A MULTI–AGENT BROKERAGE PLATFORM FOR MEDIA CONTENT RECOMMENDATION

BRUNO VELOSO [a,c], BENEDITA MALHEIRO [b,c], JUAN CARLOS BURGUILLO [a,*]

[a] School of Telecommunication Engineering
University of Vigo, Campus Universitario, E-36310 Vigo, Spain
e-mail: `bruno.miguel.veloso@gmail.com,J.C.Burguillo@uvigo.es`

[b] School of Engineering
Polytechnic Institute of Porto, Rua Dr. António Bernardino de Almeida, 431, 4249-015 Porto, Portugal
e-mail: `mbm@isep.ipp.pt`

[c] INESC TEC
Campus da FEUP, Rua Dr. Roberto Frias 4200, 465 Porto, Portugal

Near real time media content personalisation is nowadays a major challenge involving media content sources, distributors and viewers. This paper describes an approach to seamless recommendation, negotiation and transaction of personalised media content. It adopts an integrated view of the problem by proposing, on the business-to-business (B2B) side, a brokerage platform to negotiate the media items on behalf of the media content distributors and sources, providing viewers, on the business-to-consumer (B2C) side, with a personalised electronic programme guide (EPG) containing the set of recommended items after negotiation. In this setup, when a viewer connects, the distributor looks up and invites sources to negotiate the contents of the viewer personal EPG. The proposed multi-agent brokerage platform is structured in four layers, modelling the registration, service agreement, partner lookup, invitation as well as item recommendation, negotiation and transaction stages of the B2B processes. The recommendation service is a rule-based switch hybrid filter, including six collaborative and two content-based filters. The rule-based system selects, at runtime, the filter(s) to apply as well as the final set of recommendations to present. The filter selection is based on the data available, ranging from the history of items watched to the ratings and/or tags assigned to the items by the viewer. Additionally, this module implements (i) a novel item stereotype to represent newly arrived items, (ii) a standard user stereotype for new users, (iii) a novel passive user tag cloud stereotype for socially passive users, and (iv) a new content-based filter named the collinearity and proximity similarity (CPS). At the end of the paper, we present off-line results and a case study describing how the recommendation service works. The proposed system provides, to our knowledge, an excellent holistic solution to the problem of recommending multimedia contents.

Keywords: multi-agent computing, brokerage platform, media content personalisation, recommendation.

1. Introduction

Media content personalisation poses a significant challenge to media content producers, distributors and viewers. The goal of media content producers is to place their media items at the highest price possible, and the goal of a content distributor is to strengthen the relationship with existing customers and increase the number of customers by offering the most appropriate media content from a multitude of sources, while taking

the viewer profile into account. The viewers are unable to search, find and choose their preferred content in near real time due to the size of the search space, both in terms of the number of offers and the diversity of sources. The nature of the relationship between content producers and content distributors is business-to-business (B2B), whereas the nature of the relationship between distributors and viewers is business-to-consumer (B2C).

The proposed approach—the multimedia brokerage (MMB) platform—integrates the B2B and B2C

---

*Corresponding author

perspectives, including recommender and negotiation services for media content producer and distributor businesses. In terms of the B2B model, the platform implements content provider lookup, just-in-time negotiation and pay-per-view charging, i.e., only relevant items are negotiated and only those that are actually watched by the viewers will be transacted. Distributor agents look up, invite and negotiate the items to fill the personalised viewer programme guides that will be presented to the viewers with producer agents in just-in-time. Producer agents incorporate and use a recommendation service to choose the items for negotiation. The selected items are negotiated between distributor and producer delegate agents and the outcome is a personalised electronic programming guide (EPG). In terms of the B2C model, the platform implements content provider lookup, just-in-time negotiation and pay-per-view charging.

The recommendation service builds stereotypes for new items, new users and passive users, and implements a rule-based switch hybrid filter that includes content-based and collaborative filters. Experiments were conducted with the HetRec 2011 data set, which contains films, user ratings and tags, to evaluate the quality and accuracy of the recommendations produced. The metrics used were the mean absolute error (MAE), the root mean square error (RMSE), *Recall*, *Precision* and *F-measure*. The F-measure was adopted as the final evaluation metric.

The creation of new users, new items and passive user stereotypes intends to address the first-rater, cold-start and the socially inactive user problems. The problems take place respectively when new products or new users are introduced in the system or when users do not rate or tag the products they select. The lack of ratings about these products or from these users prevents collaborative filters from making recommendations (Moreno *et al.*, 2011).

The main contributions of this paper are focused on two different levels: firstly, on the platform level, providing a multi-agent system with support for brokerage, negotiation, transactions and the use of business and Web standards, and secondly, on the recommendation level, with a new set of filters and stereotypes to deal with and to improve the user experience.

Being more specific, at the platform level, the contributions consider: (i) the design and development of a multi-agent system for the brokerage, negotiation and transaction of media items; (ii) the adoption of standard Web service (WS) interfaces for the interaction with businesses; (iii) the adoption of Business Process Modelling Notation (BPMN) to define the behaviour of the business representative agents within the platform; and (iv) the usage of a recommendation service to select the candidate items for negotiation, i.e., those that comply

with the viewer profile.

Considering the contributions related with the recommendation service, we may detail them as (i) the development of a rule-based switch hybrid filter to select, at runtime, the filtering algorithm to apply as well as the final set of recommendations; (ii) a new content-based filter called collinearity and proximity similarity (CPS); and (iii) a fuzzy user collaborative filter. Furthermore, to address the cold-start, first-rater and passive user problems, the following stereotypes are created: (i) a genre-based new item stereotype; (ii) a genre-based new user stereotype relying on the most popular items or on the most popular item of each genre; and (iii) a new passive user tag cloud stereotype for passive users, i.e., those that do neither rate nor tag items.

This paper is organized into eight sections. Section 1 presents the context of the project, identifies the problems addressed and describes the adopted approach. Section 2 discusses the related work in multi-agent systems and compares different multi-agent platforms with the focus on B2B and B2C platforms.

Section 3 presents existing recommender systems and compares different recommendation techniques. Section 4 describes the system architecture and platform design. Section 5 describes the recommendation service and hybrid filter implementation. Section 6 presents the adopted evaluation metrics. Section 7 describes the tests and discusses the results obtained. Finally, Section 8 provides the conclusions and presents future developments.

2. Multi-agent platforms

A system which is composed of a group of interacting agents is called a multi-agent system (MAS). Multi-agent systems can be characterized according to (i) the agent design, i.e., the agents can implement heterogeneous (the agents may have different decision-making or perception properties) or homogeneous (all agents are endowed with the same capabilities) behaviours; (ii) the environment, i.e., the agents can interact with static (a single agent solution) or dynamic (there are multiple agents interacting in the environment) environments; (iii) the perception level, i.e., global or local (each agent can observe a particular event in a different way, making the environment partially observable and leading to different behaviours); (iv) the type of control, i.e., centralized, decentralised (in multi-agent systems decentralized control is typical for reasons of robustness and fault tolerance) and hybrid; (v) the knowledge level, i.e., the knowledge about the environment may differ among agents; (vi) the communication protocol, i.e., agents must adopt a common language. Some of the applications of multi-agent systems include automated negotiation, data filtering and processing,

sensor networks, robotics and social phenomena.

2.1. B2B applications. B2B is a rich and challenging application domain for multi-agent systems. The relationships between businesses are, by nature, decentralised and include both cooperative and competitive scenarios.

The Web service discovery and negotiation system described by Lau (2007) presents a multi-agent system designed for B2B e-commerce. This system relies on Web services to expose the agents' functionalities. A universal description, discovery and integration (UDDI) registry is used to search and publish the services provided by the businesses. The negotiation between businesses is performed by a dedicated negotiation agent. In the MMB platform, businesses, rather than a single negotiation agent rely on ephemeral market delegate agents to negotiate the provisional terms of individual items, minimising the potential bottleneck problem of the single negotiation agent.

The approach proposed by Endert *et al.* (2007) uses an automated mapping from Business Process Modelling Notation, a graphical language employed to represent business processes, to belief, desire and intention (BDI) agent concepts. The authors define a graph transformation system, which is supported by a set of transformation rules, to map elements of the source model (BPMN) to elements of the target model (agents). This complete transformation process involves normalization, element mapping, structure mapping and clean-up stages. In the case of the MMB platform, the mapping from BPMN process diagrams to agent behaviour is supported by the Drools business rule management system, which includes the Java Business Process Management (jBPM) work flow engine, developed by the JBoss community.

2.2. B2C applications. B2C is an application domain where multi-agent systems are frequently adopted to model customers and/or the provided business services.

The B2C multi-agent system described by Rosaci and Sarnè (2013) assigns different user interest coefficients and weights to the identified B2C activities. Profiles and messages exchanged among agents are described in XML and have a common ontology that exploits the NAICS taxonomy. The MMB platform uses the NAICS taxonomy and the IMDb genre taxonomy directly to classify businesses and items, respectively. User profiling is out of scope of the MMB platform.

The recommendation system for personalisation in B2C e-commerce applications described by Zhang and Jiao (2007) is a classifier that uses a data mining algorithm called association rule learning.

In the MMB platform, the producers include a local hybrid recommendation service to choose the items

for negotiation. The hybrid filter implements multiple recommendation algorithms, and uses a rule-based system to dynamically select the filters to apply (based on the data available) and the recommendation set (based on the obtained results).

3. Recommendation systems

Recommendation systems are valuable support tools to help humans or applications to find items of interest whenever the search space is too large. They are frequently used in personalisation and data retrieval tasks, and rely on different type of filters such as content-based, collaborative and hybrid filters. In the specific case of multimedia, content personalisation has become indispensable due to the number of sources and items available. There are multiple examples of content-based filtering (e.g., Di Noia *et al.*, 2012; Pera and Ng, 2013), collaborative filtering (e.g., Al-Shamri, 2014; Zhang *et al.*, 2014), and hybrid filtering, (e.g., Wu *et al.*, 2014; Ghazanfar and Prugel-Bennett, 2010), applied to the multimedia domain.

3.1. Content-based filters. Content-based recommendation algorithms recommend items according to the similarity and dissimilarity calculated between the user profile and the items. This approach is applicable when the data set contains the items viewed per user, the items rated per user or the items rated and tagged per user. The content-based filters that rely solely on the items viewed per user do not experience the cold-start problem, which happens when a new user or item is represented in the system with a blank history. As a whole, content-based filtering (CbF) suffers from over-specialization, i.e., the recommendations are restricted to items identical to the ones present in viewer item history.

There are two techniques applicable to content-based filters: (i) the calculation of similarity, e.g., the cosine similarity; and (ii) the calculation of distances, e.g., the Euclidean distance and Hamming distance.

Distance or dissimilarity metrics are very popular, e.g., such as in the systems proposed by Li *et al.* (2012) or Veloso *et al.* (2013). In the first case (Li *et al.*, 2012), the authors rely on the Hamming distance to aggregate the items into clusters, apply a hidden Markov model to determine their probabilities and, finally, recommend items (films) similar to the user item history. In the second case (Veloso *et al.*, 2013), the system recommends advertisements by determining, among other metrics, the Euclidean distance between the user profile and the candidate advertisements.

The most common similarity metric is the cosine similarity (CS), which determines the collinearity between two arrays of features. For example, the recommendation system developed by

Barragáns-Martínez *et al.* (2010a) uses a cosine similarity content-based filter to minimize the effect of cold-start recommendation.

The content-based recommendation system described by Di Noia *et al.* (2012) works exclusively with data from linked open data (LOD) repositories. The system relies on the vector space model (VSM) to calculate the cosine similarity between RDF resources and allows binary user ratings. Although the MMB platform adopts the VSM in the implemented CbF, it currently does not use LOD sources.

The group recommendation system described by Pera and Ng (2013) generates recommendations for groups of viewers from social networks data. It determines the word-correlation between viewer tags, aggregates the tags in groups and global popularity and presents the top-n suggestions. Although the current work does not address group recommendation specifically, it implements user-based correlation clusters and user-based fuzzy clusters which act as group recommendation filters.

The personal recommendation system described by Kurapati *et al.* (2001) is intended for a single user. It is a multi-agent system composed of three types of recommender agents: the implicit recommender agent, the explicit recommender agent and the feedback agent. The implicit recommender agents suggest items based on the implicit profile of the viewer, using a Bayesian classifier and decision trees. The explicit recommender agents supply recommendations based on the ratings given by the users. The final recommendations result from weighting the outputs of the two recommender agents. The feedback agents allow users to provide input for the implicit and explicit recommenders.

Our switch hybrid filter includes content-based and collaborative filters, as alternatives to a content-based weighted hybrid filter, and also handles multiple viewers, instead of a single home viewer.

3.2. Collaborative filters. Collaborative recommendation is applicable when the data set includes the viewer item history, the history of items rated by the viewer or the history of items tagged by the viewer together with the genre-based description of the items. Collaborative filters (CFs) promote diversity when dealing with different types of content and create an effect of controlled serendipity.

These types of filters suffer from data sparsity and scalability issues as well as from the first-rater (the item cannot be recommended without a user assigning a rating), grey sheep (these are users who agree or disagree partially by other users) and cold-start (the initial lack of ratings) problems.

Collaborative filter has three variants: user-based, item-based and tag-based. Additionally, there are two main implementation techniques: memory-based, e.g.,

the k-nearest neighbour (kNN) and model-based, e.g., singular value decomposition (SVD).

3.2.1. Memory-based. Collaborative filters make predictions based on the available user ratings by calculating the similarity or correlation among users or items. There can be several alternatives:

User-based collaborative recommendation is a three-task process. First, it calculates the similarity among users, e.g., using a correlation coefficient such as the Pearson correlation coefficient. Next, it selects a subset of users based on their similarity to the active user. Finally, it predicts the item ratings based on the combined weights of items viewed by the neighbouring users (Melville *et al.*, 2002; Melville and Sindhwani, 2010). In order to avoid predicting items from users that have co-rated few items, Herlocker *et al.* (1999) propose the usage of a significance weighting factor.

Item-based collaborative recommendation is a three stage filter. First, it calculates the similarity among the items rated by two users, using the Pearson correlation coefficient. Then, it selects the k nearest neighbour items and, finally, it predicts their ratings and suggests the highest ranked items (Papagelis and Plexousakis, 2005).

Tag-based collaborative recommendation consists in the creation of tag clouds, i.e., sets of related tags assigned by the users to items in social networks. There are typically four types of tag clouds: (i) the user tag cloud, which is based on the tags of each user; (ii) the item tag cloud, which is based on the tags of each item; (iii) the target user tag cloud, which is based on the tags of the users who viewed a particular item; and (iv) the stereotype tag cloud, which is based on the tags attributed to items of a given genre (Rey-López *et al.*, 2010; Barragáns-Martínez *et al.*, 2010b).

Our hybrid recommender implements the following memory-based collaborative filters: item-based kNN Pearson correlation, user-based kNN Pearson correlation and a tag-based filter. The user and item-based Pearson correlations are affected by a significance weighting as proposed by Herlocker *et al.* (1999). Since these filters suffer from scalability problems, our recommender implements SVD, clustering and stereotype techniques to minimise this problem.

3.2.2. Model-based. Collaborative filters rely on the history of items rated by the user to estimate or learn a predictive model. There are two main alternatives:

Item-based collaborative recommendation involves four steps. First, it creates a model of the rating matrix using singular value decomposition. Then, it determines the similarity among items that were rated by users using the adjusted cosine similarity. Next, it chooses the k neighbours and, finally, it predicts the item ratings (Vozalis and Margaritis, 2005; Barragáns-Martínez *et al.*, 2010a). To avoid continuous recreation of the reduced SVD matrix, it is possible to adopt the folding-in technique, which allows including new users or items directly in the reduced SVD matrix (Sarwar *et al.*, 2002a). The current work reuses this technique together with a threshold that, once reached, results in the recreation of the reduced SVD matrix.

Clustering is frequently adopted to circumvent the scalability problems that arise when filters are applied to large data sets, i.e., improve the scalability of recommender systems (Sarwar *et al.*, 2002b). Correlation clustering (Bansal *et al.*, 2004) is based on a breadth-first searching algorithm (Yanxiang *et al.*, 2013) and uses the cosine similarity measure. The current work, inspired by this approach, implements two clustering algorithms: a user-based correlation clustering and a user-based fuzzy clustering.

In the collaborative recommendation system described by Al-Shamri (2014), users are compared based on the common liked and disliked items separately. It combines the Dice and Jaccard coefficients into a new correlation coefficient, which allows the tuning of the relevance given to positive matches. The final similarity measure results from averaging the positive and negative correlation coefficient values. The predictions are generated by aggregating the ratings scaled by the similarity of neighbouring users for the same item, taking into account the user mean ratings whenever possible.

The collaborative recommendation system described by Zhang *et al.* (2014) uses the K-means clustering technique to generate clusters of users. The similarity between two users takes into account the user activity (based on item categories) and the Pearson correlation coefficient.

The current approach implements the following model-based collaborative filters: item-based SVD, user-based correlation clustering and user-based fuzzy c-means clustering.

3.3. Stereotypes. Objects can be represented by default stereotypes, when there are not enough data to characterize individuals, or cluster-based stereotypes, when the size occupied by the objects matters. Whereas in the first case stereotypes are based on the properties

of related objects, in the latter case stereotypes are based on the properties of the objects within the cluster. In the case of recommendation systems, stereotypes are used to represent new users/items or clusters of users/items. A practical application of stereotypes is the creation of electronic programme guides (EPGs) targeted for sets of users (Ardissono *et al.*, 2004).

The project described by Sollenborn and Funk (2002) creates user stereotypes and applies a category-based filter to recommend contents. The users are grouped into category-based clusters. The system will automatically attempt to classify a new user by comparing the user profile with the user stereotype clusters.

The Media Scout project (Shani *et al.*, 2007) presents a recommendation system that uses stereotypes to represent groups of users. This system is mainly composed of profile clustering (adopting a distance metric to calculate the similarity between users) and a content-based filter.

In the current work, stereotypes are used when there are not enough data available, e.g., to represent new items (item stereotype), newly arrived users (user stereotype), socially passive users (passive user tag cloud stereotype), or when multiple users or items are aggregated, e.g., in clustering. The obtained results show that these stereotypes mitigate the first rater problem and the lack of information resulting from a passive user.

3.4. Hybrid filters. Hybrid filters typically combine several of the presented techniques. There are, however, different combination approaches that Burke (2002) organises into seven methods: weighted, switching, mixed, feature combination, cascade, feature augmentation and meta-level. In the weighted methodology, the scores of several recommendation techniques are combined to produce a single recommendation. The switching methodology switches between different recommendation techniques depending on the situation or data available. In the mixed methodology, the system presents simultaneous recommendations from different techniques. In the feature combination method, the features from different data classes are used together in a single recommendation algorithm. The cascade methodology uses the output of one recommendation algorithm to refine the recommendation produced by another. In the feature augmentation methodology, the result produced by a recommendation technique is used as input data for another technique. Finally, in the meta-level methodology, the model learned by a recommendation system is used as input data to another recommendation system.

The queveo.tv project (Barragáns-Martínez *et al.*, 2010a) uses a hybrid filter that mixes two techniques: a content-based filter (CS) and a collaborative filter

(item-based SVD). The generated recommendations are presented together, highlighting with a star the items simultaneously recommended by both filters. SVD is typically computed off-line and the on-line system dynamically provides predictions based on the stored SVD. This approach can generate outdated predictions. To overcome this drawback, our SVD implementation uses the folding-in technique proposed by Sarwar *et al.* (2002a) to update, in runtime, the SVD matrix.

The feature augmentation hybrid filter described by Wu *et al.* (2014) first generates a content-based model, then applies a co-clustering technique to produce clusters, and finally produces recommendations using a dual technique collaborative filter: item-based and user-based filters.

The cascade hybrid filter described by Ghazanfar and Prugel-Bennett (2010) uses the item ratings, item features (extracted from the IMDb movie info) and item demographic information (IMDb movie genres). First, it calculates the adjusted cosine similarity between the item ratings, then calculates the cosine similarity between the item demographic and item feature vectors and, finally, generates the predictions that maximize the defined utility function, which is a linear combination of the ranking, feature and demographic item similarities. ReSySTER is another hybrid recommender system that relies on cascading fuzzy logic, rough set theory and semantic technologies (Colomo-Palacios *et al.*, 2012). It assists software development project managers configuring the best team for the defined work packages based on Scrum team roles.

Our project builds stereotypes for new users and new items and implements a switch hybrid filter that includes several content-based and collaborative filters. The hybrid filter relies on an inference rule system to select the most appropriate filter and the final set of recommendations. This approach is detailed in Section 5.

4. Brokerage platform

Figure 1 depicts the architecture of the multi-agent platform which is organized into interface (1), agreement (2), enterprise (3) and market (4) layers. There are five types of agents in the platform: (i) interface agents to interact with businesses, (ii) SLA agents to celebrate and verify the fulfilment of the service level agreements (SLAs), (iii) enterprise agents that model the businesses, (iv) market delegate agents dedicated to the negotiation of specific items or advertising intervals on behalf of the enterprise agents, and (v) layer manager (L) agents (agreement, interface, enterprise and market layer agents).

Each business (producer (P) or distributor (D) enterprise) is represented at the platform by the corresponding: (i) an enterprise interface agent located in interface layer, (ii) an SLA agent located in agreement layer, (iii) enterprise agent that models the enterprise within the platform in the enterprise layer, and (iv) an undetermined number of delegate agents involved in a specific item or interval negotiations in the market layer. These agents are identified by a trading code, preventing external actors from intervening in the undergoing negotiations. With the exception of the delegate market agents, all agents offer a standard WS interface.
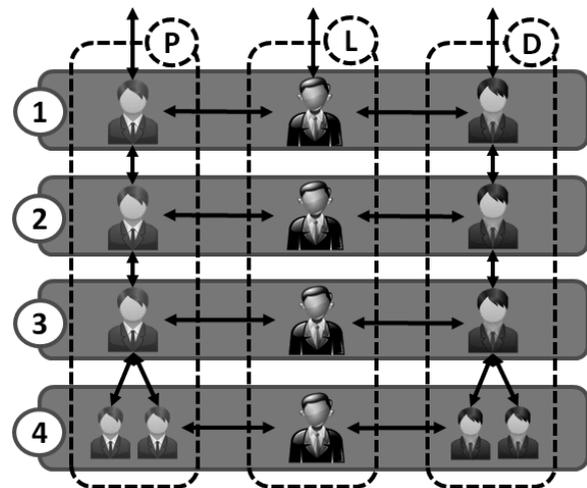

Fig. 1. Platform architecture.

4.1. Business registration and data uploading.
A business wishing to join the platform first needs to register with the interface administrator agent. Once registered, the administrator creates a dedicated interface agent to interact with the business. The interface agent provides a graphical user interface (GUI) for the uploading of the business processes, using the BPMN standard, and the business inference rules. With this information, the dedicated interface agent (i) checks whether the business is registered in the platform UDDI with its corresponding North American Industry Classification System (NAICS) code, and if not, registers the business in the UDDI platform registry; (ii) stores the business process and rule files in a local folder; and (iii) creates the enterprise agent that models the business within the platform.

Whenever a producer or distributor business registers with the platform, a new SLA is established between the business and platform, defining the terms of the provisioning brokerage services. When a producer uploads a new item, a new service level agreement template is created and the item genre-based profile is registered at the UDDI registry for automatic recommendation and negotiation. The specific SLA terms will be negotiated at the market layer between producer and distributor delegates, defining the item provisioning conditions. When a distributor uploads a new viewer profile, it immediately triggers the provider

lookup mechanism. Currently, the agreement layer is under implementation.

4.2. Brokerage. The brokerage mechanism relies on the producer and distributor agents' implemented behaviour: (i) producer agents register their multimedia items at the UDDI platform registry service using the IMDb genres taxonomy; (ii) distributor agents interact with the UDDI registry service to lookup and invite all producers with items that have an IMDb genre matching their viewer profile; and (iii) the distributor sends the viewer's profile, including the watched item history, tags and ratings to the subset of producers to invite them to the negotiation of a personalized EPG. It is a prerequisite, specified at the contractual level, that distributors must provide a minimum subset of the viewer data they hold. The lookup mechanism intends to select the subset of producers that actually holds items of interest to the viewer. Our purpose is to exclude *a priori* all producers without items that may interest the viewer from the recommendation and negotiation process.

4.3. Recommendation of candidate items. An invited producer, after receiving the viewer data, invokes the recommendation service. First, if faced with new items, a new or a passive user, he builds the corresponding stereotypes. Then, the hybrid filter uses the rule-based system to dynamically select the filters to apply. The items submitted to the hybrid filter were never suggested before by the producer to the viewer, avoiding the repetition of items. This stage terminates with the selection of the item that will be negotiated by a producer delegate agent.

4.4. Negotiation. Figure 2 represents the market layer architecture. One business may participate in multiple trading rooms by launching multiple delegate agents, i.e., a business can negotiate simultaneously n products. Each delegate agent is identified by a universally unique identifier (UUID) code generated when the producer is invited to negotiate. This approach allows using different negotiation protocols in distinct trading rooms. The implemented negotiation protocols include the Foundation for Physical Intelligent Agents (FIPA) Iterated Contract Net (ICNET) as well as English and Dutch auctions.

The outcome of each negotiation round is an item which will be added to the viewer's EPG. Since the platform adopts the pay-per-view model, a successful negotiation between a producer and a distributor does not imply immediate payment for the negotiated item. The actual transaction and payment only take place when the viewer clicks, rates or tags a negotiated item. In this case, the distributor reports to the producer on the viewer, item viewed, rating given or tagging, after which the actual
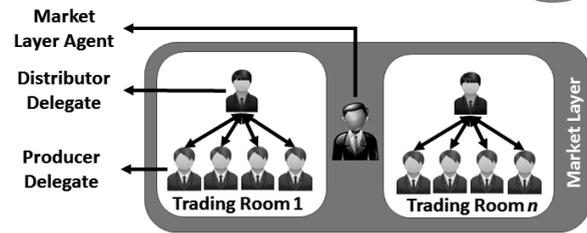

Fig. 2. Market layer.

transaction between distributor and the producer can take place.
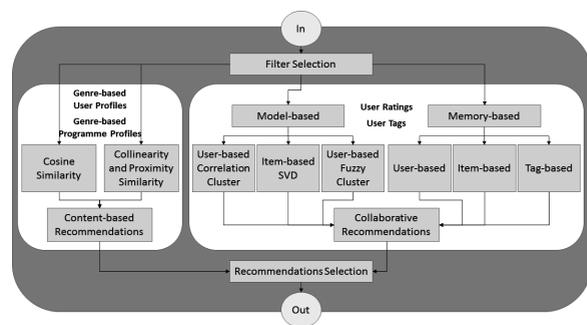

Fig. 3. Recommender service architecture.

5. Recommendation service

The designed recommendation service is a switch hybrid filter that comprises two content-based filters and six collaborative filters (three memory-based filters and three model-based filters). Each filter is an independent module, so multiple filters can run in parallel. This design choice is extensible and compatible with the eventual distribution and migration of the recommendation service into the cloud and, consequently, increases the scalability of the recommender as a whole. Furthermore, it builds user, passive user and item stereotypes to minimise the first-rater, passive user and cold-start problems.

Depending on the input data features, the recommender applies different filters: (i) if the input data are limited to genre-based user and programme profiles, the recommender selects content-based filters; (ii) if the input data are composed of ratings and tags, the recommender chooses collaborative filters. The recommendations generated are, then, evaluated by the inference rule system. In the case of off-line evaluation, the filter with the higher value of the $F - measure$ is selected and, consequently, is the one that provides the recommendations to the user. Figure 3 displays the architecture of the recommendation service.

This hybrid filter mechanism dynamically chooses and provides the best set of recommendations to each user.

Furthermore, it is extensible with new filters and rules and constitutes, as far as we know, a novel approach.

The recommender encompasses two content-based filters: (i) the cosine similarity, and (ii) the new collinearity and proximity similarity. In terms of collaborative filters, the recommender includes the following memory-based, model-based and cluster-based approaches: (i) user-based Pearson correlation (UbPC), (ii) item-based Pearson correlation (IbPC), (iii) tag-based (Tb) filter, (iv) item-based singular value decomposition (IbSVD), (v) user-based correlation cluster (UbCC), and (vi) user-based fuzzy cluster (UbFC).

5.1. Content-based filter. Equation (1) represents the Chebyshev distance dissimilarity (CDD), where $\hat{A}$ and $\hat{B}$ are the two normalised vectors of features and k represents the number of features:

$$CDD = MAX_k(|\hat{A}_k - \hat{B}_k|). \tag{1}$$

The most common similarity metric is the cosine similarity, which determines the collinearity between two arrays of features. For example, the recommendation system developed by Barragáns-Martínez *et al.* (2010a) uses the cosine similarity content-based filter to minimize the effect of cold-start recommendation in collaborative systems. Equation (2) represents the CS, where $\hat{A}$ and $\hat{B}$ are the two normalised vectors:

$$CS(\hat{A}, \hat{B}) = \frac{\hat{A} \cdot \hat{B}}{|\hat{A}||\hat{B}|} \equiv \frac{\sum_{j=1}^{n} \hat{A}_j \hat{B}_j}{\sqrt{\sum_{j=1}^{n} \hat{A}_j^2} \sqrt{\sum_{j=1}^{n} \hat{B}_j^2}}. \tag{2}$$

The current work proposes a new similarity metric called the collinearity and proximity similarity, which calculates and combines the CS with the CDD. While the cosine similarity establishes the collinearity between the two vectors of features, the Chebyshev distance determines the most disparate feature between the two vectors. Equation (3) presents the collinearity and proximity similarity metric formula, where CS is the cosine similarity, CDD is the Chebyshev distance dissimilarity and β is the combining parameter:

$$CPS = \beta \times \frac{\sum_{j=1}^{n} \hat{A}_j \hat{B}_j}{\sqrt{\sum_{j=1}^{n} \hat{A}_j^2} \sqrt{\sum_{j=1}^{n} \hat{B}_j^2}}$$
$$+ (1 - \beta) \times (1 - MAX_j(|\hat{A}_j - \hat{B}_j|)). \tag{3}$$

The goal of the CPS is to determine the similarity between two vectors, discriminating between identical, yet different vectors. According to Abouzakhar and Bello Abdulazeez (2009), the cosine similarity has a high number of false positive items and the Chebyshev distance presents a high number of false negative items.

There are several techniques for combining classifiers described by Tulyakov *et al.* (2008), such as

Borda count, majority voting, bagging, boosting and oracle (Vemulapalli *et al.*, 2009). Alternatively, there are approaches based on heuristics such as those described by Ramappa and Krishnamurthy (2013) that apply runtime classifier selection rules. In the case of the CPS, we use the combining parameter β. The adoption of the heuristic value $\beta = 0.5$ results from the simulations performed, and it allows taking the most advantage of both classifiers. For the different values of β, first the CPS *F-measure* was calculated for each user, and then the overall CPS *GF-measure*. Figure 4 shows the *GF-measure* as a function of β. This simulation was performed off-line with 80% of the historic data of each user being employed to build the genre-based user profile and 20% being used to validate the recommendation. Only recommendations regarding items actually seen by the user were considered (therefore the value of *Precision* is 1).
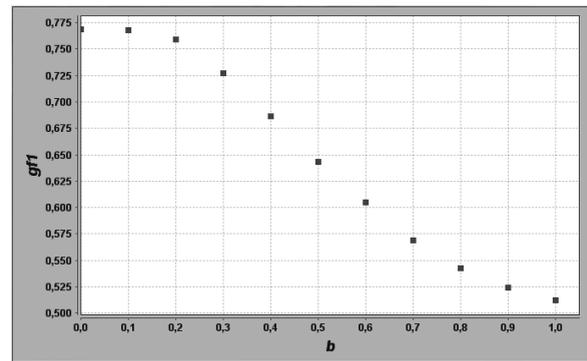

Fig. 4. CPS function and β parameter variation.

5.2. Tag-based collaborative filter. The adopted tag-based filter is an implementation of the work described by Rey-López *et al.* (2010) and Barragáns-Martínez *et al.* (2010b). This filter creates two tag clouds; the user tag cloud containing the tags assigned by the user, and the item tag cloud, containing the tags assigned to the item. Finally, a folksonomy is created that describes the relationship between the tags of the user and the item, by taking into account the number of direct and one-hop matching tags.

5.3. Stereotypes. The proposed recommendation service creates item and user stereotypes for new items, new users and socially passive users to minimise the first-rater, cold-start and passive use issues.

Item stereotypes. Stereotypes are generated for newly arrived items, i.e., items that have not been rated yet. These new item stereotypes are created with the help of clustering techniques. First, a cluster is produced for each

genre, and then a stereotype based on the average rating of the item genres is built for each new item.

Each cluster represents a genre and aggregates all available items of that same genre. For each cluster, the average genre rating is calculated, using Eqn. (4), where $\bar{r}_g$ is the average rating for each genre g and $r_{u,i}$ is the rating of user u for item i:

$$\bar{r}_g = \frac{1}{U} \sum_{i=1}^{I} \sum_{u=1}^{U} r_{u,i}. \tag{4}$$

When a new item arrives, it is represented by a 21-genre binary array. The new genre-based item stereotype is generated from the clusters of the genre(s) present in the item, using the average rating of each genre. Equation (5) describes this calculation, where $g_{i,j}$ is the value of the feature of the item i to genre j, $\bar{r}_j$ is the average rating of genre j and NG is the number genres of item i:

$$\bar{r}_i = \frac{1}{NG} \sum_{j=1}^{G} g_{i,j} \times \bar{r}_j. \tag{5}$$

To overcome the first-rater problem that afflicts collaborative filters, all users who rated items with the same genre(s) as the new item contributes to the initial rating of the new item stereotype. Once a user effectively classifies the new item, the classification assigned by the stereotype filter is removed.

The item stereotype will be used as an input to memory-based collaborative filters. This way, non-rated items can still be recommended. Furthermore, this approach supports the definition of personalised genre-based templates for electronic programme guides, and thus increases the scalability of the recommendation system by circumventing the calculation of predictions or the generation of user models.

User stereotypes. User stereotypes are created for new users entering the system and for socially passive users. The default user stereotype for newly arrived users can be generated from the most popular items in the platform or using the most popular item of each genre. The latter approach has the advantage of presenting the user with a genre-based menu. Once the viewer selects an item, the system immediately identifies the associated preferred genre(s) and updates the viewer profile accordingly.

The passive user tag cloud stereotype is created to allow the application of tag-based filters to users that watch but do not rate or tag the items. This stereotype is created from the set of tags attributed by other users to the items viewed by the social inactive user or passive user. The function $w : TC_i \to \mathbb{R}$ relates to each tag with its weight in the tag cloud, denoted by $w(t_k, TC_i)$.

$$w(t_k, TC_i) = \frac{f(t_k, TC_i)}{|C_i|}. \tag{6}$$

$|C_i|$ is the cardinality of the tag cloud, i.e., the result of adding the multiplicities of all of its tags:

$$|C_i| = \sum_{\forall k / t_k \in TC_i} f(t_k, TC_i). \tag{7}$$

The relationship presented in the Eqn. (8) takes into account the relative importance of the coincident tags in the tag cloud:

$$R(c_k, c_l) = \sum_{\forall i / t_i \in |TC_k \cap TC_l|} \sqrt{w(t_i, TC_k) w(t_i, TC_l)}. \tag{8}$$

5.4. User-based cluster FCM. This collaborative technique, which can be used to reduce the scalability problem, organises users into clusters, and then produces cluster-based recommendations. Fuzzy c-means (FCM) is a clustering methodology which assigns one user to several clusters. This methodology was developed by Dunn (1973) and improved by Bezdek in 1981 (Cannon *et al.*, 1986). The algorithm is based on the minimization of the objective function described in Eqn. (9), where m is the fuzziness exponent (a real number greater than 1), U is the number of users, C is the number of clusters, $dm_{u,j}$ is the degree of membership of x_u in the cluster j, x_u is the u-th element of d-dimensional measured data, c_j is the d-dimensional centre of the cluster, and finally $\| \cdot \|$ is any norm expressing the similarity between the measured data and the centre:

$$J_m = \sum_{u=1}^{U} \sum_{j=1}^{C} dm_{u,j}^m \parallel x_u - c_j \parallel^2, \tag{9}$$

$1 \le m < \infty$.

Once the user cluster membership degree $dm_{u,j}$ provided by Eqn. (10) and the cluster centres c_j defined by Eqn. (11) are updated, the fuzzy partitioning is carried out through an iterative optimization of the objective function:

$$dm_{u,j} = \left[\sum_{k=1}^{C} \left(\frac{\parallel x_i - c_j \parallel}{\parallel x_i - c_k \parallel} \right)^{\frac{2}{m-1}} \right]^{-1}, \tag{10}$$

$$c_j = \frac{\sum_{i=1}^{N} dm_{u,j}^m \times x_u}{\sum_{i=1}^{N} dm_{u,j}^m}, \tag{11}$$

In Eqns. (10) and (11) $\parallel x_i - c_j \parallel$ is the distance (e.g., Euclidean) from point i to the current cluster centre j and $\parallel x_i - c_k \parallel$ is the distance from point i to the remaining cluster centres k.

The iteration stops when the condition specified in Eqn. (12) is reached, where ε represents the termination criterion (a value between 0 and 1) and k is the number of iterations:

$$\max_{u,j} = \{\parallel dm_{u,j}^{k+1} - dm_{u,j}^k \parallel\} < \varepsilon. \tag{12}$$

The initial idea was to use the ratings of each user to create the fuzzy clusters. However, the technique of fuzzy c-means has problems when dealing with high dimensional data. This is caused by the calculation of the cluster centroids, which converge to the centre of gravity of the data set (Winkler *et al.*, 2012; 2011). This problem arises whenever the dimension of the ratings vector is high. To avoid this problem, our approach uses a 21-feature vector to represent the profile of each user.

After the determination of the clusters, it is necessary to calculate the ratings of the centroid of each cluster. This takes into account the user cluster membership degree as well as the user item ratings as described in Eqn. (13), where r_{ki} is the rating of user k for item i:

$$cr_{j,i} = \frac{1}{U} \sum_{k=1}^{U} r_{ki} \times dm_{k,j}. \tag{13}$$

The predictions are determined taking into account, for all clusters, the average user ratings as well as the user cluster membership degree. Equation (14) represents the generated predictions, where $\bar{r_u}$ is the average of the ratings of user u, $cr_{j,i}$ is the clustering rating of cluster j for item i and $dm_{u,j}$ is the degree of membership of user u in the cluster j:

$$P_{u,i} = \bar{r_u} + \sum_{j=1}^{C} cr_{j,i} \times dm_{u,j}. \tag{14}$$

5.5. Summary. This recommender service is supported by the creation of stereotypes for new items, new users, and passive users and by a rule-based switch hybrid filter. The rule-based system selects at runtime the most appropriate filter(s) to apply, taking into account the available data, and the set of recommendations to provide to the user. Although the CF solves the over-specialisation that afflicts CbF, it suffers from scalability and matrix sparsity. The standard approach to tackle scalability, which is based on off-line processing, may produce outdated recommendations. We use the folding-in technique to overcome this undesirable effect. To reduce the impact of the first-rater, cold-start and passive user problems, the service generates item stereotypes beforehand, user stereotypes and passive user tag cloud stereotypes. the proposed item stereotype and passive user tag cloud stereotype are, as far as we know, novel approaches.

6. Evaluation metrics

The metrics for evaluating recommender systems can be divided into predictive accuracy metrics and classification accuracy ones. Since the data set used contains scattered information, only items that were actually ranked by the

user are recommended. This way, the system increases the accuracy of the predictions but, since the suggested items belong to the group of items seen by the user, the calculated precision value is always one (Herlocker *et al.*, 2004).

6.1. Predictive accuracy metrics. This type of metrics measures the error between the predicted rating and the rating assigned by the user. According to Herlocker *et al.* (1999), there are two important metrics: the mean absolute error, which measures the average absolute deviation between the predicted rating and the rating assigned by the user, and the root mean square error, which emphasizes the largest errors. The MAE and $RMSE$ range corresponds to the ratings range. Equation (15) represents the MAE and Eqn. (16) the $RMSE$, where p_i is the predicted user rating for item i, r_i the user rating for item i and n corresponds to the number of items:

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |p_i - r_i|, \tag{15}$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (p_i - r_i)^2}. \tag{16}$$

The global MAE is used to evaluate the performance of the item stereotype approach. Equation (17) displays the $GMAE$, which is the weighted average of the MAE for all users of the data set, and where u represents the number of users:

$$GMAE = \frac{1}{u} \sum_{u=1}^{m} \left(\frac{1}{n} \sum_{i=1}^{n} |p_{i,u} - r_{i,u}| \right). \tag{17}$$

6.2. Classification accuracy metrics. These metrics measure the frequency with which the recommendation system provides correct recommendations. There are three important metrics: *Recall*, which determines the number of relevant items selected from the total number of relevant items available, *Precision*, which defines the number of relevant items selected from the total number of items, and, finally, the *F-measure*, which combines *Recall* and *Precision* into a single metric (Basu *et al.*, 1998). The *Precision*, *Recall* and *F-measure* metrics range between 1 (best) and 0 (worst). Equation (18) represents *Recall*, Eqn. (19) *Precision* and Eqn. (20) the *F-measure*, where TP is the number of relevant items recommended by the system, i.e., true positive results, FN is the number of relevant items not recommended by the system, i.e., the false negative results, and FP corresponds to the number of irrelevant items recommended by the system, i.e., the false positive results:

$$Recall = \frac{TP}{TP + FN}, \tag{18}$$

$$Precision = \frac{TP}{TP + FP}, \qquad (19)$$

$$F\text{-}measure = 2\frac{Precision \times Recall}{Precision + Recall}. \qquad (20)$$

The global F-*measure* was used to evaluate the performance of the different filters. Equation (21) displays the GF-*measure*, which is the weighted average of the F-*measure* for all users of the data set:

$$GF\text{-}measure = \frac{1}{n}\sum_{i=1}^{n}\left(2\frac{Precision \times Recall}{Precision + Recall}\right). \qquad (21)$$

7. Tests and results

The following subsections present the data set used, the global results obtained and an illustrative case study.

The recommender holds two content-based filters: (i) the cosine similarity, and (ii) the new collinearity and proximity similarity. In terms of collaborative filters, the recommender, which includes memory-based, model-based and cluster-based approaches, implements the following filters: (i) user-based Pearson correlation, (ii) item-based Pearson correlation, (iii) tag-based, (iv) item-based singular value decomposition, (v) user-based correlation cluster, and (vi) user-based fuzzy cluster.

7.1. Data set. The recommendation service was evaluated off-line with HetRec 2011 v. 2.0[1]. This data set was chosen due to lower data sparsity and data set size. It contains information about users and movies, including user ratings or user tags per movie together with timestamps. In our tests we only use items that were simultaneously rated and tagged. The data set was divided into a training set (80%) and a test set (20%), ensuring that, regardless of the number of ratings, each user has 80% of their ratings in the training set and the remaining 20% are in the testing set.

7.2. Global results. The off-line performance of the recommendation service was determined by testing each implemented filter separately. The training data set was submitted to each filter and the individual user's recommendations were generated. Then, these user recommendations were evaluated using the test data set for calculating the corresponding F-*measure*. The global F-*measure* (GF-*measure*) of a filter corresponds to the average of the F-*measure* values of the filter's recommendations. Table 1 displays the GF-*measure* and $GMAE$ results for the implemented filters, using a 60% similarity threshold and a 3.5 rating threshold.

To evaluate the off-line impact of using the novel passive tag cloud stereotype and the item stereotype

---

[1]http://www.grouplens.org/datasets/hetrec-2011/.

Table 1. GF-*measure* and $GMAE$ off-line filter results.

Stereotype	Filter	GF-*measure*	$GMAE$
	CS	0.512	–
	CPS	0.644	–
	Tb	0.558	–
PTCS	Tb	0.818	–
	UbPC	0.659	0.654
IS	UbPC	0.767	0.748
	IbPC	0.715	0.688
	IbSVD	0.724	0.699
	UbCC	0.709	0.784
	UbFC	0.609	0.783

approaches, we adopted a new concept: the *alter ego* and *alter item*.

In the case of the passive tag cloud stereotype, a twin user, called the *alter ego*, was created for all users with tags. This *alter ego* has seen the same films as the original user, but has provided no tags nor ratings. The user's passive cloud is produced based on the tags assigned by other users to the films watched by the passive *alter ego*. Finally, to validate the passive tag cloud stereotype approach, the GF-*measure* for the data set was calculated and compared with the original GF-*measure*: it showed an improvement of 0.260.

To evaluate the item stereotype approach, a twin item was created for all items. This *alter item* has the same genres as the original one, but has no tags nor ratings. Then, the item stereotype was created and the results were used as an input to the user-based Pearson correlation filter. The $GMAE$ results using the user-based Pearson correlation for the original item were 0.665 and for the *alter item* 0.748, i.e., it increased by 0.083. These results showed that, although the Item Stereotype generates less accurate predictions, they are not distant from the predictions generated for the original item rating.

7.3. Case study. Since the recommendation service is intended for near real-time recommendation, the time each filter takes to generate recommendations for a user was measured. These tests were conducted by selecting a user randomly and applying the different algorithms. Five measurements were taken and averaged for each filter with a cold-start, i.e., without any pre-processing. Tables 2 and 3 show the average results for User 57 obtained with a Dual Core 2.2 GHz CPU and 4 GB DDR2 RAM platform. Although it is impossible to foresee the actual time required to generate recommendations in near real time, the provided execution time is a good indicator of whether the filter is suitable for near real time operation. Nevertheless, no filter is dismissed by default since the hybrid filter can work in near real time, i.e., with a time frame of a few seconds, or off-line, i.e., when recommendations are calculated beforehand.

From the obtained results, we concluded that the IbPC and IbSVD are unsuited for near real time operations. The $F\text{-}measure$ ranges between 1 (best) and 0 (worst). The MAE and $RMSE$ range between 0 (best) and 5 (worst), which is the ratings' range. In the case of the passive tag cloud stereotype, the $F\text{-}measure$ value for the original User 57 was 0.530 and for the *alter ego* it was 0.832, i.e., the $F\text{-}measure$ improved by 0.302. In the case of the item stereotype, the predicted ratings for User 57 using the UbPC have an average of 3.193 and a MAE of 0.480, whereas the predicted ratings for User 57 using the UbPC generated from User 57 actual ratings have an average of 3.317 and an MAE of 0.397. These results for User 57 show that the combination of (i) PTCS and Tb improves the $F\text{-}measure$ significantly, allowing the application of tag-based collaborative filters to passive users; and (ii) IS and UbPC increases the $F\text{-}measure$ significantly, but generates slightly worse predictions than those based on the actual user ratings, making it a good technique to minimize the first-rater problem.

Table 2. Filter $F\text{-}measure$ and execution time for User 57.

Stereotype	Filter	$F\text{-}measure$	$\overline{\Delta t}$ (s)
	CS	0.377	002.022
	CPS	0.598	001.989
	Tb	0.530	001.390
PTCS	Tb	0.832	002.187
	UbPC	0.582	002.826
IS	UbPC	0.700	078.175
	IbPC	0.517	182.318
	IbSVD	0.741	537.089
	UbCC	0.783	009.526
	UbFC	0.590	046.280

Table 3. Collaborative filter MAE and $RMSE$ for User 57.

Stereotype	Filter	$\bar{r}$	MAE	$RMSE$
	UbPC	3.317	0.397	0.533
IS	UbPC	3.193	0.480	0.615
	IbPC	3.344	0.424	0.545
	IbSVD	3.484	0.418	0.522
	UbCC	3.708	0.529	0.677
	UbFC	3.497	0.477	0.610

After running the selected filters, the hybrid filter applies the output rule to choose the recommendations from the filter with the highest $F\text{-}measure$ value. Once the recommender selects the final set of user recommendations, the last step is the negotiation of the recommended items. The negotiation protocol used in this test was the FIPA Iterated Contract Net and the item price adaptation tactic used was of the linear incremental type. Table 4 shows the reference price ($RfPr$), incremental price ($IcPr$), maximum price ($MxPr$) and the negotiated

price ($NgPr$), per minute of transmission. Table 5 shows

Table 4. Negotiation for User 57 (values are expressed in EUR).

Item	$RfPr$ min	$IcPr$ min	$MxPr$ min	$NgPr$ min
The Usual Suspects	17	4	74	37
Star Wars: Episode V	24	4	72	44
Donnie Darko	20	5	70	45
Before Sunrise	16	5	73	41
Kill Bill: Vol. 2	15	4	79	35
Rear Window	22	3	82	37
The Incredibles	20	1	76	25
The Hunt for Red October	16	3	77	31
Rudy	22	3	74	37
The Lord of the Rings	18	2	74	28
Ghost World	18	5	72	43
Monty Python and the Holy Grail	24	5	75	49

the resulting EPG time line for User 57, which contains the set of successfully negotiated items.

Table 5. Personal EPG time line for User 57.

Start	Item
00:00	The Usual Suspects
01:46	Star Wars: Episode V
03:50	Donnie Darko
05:43	Before Sunrise
07:28	Kill Bill: Vol. 2
09:45	Rear Window
11:37	The Incredibles
13:32	The Hunt for Red October
15:46	Rudy
17:40	The Lord of the Rings
21:10	Ghost World
22:52	Monty Python and the Holy Grail

8. Conclusion

Achievements. The integration of the B2B brokerage with the B2C personalisation within the multimedia brokerage platform proved to be a promising solution for media businesses. It adopts a pay-per-view charging model, i.e., only the items that are actually watched by the viewers will be charged.

The platform models businesses as agents while keeping the strategic business knowledge private, i.e., only the data required for negotiation are sent to the platform. The use of standards at the business process level promotes the adoption of this platform by businesses.

The implemented switch hybrid recommender is a modular, extensible multi-filter system composed of

content-based and collaborative filters. For each user, it applies inference rules to choose from, depending on the available input data and the filters to apply, and to select the recommendation set to provide the viewer with the best option. The initial filter selection is advantageous since it chooses the best filter according to the available data. In the case of the off-line tests conducted, the recommender applies the $F\text{-}measure$-based off-line test rule to select the best set of recommendations.

The performed off-line tests inspired the specification of the inference rules that will govern the final near real time recommender. The results obtained have already excluded some filters for near real time operation due to their high execution time. The new content-based CPS filter performed better than the standard CS: the $GF\text{-}measure$ increased by 13.2% and the execution time remained unchanged. In the case of collaborative filters, since the HetRec data set has more movies than users, the user-based Pearson correlation is the fastest. If the data provided includes user ratings and the time is not pressing, the item-based SVD collaborative filter is the best. This filter, which takes significantly longer than the Pearson correlation, has the highest $GF\text{-}measure$ and lowest MAE. The tag-based filter performed worse than the other filters.

The adoption of model-based collaborative filters is a good choice to reduce the computation time. These filters, once the model is built, are fast and the results, in terms of $GF\text{-}measure$, are reasonable.

In terms of stereotypes, the new item stereotype minimizes the first-rater problem that afflicts collaborative filters. In the case of passive users, i.e., those who only watch items and do not assign tags, the new passive user tag cloud stereotype allows the application of tag-based collaborative filters. Furthermore, the results using this approach showed an improvement in the $GF\text{-}measure$.

The passive user tag cloud stereotype and item stereotype were evaluated off-line using a new concept: the generation of *alter ego* and *alter item* copies with the required properties. This evaluation technique proved to be a good solution, since it allowed the comparison between the recommendations produced using the actual user and item data, and the recommendations produced using the generated stereotypes. Without this technique, the evaluation of the recommendations obtained for the passive user tag cloud stereotype, or using the item stereotype approach, would only be possible with the help of a group of viewers.

Future work. Concerning the recommendation service, we plan to extend our research in tag-based filters, since social network tagging is popular and corresponds to explicit user preferences, and explore semantic enrichment.

In terms of B2C, the focus will be on the user profiling service. This distributor side service collects continuously implicit information,e.g., social networks, and explicit information, e.g., ratings, tags and the user's history, to build and update the profiles of users.

Finally, we intend to perform the evaluation of the recommendation system with real viewers, including different ages, interests and gender.

Acknowledgment

References

Abouzakhar, N. and Bello Abdulazeez, M. (2009). A fingerprint matching model using unsupervised learning approach, *3rd International Conference on Cybercrime Forensics Education & Training, Canterbury, UK*.

Al-Shamri, M.Y.H. (2014). Power coefficient as a similarity measure for memory-based collaborative recommender systems, *Expert Systems with Applications* **41**(13): 5680–5688.

Ardissono, L., Gena, C., Torasso, P., Bellifemine, F., Difino, A. and Negro, B. (2004). User modeling and recommendation techniques for personalized electronic program guides, *in* L. Ardissono, A. Kobsa and M.T. Maybury (Eds.), *Personalized Digital Television*, Springer, Dordrecht, pp. 3–26.

Bansal, N., Blum, A. and Chawla, S. (2004). Correlation clustering, *Machine Learning* **56**(1–3): 89–113.

Barragáns-Martínez, A.B., Costa-Montenegro, E., Burguillo, J.C., Rey-López, M., Mikic-Fonte, F.A. and Peleteiro, A. (2010a). A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition, *Information Sciences* **180**(22): 4290–4311.

Barragáns-Martínez, A.B., Rey-López, M., Costa Montenegro, E., Mikic-Fonte, F.A., Burguillo, J.C. and Peleteiro, A. (2010b). Exploiting social tagging in a web 2.0 recommender system, *IEEE Internet Computing* **14**(6): 23–30.

Basu, C., Hirsh, H.and Cohen, W. (1998). Recommendation as classification: Using social and content-based information in recommendation, *15th National/10th Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence, Madison, WI, USA*, pp. 714–720.

Burke, R. (2002). Hybrid recommender systems: Survey and experiments, *User Modeling and User-adapted Interaction* **12**(4): 331–370.

Cannon, R.L., Dave, J.V. and Bezdek, J. (1986). Efficient implementation of the fuzzy c-means clustering

algorithms, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **8**(2): 248–255.

Colomo-Palacios, R., González-Carrasco, I., López-Cuadrado, J.L. and García-Crespo, Á. (2012). ReSySTER: A hybrid recommender system for scrum team roles based on fuzzy and rough sets, *International Journal of Applied Mathematics and Computer Science* **22**(4): 801–816, DOI: 10.2478/v10006-012-0059-9.

Di Noia, T., Mirizzi, R., Ostuni, V. C., Romito, D. and Zanker, M. (2012). Linked open data to support content-based recommender systems, *Proceedings of the 8th International Conference on Semantic Systems, Graz, Austria*, pp. 1–8.

Dunn, J.C. (1973). A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters, *Journal of Cybernetics* **3**(3): 32–57.

Endert, H., Küster, T., Hirsch, B. and Albayrak, S. (2007). Mapping BPMN to agents: An analysis, *Agents, Web-Services, and Ontologies Integrated Methodologies, Durham, UK*, pp. 43–58.

Ghazanfar, M.A. and Prugel-Bennett, A. (2010). A scalable, accurate hybrid recommender system, *3rd International Conference on Knowledge Discovery and Data Mining, WKDD'10, Phuket, Thailand*, pp. 94–98.

Herlocker, J.L., Konstan, J.A., Borchers, A. and Riedl, J. (1999). An algorithmic framework for performing collaborative filtering, *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Berkeley, CA, USA*, pp. 230–237.

Herlocker, J.L., Konstan, J.A., Terveen, L.G. and Riedl, J.T. (2004). Evaluating collaborative filtering recommender systems, *ACM Transactions on Information Systems* **22**(1): 5–53.

Kurapati, K., Gutta, S., Schaffer, D., Martino, J. and Zimmerman, J. (2001). A multi-agent TV recommender, *Proceedings of the UM 2001 Workshop on Personalization in Future TV, Sonthofen, Germany*.

Lau, R.Y. (2007). Towards a web services and intelligent agents-based negotiation system for B2B ecommerce, *Electronic Commerce Research and Applications* **6**(3): 260–273.

Li, H., Cai, F. and Liao, Z. (2012). Content-based filtering recommendation algorithm using HMM, *Proceedings of the 4th International Conference on Computational and Information Sciences (ICCIS), Chongqing, China*, pp. 275–277.

Melville, P., Mooney, R.J. and Nagarajan, R. (2002). Content-boosted collaborative filtering for improved recommendations, *18th National Conference on Artificial Intelligence, Edmonton, Alberta, Canada*, pp. 187–192.

Melville, P. and Sindhwani, V. (2010). Recommender systems, *Encyclopedia of Machine Learning*, Springer, New York, NY, pp. 829–838.

Moreno, M.N., Segrera, S., López, V.F., Muñoz, M.D. and Sánchez, A.L. (2011). Mining semantic data for solving first-rater and cold-start problems in recommender systems, *Proceedings of the 15th Symposium on International Database Engineering & Applications, IDEAS '11, Lisbon, Portugal*, pp. 256–257.

Papagelis, M. and Plexousakis, D. (2005). Qualitative analysis of user-based and item-based prediction algorithms for recommendation agents, *Engineering Applications of Artificial Intelligence* **18**(7): 781–789.

Pera, M.S. and Ng, Y.-K. (2013). A group recommender for movies based on content similarity and popularity, *Information Processing & Management* **49**(3): 673–687.

Ramappa, M.H. and Krishnamurthy, S. (2013). A comparative study of different feature extraction and classification methods for recognition of handwritten Kannada numerals, *International Journal of Database Theory & Application* **6**(4): 71–90.

Rey-López, M., Díaz-Redondo, R.P., Fernández-Vilas, A. and Pazos-Arias, J.J. (2010). T-learning 2.0: A personalised hybrid approach based on ontologies and folksonomies, *in* F. Xhafa *et al.* (Eds.), *Computational Intelligence for Technology Enhanced Learning*, Berlin/Heidelberg, Springer, pp. 125–142.

Rosaci, D. and Sarnè, G. (2013). Multi-agent technology and ontologies to support personalization in B2C e-commerce, *Electronic Commerce Research and Applications* **13**(1): 13–23, DOI: 10.1016/j.elerap.2013.07.003.

Sarwar, B., Karypis, G., Konstan, J. and Riedl, J. (2002a). Incremental singular value decomposition algorithms for highly scalable recommender systems, *Proceedings of the 5th International Conference on Computer and Information Technology, Dhaka, Bangladesh*.

Sarwar, B.M., Karypis, G., Konstan, J. and Riedl, J. (2002b). Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering, *Proceedings of the 5th International Conference on Computer and Information Technology, Dhaka, Bangladesh*.

Shani, G., Meisles, A., Gleyzer, Y., Rokach, L. and Ben-Shimon, D. (2007). A stereotypes-based hybrid recommender system for media items, *Workshop on Intelligent Techniques for Web Personalization, Vancouver, Canada*, pp. 76–83.

Sollenborn, M. and Funk, P. (2002). Category-based filtering and user stereotype cases to reduce the latency problem in recommender systems, *in* S. Craw and A. Preece (Eds.), *Advances in Case-Based Reasoning*, Springer, Berlin/Heidelberg, pp. 395–405.

Tulyakov, S., Jaeger, S., Govindaraju, V. and Doermann, D. (2008). Review of classifier combination methods, *in* S. Marinai and H. Fujisawa (Eds.), *Machine Learning in Document Analysis and Recognition*, Springer, Berlin/Heidelberg, pp. 361–386.

Veloso, B., Sousa, L. and Malheiro, B. (2013). Personalised advertising supported by agents, *in* S. Omatu *et al.* (Eds.), *Distributed Computing and Artificial Intelligence*, Springer, Cham, pp. 473–481.

Vemulapalli, S., Luo, X., Pitrelli, J.F. and Zitouni, I. (2009). Classifier combination techniques applied to coreference resolution, *in* U. Germann *et al.* (Eds.), *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Student Research Workshop and Doctoral Consortium*, Association for Computational Linguistics, Stroudsburg, PA, pp. 1–6.

Vozalis, M.G. and Margaritis, K.G. (2005). Applying SVD on item-based filtering, *Proceedings of the 5th International Conference on Intelligent Systems Design and Applications, Wrocław, Poland*, pp. 464–469.

Winkler, R., Klawonn, F. and Kruse, R. (2011). Fuzzy c-means in high dimensional spaces, *International Journal of Fuzzy System Applications* **1**(1): 1–16.

Winkler, R., Klawonn, F. and Kruse, R. (2012). Problems of fuzzy c-means clustering and similar algorithms with high dimensional data sets, *in* W. Gaul *et al.* (Eds.), *Challenges at the Interface of Data Analysis, Computer Science, and Optimization*, Springer, Berlin/Heidelberg, pp. 79–87.

Wu, M.-L., Chang, C.-H. and Liu, R.-Z. (2014). Integrating content-based filtering with collaborative filtering using co-clustering with augmented matrices, *Expert Systems with Applications* **41**(6): 2754–2761.

Yanxiang, L., Deke, G., Fei, C. and Honghui, C. (2013). User-based clustering with top-n recommendation on cold-start problem, *Proceedings of the 3rd International Conference on Intelligent System Design and Engineering Applications (ISDEA), Hong Kong, China,* pp. 1585–1589.

Zhang, L., Tao, Q. and Teng, P. (2014). An improved collaborative filtering algorithm based on user interest, *Journal of Software* **9**(4): 999–1006.

Zhang, Y. and Jiao, J. R. (2007). An associative classification-based recommendation system for personalization in B2c e-commerce applications, *Expert Systems with Applications* **33**(2): 357–367.

Bruno Veloso holds an M.Sc. in electrical and computer engineering, specializing in telecommunications, and a three-year graduate degree in electrical and computer engineering from the Polytechnic Institute of Porto (School of Engineering). He is a researcher at INESC TEC (Centre of Robotics and Intelligent Systems). His interests include distributed artificial intelligence, multi-agent systems, personalisation and recommendation systems.

Benedita Malheiro holds a Ph.D. and an M.Sc. in electrical and computer engineering from the University of Porto (Faculty of Engineering). She is an adjunct professor at the Polytechnic Institute of Porto (School of Engineering) and a senior researcher at INESC TEC (Centre of Robotics and Intelligent Systems). Her research interests include multi-agent systems, conflict resolution, belief revision, personalisation, location-based and context-aware systems.

Juan Carlos Burguillo holds an M.Sc. in telecommunications and a Ph.D. in telematics from the University of Vigo, Spain. He is currently an associate professor at the Department of Telematic Engineering at the same university. He has managed several R&D projects, and has published more than 100 papers in international journals and conferences. His research interests include multi-agent systems, evolutionary algorithms, game theory and telematic services.