

The color histogram distance is one of the most standard descriptors used as a measure of (dis)similarity between consecutive video frames in applications including the content-based video retrieval, object recognition, etc. A histogram is computed for each frame in the video, and the *histogram intersection* is used as the means of comparing the local activity according to Eq. 1,

$$\mathbf{s}(h_t, h_{t+1}) = \sum_b \min(h_t(b), h_{t+1}(b)) \quad (1)$$

where h_t and h_{t+1} are histograms of successive frames and b is the index of the histogram bin. By comparing $\mathbf{s}$ with a predefined threshold, we segment the videos in our dataset into shots. We set the histogram similarity threshold to 0.75.

4.2 Feature extraction

For each key frame, visual features are extracted by using either MPEG-7 descriptors or pre-trained deep-learning networks.

4.2.1 MPEG-7 features

The MPEG-7 standard specifies descriptors that allow users to measure visual features of images. More specifically, MPEG-7 specifies 17 descriptors divided into four categories: color, texture, shape, and motion [36]. In our work we have focused our attention on the following five *color* and *texture* descriptors, as previous experiments have proven the expressiveness of color and texture for similarity-based visual retrieval applications [36,54]:

- **Color descriptors**
  - *Scalable color descriptor (SCD)* is the color histogram of an image in the HSV color space. In our implementation we have used SCD with 256 coefficients (histogram bins).
  - *Color structure descriptor (CSD)* creates a modified version of the SCD histogram to take into account the physical position of each color inside the images, and thus it can capture both color content and information about the structure of this content. In our implementation, CSD is described by a feature vector of length 256.
  - *Color layout descriptor (CLD)* is a very compact and resolution-invariant representation of color obtained by applying the DCT transformation on a 2-D array of representative colors in the YUV color space. CLD is described by a feature vector of length 120 in our implementation.
- **Texture descriptors**
  - *Edge histogram descriptor (EHD)* describes local *edge distribution* in the frame. The image is divided into 16 non-overlapping blocks (subimages). Edges within each block are classified into one of five edge categories: vertical, horizontal, left diagonal, right diagonal and non-directional edges. The final local edge descriptor is composed of a histogram with $5 \times 16 = 80$ histogram bins.
  - *Homogeneous texture descriptor (HTD)* describes homogeneous texture regions within a frame, by using a vector of 62 energy values.

4.2.2 Deep-learning features

An alternative way to extract visual features from an image is to use the inner layers of pre-trained deep-learning networks [32]. We have used the 1024 inner neurons of GoogLeNet, a 22 layers deep network trained to classify over 1.2 million images classified into 1000 categories [50]. Each key frame is provided as input to the network, and the activation values of inner neurons are used as visual features for the frame.

4.3 Feature aggregation

The previous step extracts a vector of features from each key frame of a video. We adopt functions to aggregate all these vectors into a single feature vector descriptive of the whole video. The MPEG-7 standard defines an extension of the descriptors to a collection of pictures known as *group of pictures* descriptors based on statistical summarization methods [3,37]. The main aggregation functions are *minimum*, *mean*, *median* and, *maximum*. Inspired by this, our proposed aggregation functions consist of the following:

- *minimum* each element of the aggregated feature vector is the *minimum* of the corresponding elements of the feature vectors from key frames;
- *mean* each element of the aggregated feature vector is the *average* of the corresponding elements of the feature vectors from key frames;
- *median* each element of the aggregated feature vector is the *median* of the corresponding elements of the feature vectors from key frames.
- *maximum* each element of the aggregated feature vector is the *maximum* of the corresponding elements of the feature vectors from key frame.

In the experiments, we have applied each aggregation function to both of the MPEG-7 and deep-learning features.

4.4 Feature fusion

Motivated by the approach proposed in [19], we employed the fusion method based on *canonical correlation analysis* (CCA) which exploits the low-level correlation between two sets of visual features and learns a linear transformation to maximize the pairwise correlation between the two sets of MPEG-7 and deep-learning networks features.

This fusion mechanism aims at combining information obtained from different sources of features (i.e., different MPEG-7 descriptors and/or deep-learning layers). CCA is

a popular method in multi-data processing—mainly used to analyze the relationships between two sets of features originated from different sources of information [29,30].

Given two set of features $X \in R^{p\times n}$ and $Y \in R^{q\times n}$, where p and q are the dimension of features extracted from the n items, let $C_{xx} \in R^{p\times p}$ and $C_{yy} \in R^{q\times q}$ be the *between-set* and $C_{xy} \in R^{p\times q}$ be the *within-set* covariance matrix. Also let us define $C \in R^{(p+q)\times(p+q)}$ to be the *overall covariance matrix*—a complete matrix which contains information about association between pairs of features—represented as following

$$C = \begin{pmatrix} C_{xx} & C_{xy} \\ C_{yx} & C_{yy} \end{pmatrix} \tag{2}$$

then CCA aims to find a linear transformation $X^* = W_x^T X$ and $Y^* = W_y^T Y$ that maximizes the pairwise correlation across two feature set as given by

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

where $\text{cov}(X^*, Y^*) = W_x^T C_{xy} W_y$, $\text{var}(X^*) = W_x^T C_{xx} W_x$ and $\text{var}(Y^*) = W_y^T C_{yy} W_y$. We adopt the maximization procedure described in [29] and solving the eigenvalue equation

$$\begin{cases} C_{xx}^{-1} C_{xy} C_{yy}^{-1} C_{yx} \hat{W}_x = \Lambda^2 \hat{W}_x \\ C_{yy}^{-1} C_{yx} C_{xx}^{-1} C_{xy} \hat{W}_y = \Lambda^2 \hat{W}_y \end{cases} \tag{4}$$

where $W_x, W_y \in R^{p\times d}$ are the eigenvectors and Λ^2 is the diagonal matrix of eigenvalues or squares of the *canonical correlations*. Finally, $d = \text{rank}(C_{xy}) \leq \min(n, p, q)$ is the number of nonzero eigenvalues in each equation. After calculating $X^*, Y^* \in R^{d\times n}$, feature-level fusion is performed by concatenating the transformed features:

$$Z^{\text{ccat}} = \begin{pmatrix} X^* \\ Y^* \end{pmatrix} = \begin{pmatrix} W_x^T . X \\ W_y^T . Y \end{pmatrix} \tag{5}$$

4.5 Recommendations

In order to test the effectiveness of low-level visual features in video recommendations, we have experimented with two widely adopted algorithms: a CBF algorithm and a hybrid CF algorithm enriched with side information.

4.5.1 Content-based filtering

We tested a widely used adopted CBF algorithm based on k-nearest neighbors [23] in which the unknown preference score (rating) $\hat{\mathbf{r}}_{ui}$ for user u and item i is computed as an aggregation of the ratings of others, similar items given by:

$$\hat{\mathbf{r}}_{ui} = \frac{1}{\sum_{j\in N_u(i)} s_{ij}} \sum_{j\in N_u(i)} s_{ij}\, \mathbf{r}_{uj} \tag{6}$$

where $N_u(i)$ denotes the items rated by user u most similar to item i and s_{ij} is the similarity score between item i and item j (the CB similarity).

4.5.2 Hybridized CBF with rich side information

In order to test the effectiveness of low-level visual features together with the collaborative knowledge captured by a CF model, we employed *collective sparse linear method* (cSLIM) that includes item features as side information to improve quality of recommendations [41].

This variation of SLIM assumes that there exist correlations between users co-consumption patterns and the CB similarity of the two items intrinsic properties encoded in their side information [41]. In order to enforce this correlation, cSLIM imposes an additional requirement (compared with other variation of SLIM) that both the user-item profile matrix R and the item side information matrix F should be reproduced by the same sparse linear aggregation. Therefore the coefficient matrix S should also satisfy $F \sim FS$ giving rise to the following model

$$\underset{S}{\text{argmin}}\ \alpha \|R - RS\| + (1-\alpha)\|F - FS\| + \gamma \|S\| \tag{7}$$

where R is the user-rating matrix, F is the feature-item matrix, and α is a parameter controlling the relative importance of each of the two error terms, and γ is a regularization term. Both terms are found experimentally. The algorithm is trained using Bayesian pairwise ranking (BPR) [44].

5 Evaluation and results

5.1 Dataset

We have used the latest version of the 20M Movielens dataset [31]. For each movie in the Movielens dataset, the title has been automatically queried in YouTube[2] to search for the trailer. These movies are deeply analyzed, and the features are extracted.

The final dataset contains 8,931,665 ratings, and 586,994 tags, provided by 242,209 users to 3964 movies (sparsity 99.06%) classified along 19 genres, i.e., action, adventure, animation, children's, comedy, crime, documentary, drama,

[2] www.youtube.com.

Table 1 Comparing different types of used features

	MPEG-7		DNN	Genre	Tag
	Color	Texture			
Length (per movie)	632	142	1024	19	∼148
Density (%)	100	100	100	100	4

fantasy, film-noir, horror, musical, mystery, romance, sci-fi, thriller, war, western, and unknown (Table 1).

For each movie, the corresponding video trailer is available. Low-level visual features are extracted from the trailers according to the methodology described in the previous section. The dataset, together with trailers and low-level features, is available for download.[3]

In order to evaluate the effectiveness of low-level visual features, we have used two baseline set of traditional attributes *i.e.,* genre and tag. We have adopted latent semantic analysis (LSA) to pre-process the tag-item matrix in order to better exploit the implicit structure in the association between tags and items. The technique includes decomposing the tag-item matrix into a set of orthogonal factors whose linear combination approximates the original matrix [11].

5.2 Methodology

We evaluate the Top-N recommendation quality by using a procedure similar to the one described in [12].

- We perform evaluation by splitting the ratings of the user-item rating matrix into non-overlapping subsets where 80% of ratings are used as train and 20% as test.
- We measure the quality of the recommendations in terms of recall, precision, F1, and MAP for different cutoff values $N = 1$, 10, and 20.

5.3 Preliminary analysis: DNN aggregation method

In a preliminary experiment, we tested different aggregation methods, i.e., each of the deep-learning (DNN) features aggregated with the four different aggregation functions (min, max, mean, and median). The results are summarized in Table 2.

Our first goal is to identify the most useful aggregation function for DNN features in terms of the accuracy of recommendations. Overall, almost in terms of all evaluation metrics and across all the cutoff values, the best aggregation function for DNN features is mean, resulting in the highest quality of recommendation. We keep this feature aggregation for reporting the subsequent results.

[3] recsys.deib.polimi.it.

After the preliminary analysis, we validate the effectiveness of different features and their combinations based on two recommendation models presented above: a pure CBF, using k-nearest neighbor and a CF+CBF model using a variant of SLIM. We feed each of these two recommendation models with visual features based on MPEG-7 and deep-learning networks as well as their combination generated using the CCA method. Finally, we use two types of baselines defined by genre labels and tag attributes, where tags are represented with LSA.

5.4 Results: content-based filtering (CBF)

Figure 3 presents the results of the experiment, obtained by adopting the CBF algorithm, in terms of precision, recall, F1 and MAP across various cutoff values, $N = 1$, 10, and 20.

As it can be seen, using pure CBF model, the visual features based on MPEG-7 achieves the best results in comparison with other features in terms of all presented metrics and across all cutoff values. The next best recommendation quality is achieved by exploiting tag attributes. Surprisingly, in this setting, using MPEG-7 together with DNN feature performs not better than the other baselines, i.e., genre and DNN features. This indicates that movies that are similar in terms of MPEG-7 explicit features are not necessarily similar in terms of DNN latent features and using them together, fed to the basic CBF algorithm, may actually ruin the effectiveness of both.

In fact, the MPEG-7 and deep-learning networks features, capture different visual properties in an image. While MPEG-7 features focus on stylistic properties in a video (i.e., mise-en-scène elements such as color, texture, or lighting), deep-learning networks features capture the semantics content of a video (e.g., objects, people, etc). The results obtained show that MPEG-7 can better describe the perceived similarities between movies with respect to automatically extracted semantic features (DNN) which can be explained by the fact that the DNN can recognize relevant semantic features (such as actors, which can be relevant in comparing movies) as well as non-relevant semantic features (such as dogs, buildings, cars, landscape, which may not be relevant in comparing movies). These non-relevant semantic features create noise that undermine the efficacy of all DNN features (as well as the efficacy of MPEG-7 features in the MPEG-7 + DNN scenario) in the pure CBF scenario.

5.5 Results: hybridized content-based filtering (CF+CBF)

Figure 4 presents the results of the experiment, obtained by adopting the cSLIM hybrid algorithm, in terms of the considered different evaluation metrics and cutoff values. As it can be seen in Fig. 4, different from the previous results, in

Table 2 Comparison of best aggregation functions for DNN features

Feature	Aggr.	Recall			Precision			MAP		
		@1	@10	@20	@1	@10	@20	@1	@10	@20
DNN	Min	0.0084	0.0198	0.0890	0.0055	0.0042	0.0029	0.0061	0.0064	0.0076
DNN	Max	0.0083	0.0203	**0.1036**	0.0058	**0.0046**	**0.0037**	0.0061	0.0064	**0.0082**
DNN	Mean	**0.0092**	**0.0216**	0.0927	**0.0063**	**0.0046**	0.0031	**0.0064**	**0.0067**	**0.0082**
DNN	Median	0.0090	0.0205	0.0933	0.0060	0.0045	0.0031	0.0063	0.0066	**0.0082**

Bold values represent the best values achieved

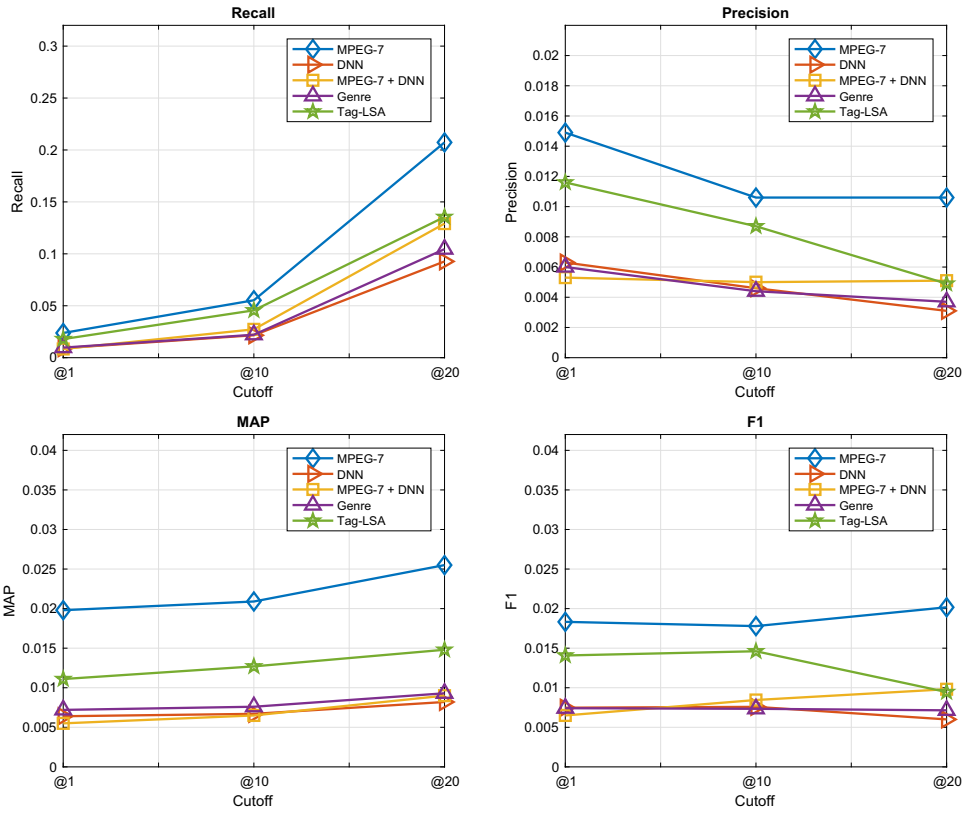

Fig. 3 Comparison of MPEG-7 and DNN features with genres and tags: CBF

terms of Recall, MPEG-7 + DNN outperforms the rest features and combinations while MPEG-7 alone has shown the second best result.

The contrasting nature of the results obtained by the hybrid recommender compared with the results of the CBF model can be due the differences in the nature of the core recommendation algorithm. When used with the cSLIM hybrid algorithm, all features (MPEG-7 and DNN) are automatically (implicitly) weighted by the algorithm (the weights being computed from the ratings of the users). Therefore, cSLIM is able to clean out the non-relevant features extracted by the DNN. More technically, cSLIM can smooth down the similarities between items that are obtained with non-relevant features, if these similarities are not confirmed by the true opinions of the users.

In terms of precision, on the other hand, MPEG-7 achieves the superior results in comparison with the other features. Finally, in terms of MAP and F1, using MPEG-7 individually and combined with DNN features exhibit excellent performance. Unexpectedly, the recommendation based on tag does not show a good performance.

Overall, we can conclude that the obtained results are very promising and they present the power of recommendation based on MPEG-7 features, used individually or in combination with DNN features. Indeed, the results show that recommendation based on MPEG-7 features outperform genre- and tag-based recommendations (CBF), and the combination of MPEG-7 features with deep-learning networks substantially improves the quality of the hybrid recommender algorithm (CF+CBF) and provides the best recommendation results overall.

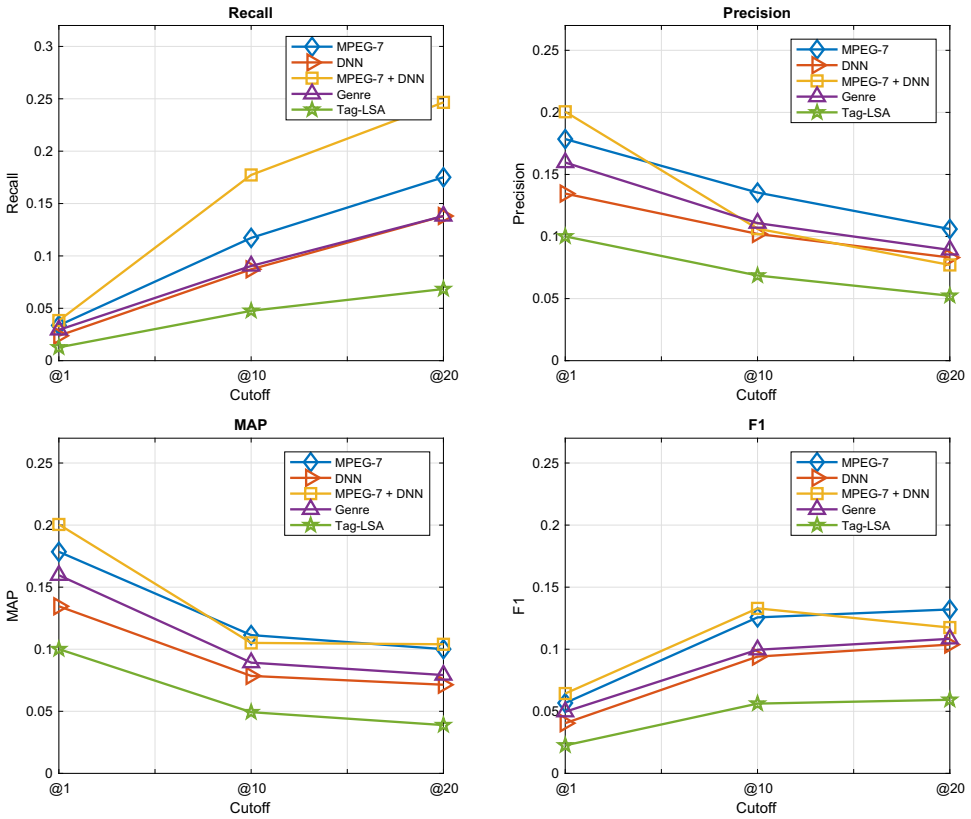

Fig. 4 Comparison of MPEG-7 and DNN features with genre and tag attributes: hybridized CBF (cSLIM)

6 Discussion

Our results provide an empirical evidence that visual low-level features extracted from MPEG-7 descriptors provide better top-N recommendations than traditional attributes, such as genre and tags, while the same may not apply to visual low-level features extracted from pre-trained deep-learning networks.

These two categories of features (MPEG-7 and DNN) can encapsulate different levels of expressiveness. While MPEG-7 captures the syntactic or stylistic (mise-en-scène) properties of a video (e.g., color, texture, and lighting), DNN features are expected to uncover the semantic properties of a video (e.g., objects, people, etc.,). DNN features are more similar to traditional textual attributes of videos, that focus more on the content of the video, not on its style. Moreover, MPEG-7 are mature and well-consolidated features, thanks for being part of the MPEG standardized video compression, while DNN features can be seen as being in their early path at least for application domains relating to RSs. This opens up new research ideas toward this direction.

The conclusions drawn are not tied to a particular recommender algorithm. In our experiments, we have tested two different algorithms that have shown the superior performance of recommendation generation based on visual features in adopting both of these algorithms. We believe that these results man be generalized to other algorithms as well.

Taking everything into account, our experiments can be seen as a proof of the effectiveness of movie recommendations based on visual features, automatically extracted from movies. Recommendations based on deep-learning networks visual features can provide good quality recommendations, in line with recommendations based on human-generated attributes, such as genres and tags, while visual features extracted from MPEG-7 descriptors consistently provide better recommendations. Moreover, fusion of the deep-learning networks and MPEG-7 visual features leads to the best hybrid recommendations.

We would view these results as a good news for practitioners of movie recommender systems, as low-level features combine multiple advantages.

First, mise-en-scène features have the convenience of being computed automatically from movie files, offering designers more flexibility in handling new items, without the need to wait for costly editorial or crowd-based tagging. Moreover, it is also possible to extract low level features from movie trailers, without the need to work on full-length movies [17]. This guarantees good scalability. Finally, viewers are less consciously aware of movie styles and we expect that recommendations based on low-level features could be

more attractive in terms of diversity, novelty and serendipity, as well.

We would like to offer an explanation as to why mise-en-scène low-level features consistently deliver better top-N recommendations than a much large number of high-level attributes. This may have to do with a limitation of high-level attributes, which are binary in nature: movies either have or not have a specific attribute. On the contrary low-level features are continuous in their values and they are present in all movies, but with different weights.

A potential difficulty in exploiting mise-en-scène low-level visual features is the computational load required for the extraction of features from full-length movies. However, we have observed that low-level visual features extracted from the movie trailers are highly correlated with the corresponding features extracted from full-length movies [17]. Accordingly, the observed strong correlation indicates that the movie trailers are indeed perfect representatives of the corresponding full-length movies. Hence, instead of analyzing the lengthy full movies the trailers can be properly analyzed which can result in substantial reduction in the computational load of using mise-en-scène low-level visual features.

7 Conclusion and future work

This work presents a novel approach in the domain of movie recommendations. The technique is based on the analysis of movie content and extraction of stylistic low-level features that are used to generate personalized recommendations for users. This approach makes it possible to recommend items to users without relying on any high-level semantic features (such as, genre, or tag) that are expensive to obtain, as they require expert level knowledge, and shall be missing (e.g., in the new item scenario).

While the results of this study would not underestimate the importance of high-level semantic features, however, they provide a strong argument for exploring the potential of low-level visual features that are automatically extracted from movie content.

For future work, we consider the design and development of an online web application in order to conduct online studies with real users. The goal is to evaluate the effectiveness of recommendations based on low-level visual features not only in terms of relevance, but also in terms of novelty and diversity. For that, we will investigate the potential interface design patterns [9,10,18], and explanation techniques [28], frequently used in recommender systems, that can enable the system to assist the users to understand why certain movies are recommended for them.

Moreover, we will extend the range of low-level features extracted, and also, include audio features (see [14]). We plan to conduct further experiments and compare the quality of the proposed low-level features with other traditional attributes, e.g., those collected in [40] through the crowd-sourcing.

Finally, we would feed the MPEG-7 features as input to the initial layer of deep-networks and build the model accordingly. We are indeed, interested to investigate the possible improvement of recommendation based on the features provided by the deep-networks trained in this way.

Acknowledgements This work is supported by Telecom Italia S.p.A., Open Innovation Department, Joint Open Lab S-Cube, Milan. The work has been also supported by the Amazon AWS Cloud Credits for Research program.

References

1. Adomavicius G, Tuzhilin A (2005) Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans Knowl Data Eng 17(6):734–749
2. Bao X, Fan S, Varshavsky A, Li K, Roy Choudhury R (2013) Your reactions suggest you liked the movie: automatic content rating via reaction sensing. In: Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing. ACM, pp 197–206
3. Bastan M, Cam H, Gudukbay U, Ulusoy O (2010) Bilvideo-7: an mpeg-7-compatible video indexing and retrieval system. IEEE MultiMed 17(3):62–73
4. Bogdanov D, Serrà J, Wack N, Herrera P, Serra X (2011) Unifying low-level and high-level music similarity measures. IEEE Trans Multimed 13(4):687–701
5. Braunhofer M, Elahi M, Ricci F (2014) Techniques for cold-starting context-aware mobile recommender systems for tourism. Intelligenza Artificiale 8(2):129–143
6. Brezeale D, Cook DJ (2008) Automatic video classification: a survey of the literature. IEEE Trans Syst Man Cybern Part C Appl Rev 38(3):416–430
7. Buckland W (2008) What does the statistical style analysis of film involve? A review of moving into pictures. More on film history, style, and analysis. Lit Linguist Comput 23(2):219–230
8. Cantador I, Szomszor M, Alani H, Fernández M, Castells P (2008) Enriching ontological user profiles with tagging history for multi-domain recommendations. In: 1st International workshop on collective semantics: collective intelligence & the semantic web (CISWeb 2008), Tenerife, Spain
9. Cremonesi P, Elahi M, Garzotto F (2015) Interaction design patterns in recommender systems. In: Proceedings of the 11th biannual conference on Italian SIGCHI chapter. ACM, pp 66–73
10. Cremonesi P, Elahi M, Garzotto F (2017) User interface patterns in recommendation-empowered content intensive multimedia applications. Multimed Tools Appl 76(4):5275–5309
11. Cremonesi P, Garzotto F, Negro S, Papadopoulos AV, Turrin R (2011) Looking for good recommendations: a comparative evaluation of recommender systems. In: Human–computer interaction–INTERACT 2011. Springer, pp 152–168
12. Cremonesi P, Koren Y, Turrin R (2010) Performance of recommender algorithms on top-n recommendation tasks. In: Proceedings of the 2010 ACM conference on recommender systems, RecSys 2010, Barcelona, Spain, September 26–30, 2010, pp 39–46

13. Deldjoo Y, Atani RE (2016) A low-cost infrared-optical head tracking solution for virtual 3d audio environment using the nintendo wii-remote. Entertain Comput 12:9–27
14. Deldjoo Y, Constantin MG, Schedl M, Ionescu B, Cremonesi P (2018) Mmtf-14k: a multifaceted movie trailer feature dataset for recommendation and retrieval. In: Proceedings of the 9th ACM multimedia systems conference. ACM
15. Deldjoo Y, Cremonesi P, Schedl M, Quadrana M (2017) 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. ACM, p 20
16. Deldjoo Y, Elahi M, Cremonesi P, Garzotto F, Piazzolla P (2016) Recommending movies based on mise-en-scene design. In: Proceedings of the 2016 CHI conference extended abstracts on human factors in computing systems. ACM, pp 1540–1547
17. Deldjoo Y, Elahi M, Cremonesi P, Garzotto F, Piazzolla P, Quadrana M (2016) Content-based video recommendation system based on stylistic visual features. J Data Semant 5:1–15
18. Deldjoo Y, Elahi M, Quadrana M, Cremonesi P, Garzotto F (2015) Toward effective movie recommendations based on mise-en-scène film styles. In: Proceedings of the 11th biannual conference on Italian SIGCHI chapter. ACM, pp 162–165
19. Deldjoo Y, Elahi Y, Cremonesi P, Moghaddam FB, Caielli ALE (2017) How to combine visual features with tags to improve movie recommendation accuracy? In: E-commerce and web technologies: 17th international conference, EC-Web 2016, Porto, Portugal, September 5–8, 2016, Revised Selected Papers, vol. 278. Springer, p 34
20. Deldjoo Y, Frà C, Valla M, Cremonesi P (2017) Letting users assist what to watch: an interactive query-by-example movie recommendation system. In: Proceedings of the 8th Italian information retrieval workshop, Lugano, Switzerland, June 05–07, 2017, pp 63–66. http://ceur-ws.org/Vol-1911/10.pdf. Accessed 15 Dec 2017
21. Dorai C, Venkatesh S (2001) Computational media aesthetics: finding meaning beautiful. IEEE MultiMed 8(4):10–12
22. Elahi M, Braunhofer M, Ricci F, Tkalcic M (2013) Personality-based active learning for collaborative filtering recommender systems. In: Congress of the Italian association for artificial intelligence. Springer, pp 360–371
23. Elahi M, Deldjoo Y, Bakhshandegan Moghaddam F, Cella L, Cereda S, Cremonesi P (2017) Exploring the semantic gap for movie recommendations. In: Proceedings of the eleventh ACM conference on recommender systems. ACM, pp 326–330
24. Elahi M, Ricci F, Repsys V (2011) System-wide effectiveness of active learning in collaborative filtering. In: Proceedings of the international workshop on social web mining, co-located with IJCAI, Barcelona, Spain
25. Elahi M, Ricci F, Rubens N (2013) Active learning strategies for rating elicitation in collaborative filtering: a system-wide perspective. ACM Trans Intell Syst Technol (TIST) 5(1):13
26. Elahi M, Ricci F, Rubens N (2016) A survey of active learning in collaborative filtering recommender systems. Comput Sci Rev 20:29–50
27. Fleischman M, Hovy E (2003) Recommendations without user preferences: a natural language processing approach. In: Proceedings of the 8th international conference on Intelligent user interfaces. ACM, pp 242–244
28. Gedikli F, Jannach D, Ge M (2014) How should i explain? A comparison of different explanation types for recommender systems. Int J Hum Comput Stud 72(4):367–382
29. Haghighat M, Abdel-Mottaleb M, Alhalabi W (2016) Fully automatic face normalization and single sample face recognition in unconstrained environments. Expert Syst Appl 47:23–34
30. Hardoon DR, Szedmak S, Shawe-Taylor J (2004) Canonical correlation analysis: an overview with application to learning methods. Neural Comput 16(12):2639–2664
31. Harper FM, Konstan JA (2015) The movielens datasets: history and context. ACM Trans Interact Intell Syst (TiiS) 5(4):19
32. He R, McAuley J (2015) Vbpr: visual bayesian personalized ranking from implicit feedback. arXiv preprint arXiv:1510.01784
33. Hu W, Xie N, Li L, Zeng X, Maybank S (2011) A survey on visual content-based video indexing and retrieval. IEEE Trans Syst Man Cybern Part C Appl Rev 41(6):797–819
34. Jakob N, Weber SH, Müller MC, Gurevych I (2009) Beyond the stars: exploiting free-text user reviews to improve the accuracy of movie recommendations. In: Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion. ACM, pp 57–64
35. Lika B, Kolomvatsos K, Hadjiefthymiades S (2014) Facing the cold start problem in recommender systems. Expert Syst Appl 41(4):2065–2073
36. Manjunath BS, Ohm JR, Vasudevan VV, Yamada A (2001) Color and texture descriptors. IEEE Trans Circuits Syst Video Technol 11(6):703–715
37. Manjunath BS, Salembier P, Sikora T (2002) Introduction to MPEG-7: multimedia content description interface, vol 1. Wiley, Chichester
38. Melville P, Sindhwani V (2011) Recommender systems. In: Encyclopedia of machine learning. Springer, pp 829–838
39. Musto C, Narducci F, Lops P, Semeraro G, de Gemmis M, Barbieri M, Korst J, Pronk V, Clout R (2012) Enhanced semantic tv-show representation for personalized electronic program guides. In: User modeling, adaptation, and personalization. Springer, pp 188–199
40. Nasery M, Elahi M, Cremonesi P (2015) Polimovie: a feature-based dataset for recommender systems. In: ACM RecSys workshop on crowdsourcing and human computation for recommender systems (CrawdRec), vol 3, pp 25–30
41. Ning X, Karypis G (2012) Sparse linear methods with side information for top-n recommendations. In: Proceedings of the sixth ACM conference on Recommender systems. ACM, pp 155–162
42. Rasheed Z, Shah M (2003) Video categorization using semantics and semiotics. In: Video mining. Springer, pp 185–217
43. Rasheed Z, Sheikh Y, Shah M (2005) On the use of computable features for film classification. IEEE Trans Circuits Syst Video Technol 15(1):52–64
44. Rendle S, Freudenthaler C, Gantner Z, Schmidt-Thieme L (2009) Bpr: Bayesian personalized ranking from implicit feedback. In: Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence. AUAI Press, pp 452–461
45. Rubens N, Elahi M, Sugiyama M, Kaplan D (2015) Active learning in recommender systems. In: Recommender systems handbook. Springer, pp 809–846
46. Saveski M, Mantrach A (2014) Item cold-start recommendations: learning local collective embeddings. In: Proceedings of the 8th ACM conference on recommender systems. ACM, pp 89–96
47. Schedl M, Zamani H, Chen CW, Deldjoo Y, Elahi M (2018) Current challenges and visions in music recommender systems research. Int J Multimed Inf Retr. https://doi.org/10.1007/s13735-018-0154-2
48. Schein AI, Popescul A, Ungar LH, Pennock DM (2002) Methods and metrics for cold-start recommendations. In: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, pp 253–260
49. Shi Y, Larson M, Hanjalic A (2014) Collaborative filtering beyond the user-item matrix: a survey of the state of the art and future challenges. ACM Comput Surv (CSUR) 47(1):3
50. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1–9

51. Szomszor M, Cattuto C, Alani H, O'Hara K, Baldassarri A, Loreto V, Servedio VDP (2007) Folksonomies, the semantic web, and movie recommendation. In: 4th European Semantic Web Conference, Bridging the Gap between Semantic Web and Web 2.0, Innsbruck, Austria
52. Tubularinsights: 500 hours of video uploaded to youtube every minute [forecast]. http://tubularinsights.com/hours-minute-uploaded-youtube/. Accessed 19 Jan 2018
53. Vig J, Sen S, Riedl J (2009) Tagsplanations: explaining recommendations using tags. In: Proceedings of the 14th international conference on intelligent user interfaces. ACM, pp 47–56
54. Wang XY, Zhang BB, Yang HY (2014) Content-based image retrieval by integrating color and texture features. Multimed Tools Appl 68(3):545–569
55. Wang Y, Xing C, Zhou L (2006) Video semantic models: survey and evaluation. Int J Comput Sci Netw Secur 6:10–20
56. Xu S, Jiang H, Lau F (2008) Personalized online document, image and video recommendation via commodity eye-tracking. In: Proceedings of the 2008 ACM conference on recommender systems. ACM, pp 83–90
57. Yang B, Mei T, Hua XS, Yang L, Yang SQ, Li M (2007) Online video recommendation based on multimodal fusion and relevance feedback. In: Proceedings of the 6th ACM international conference on image and video retrieval. ACM, pp 73–80
58. Zettl H (2002) Essentials of applied media aesthetics. In: Dorai C, Venkatesh S (eds) Media computing. The Springer international series in video computing, vol 4. Springer, Berlin, pp 11–38
59. Zettl H (2013) Sight, sound, motion: applied media aesthetics. Cengage Learning, Boston
60. Zhang ZK, Liu C, Zhang YC, Zhou T (2010) Solving the cold-start problem in recommender systems with social tags. EPL (Europhys Lett 92(2):28,002
61. Zhao X, Li G, Wang M, Yuan J, Zha ZJ, Li Z, Chua TS (2011) Integrating rich information for video recommendation with multi-task rank aggregation. In: Proceedings of the 19th ACM international conference on Multimedia. ACM, pp 1521–1524
62. Zhou H, Hermans T, Karandikar AV, Rehg JM (2010) Movie genre classification via scene categorization. In: Proceedings of the international conference on multimedia. ACM, pp 747–750