

Recommendation in the Digital TV Domain: an Architecture based on Textual Description Analysis

Felipe Ramos

Federal University of Campina Grande
Campina Grande, Brazil
feliperamos@copin.ufcg.edu.br

Alexandre Costa

Federal University of Campina Grande
Campina Grande, Brazil
antonioalexandre@copin.ufcg.edu.br

Reudismam Rolim

Federal University of Campina Grande
Campina Grande, Brazil
reudismam@copin.ufcg.edu.br

Gustavo Soares

Federal University of Campina Grande
Campina Grande, Brazil
gsoares@computacao.ufcg.edu.br

Hyggo Almeida

Federal University of Campina Grande
Campina Grande, Brazil
hyggo@dsc.ufcg.edu.br

Angelo Perkusich

Federal University of Campina Grande
Campina Grande, Brazil
perkusic@dee.ufcg.edu.br

Abstract—Recommendation systems have been used in several application domains, most recently for TV (Digital TV, Smart TV, etc.). Several approaches can be used to recommend items, tags, etc., mainly based on user feedback. However, in the Digital TV domain, user feedback has to be done generally by using the remote control, which should be avoided to improve user experience, since assigning explicit feedback to items is restricted by the characteristics of this domain (difficulties when typing with the remote control, etc.). Moreover, in the Smart TV environment several types of items can be recommended (movies, musics, books, etc.). Thus, the recommendation should be generic enough to suit to different content. To solve the problem of acquiring explicit feedback and still generate personalized recommendations to be used by different Smart TV applications, this work proposes a recommendation architecture based on the extraction and classification of terms by analyzing the textual descriptions of TV programs present on electronic programming guides. In order to validate the proposed solution, a prototype using a real dataset has been developed, showing that using the recommended terms it is possible to generate final recommendations for different Smart TV applications.

Keywords—*Digital TV, Term Classification, Term Recommendation.*

I. INTRODUCTION

Due to the significant growth in recent years of medias such as TV and Internet, the access to information has become increasingly easy. Therefore, a new range of services and information are available for users. However, given the large amount of content available, it is difficult for users to find relevant information [1]. In this context, recommendation systems (RSs) are presented as important tools, because they help users to select items and contents.

Recommendation systems work with the concept of items and users, where “item” is the general term used to denote what is recommended and “user” is the term used to represent who consumes the recommendation [11]. To perform personalized recommendations, in general, RSs need to identify the main

features of users or items (e.g., item descriptions, user histories, user ratings, etc.). These features are used to construct profiles, which are generated from the extraction of dataset information.

RSs can be used in different application domains to help users make better choices [11], [8], especially in Digital TV (DTV), where a large number of channels, programs and the content diversity complicate the user decision making [9]. In DTV, user profiles are usually generated by analyzing their TV viewing history, whereas item profiles are usually generated by extracting information from the electronic programming guide (EPG) [15], which provides program details, such as title, description, categories, etc. Besides the content diversity of Digital TV, with the advent of Smart TVs, which integrate features of the Internet and Web 2.0 on TV devices [3], the range of options for users has become even greater, since they can access information from multiple applications with different types of items. In addition of programs and channels, Smart TVs applications can recommend items of distinct types, such as movies, news, videos, musics, etc. Thus, given the heterogeneity of applications, the Smart TV domain demands an approach that ensures interoperability of recommendations, otherwise recommendation systems have to be developed for applications of different contexts, or at least different components of these RSs must be developed based on item types or features, such as the profile managers, where for each application an user would have a different profile.

Although some previous works focus on recommendation architectures applied to DTV / Smart TV [2], [3], [7], they normally deal with the recommendation of specific items (i.e., TV programs). However, many recommendation approaches can be applied with different features and goals, such as recommendation of terms, tags, etc.

In the recommendation of tags, for example, the model based on user feedback is well known [16], where the user assigns tags to several items, but this model is not suitable for the domain of DTV, since the action of assigning tags and giving explicit feedback in DTV is restricted by user

experience requirements, which demand the use of the remote control as less as possible [2], because watching TV and typing with it at the same time is a time consuming and difficult task. So, ways of acquiring implicit information in DTV need to be investigated, such as the extraction of terms from program descriptions present on the EPG.

Hence, DTV / Smart TV domain presents two important specifications that were not faced together in previous works, all the information must be implicitly collected and the recommendation must be generic enough to suit to different contexts. In our work, we propose a recommendation architecture based on the extraction and classification of terms from TV program descriptions. The main steps of the proposed solution are as follows:

- **Profiles generation:** despite the great number of different types of application users can interact, they consume the same type of item, TV programs. In our work, we build user profiles based on their TV viewing histories (i.e., programs they watched before), hence, the users have a unique profile independent of the applications they interact;
- **Term extraction and classification:** to represent the items (i.e., TV programs) we extract terms from program descriptions. In order to generate a generic recommendation, we classify extracted terms based on EPG categories, hence, it is possible to identify terms related to a given application, for example, terms of sports;
- **Recommendation:** instead of recommending programs only, we recommend classified terms based on program recommendation. Our main goal is to generate an intermediary recommendation, thus, it is possible to recommend different items, since the final recommendation for different applications can be processed from the term recommendation, avoiding the need to develop different RSs to different applications.

In order to validate the proposed solution, we developed a prototype using a real dataset, recommending two types of items, movies and books. The prototype consisted in the generation of two recommendation adapters, where the final recommendation was processed from the term recommendation. It showed that is possible to recommend for different Smart TV applications based on terms classified by EPG categories.

II. RELATED WORK

Similar to other areas, Digital TV suffers from information overload due to the growth in the number of TV programs and channels. Therefore, some studies are focused on this application domain [2], [3], [7], [12].

Chang et al. [3] proposed a TV program recommender framework for Smart TV, addressing several issues (such as accuracy, diversity, novelty, etc.), which contains three components: TV program content analysis module, user profile analysis module and user preference learning module.

Bambini et al. [2] described the integration of a recommendation system into FastWeb, a large IP Television

(IPTV) provider. The recommendation system implemented both collaborative and content-based techniques, in order to recommend programs and videos on demand.

Krauss et al. [7] proposed a system (TV Predictor) that includes recommendation mechanisms to Smart TVs, aiming to generate personalized program guides, which consist of personal channels for each user. Additionally, the TV Predictor Autopilot enables the TV set to automatically change the currently viewed channel, allowing the user to watch the personalized programming without further user input.

Unlike previously mentioned works, which intended to recommend specific items (i.e., TV programs), in this work the main goal is to propose an architecture to recommend terms that can be used by different Smart TV applications. The terms are extracted from textual descriptions of programs and classified based on program categories specified on the EPG. Thus, additionally to the program recommendation (component of the proposed architecture), which is processed by the analysis of user viewing histories, the recommendation of terms is also generated. Therefore, the proposed architecture aims to make the process of generating recommendations for applications of different contexts easier, and this is the main difference among this work and others mentioned before.

As a large part of content available is presented in textual format [13] (EPG, for example), some works focus on text categorization [13], [16].

Rossi et al. [13] proposed a textual document categorization algorithm to define a model inspired on a bipartite heterogeneous network. The network consists of two different types of objects: documents and terms extracted from their textual descriptions, in which the training set has some previously classified documents, and the induction consists of assigning weights to terms related to the known document classes. In our work, we adapt the proposed approach to perform the term classification phase, considering each program as a document and program categories as bipartite network classes, as illustrated in Figure 1. Each program on the EPG has its predefined categories, so the adaptation aimed only to identify the relationship between terms extracted from program descriptions and categories.

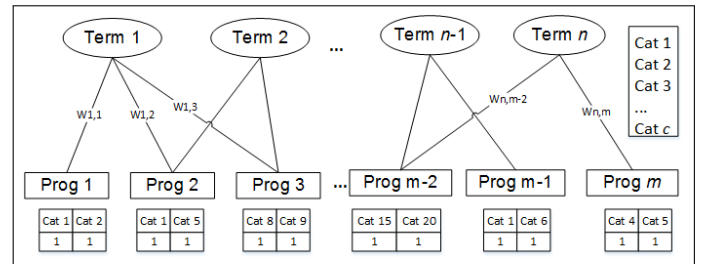


Fig. 1: Bipartite heterogeneous network.

III. PROPOSED SOLUTION

In this section, we present the proposed recommendation architecture, that aims to generate term recommendations by following Digital TV domain specifications to maximize the user experience while watching TV, which are as follows:

- **Different applications:** several kinds of Smart TV applications can be added to the Digital TV domain, featuring different types of items such as books, movies, news, etc. Thus, the generated recommendation must be intermediary, i.e., it must be possible to generate a final recommendation from it, ensuring interoperability;
- **Remote control as interaction source:** user interacts with TV through the remote control. Therefore, to avoid the use of this device and maximize user experience while watching TV, information must be collected by implicit feedback;
- **Implicit feedback:** Digital TV implicit feedback calculation is different from other domains (e-commerce, etc.), since there is a period that a user can consume an item. For example, if a program is presented once a week, a given user can only watch it on the exact day and moment of its transmission. Thus, we calculate a user implicit feedback by dividing the number of times a program was watched by the number of times it was presented;
- **EPG with predefined categories:** unlike other domains, items in Digital TV (TV programs) generally present two categories specified on the EPG. Therefore, with program descriptions and their respective categories it is possible to identify the relationship between terms extracted from descriptions and categories (action, sports, news, etc.). So, it is possible to recommend terms of a specific category related to an application, i.e., to personalize term recommendation by categories (action terms, news terms, sport terms, etc.).

An overview of the proposed architecture is shown in Figure 2, with the following components:

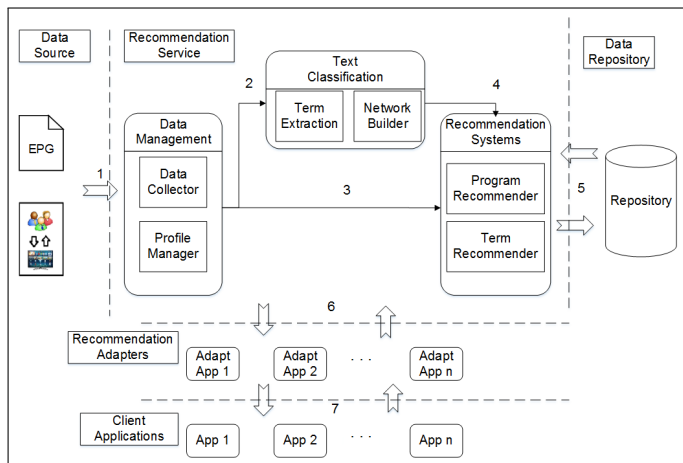


Fig. 2: General architecture.

- **Data Management:** collect (1) and manage information to generate user and item profiles (3 and 2);
- **Text Classification:** classify terms extracted from textual descriptions present on the EPG based on program categories;

- **Recommendation Systems:** generate program and term recommendations based on user and item profiles (3 and 2) and classified terms (4);
- **Recommendation Adapters:** generate final recommendation (7) from the term recommendation (6);
- **Repository:** store and retrieve (5) information from user and item profiles.

A. Data Collection and Management

The data collection of TV programs and users is performed by the Data Collector (Figure 2), which is a common element of recommendation architectures [2]. In the proposed architecture, program information is extracted from the EPG, while user information is collected implicitly by the analysis of their TV viewing history.

The Profile Manager (Figure 2) is responsible for managing the information collected from the data sources (1) and aims to create user and item profiles that are stored in the repository (5) and, subsequently, used during classification and recommendation phases.

Program profiles are formed by their textual features extracted from the EPG (e.g., title, description, categories, etc.). On the other hand, to create user profiles we calculate the implicit feedback by counting the number of times a user u viewed a particular program p and how often this program is weekly presented, and apply Equation 1.

$$\bar{r}_{up} = \left\lfloor \frac{v_{up}}{f_p} \right\rfloor \times 5, f_p > 0, \quad (1)$$

where v_{up} is the number of times the user u watched the program p and f_p represents the number of times the program p is weekly presented. The obtained rating is a normalized value from 1 to 5. An example of applying Equation 1 is shown in Table I, for example, user 1 watched program 1 three times in a week, and the program is presented three times in the same period. Thus, the user implicit feedback is 5.

TABLE I: Implicit feedback examples

User Id	Prog Id	Prog freq	week	User watch freq	Impl rating
1	1	3	3	5	5
1	2	1	1	5	5
2	1	3	1	2	2
⋮	⋮	⋮	⋮	⋮	⋮
2	3	7	5	4	4

User profile information (3) is used by the Recommendation Systems component (Figure 2) to generate recommendations. On the other hand, item profile information (2 and 3) is used by the Recommendation Systems and by the Text Classification component (Figure 2), which processes term extraction and classification.

B. Term Extraction and Classification

This step consists of extracting terms from TV program descriptions (performed by the Term Extraction component (Figure 2)) and classifying them based on program categories (performed by the Network Builder component (Figure 2)).

The term extraction is performed by mining TV program textual descriptions to retrieve representative terms. To accomplish this task, we first discard *stop words* (less significant words such as prepositions, articles, etc.) [4]. Then, we perform *stemming* (reduce words to roots) [4]. At the end, each program will have a vector of terms where each position in this vector corresponds to the frequency of the term on the program textual description.

The term classification consists of categorizing previously extracted terms into EPG categories. Since each program on the EPG has usually two predefined categories, it is possible to identify the relation between term and category through their co-occurrence. At the end of this phase, each term will have a weight assigned to the categories.

To perform the term classification, we construct a bipartite heterogeneous network, which consists of a network $G = (V, E, W)$ with different types of objects V , a set of connections between objects E — no link between objects of the same type is needed — and a set of connection weights W [13].

Our bipartite network is an adaptation of Rossi et al. [13] proposed one, including two types of objects: terms and programs. The term weight set is given by the matrix $W = \{w_1^T \dots w_\alpha^T\}^T$, where α is the number of terms extracted from program descriptions and w_{ij} is the weight of the term i to the category j . The matrix W has dimension $\alpha \times \phi$ where ϕ is the number of categories. The TV program categories are represented by the vector $c = \{c_1, \dots, c_{|C|}\}$. The terms extracted from program descriptions are represented by the vector $f = \{f_1, \dots, f_\alpha\}$. Each object of the program type has a weight vector for the categories, which is represented by the matrix $Y = \{y_1^T, \dots, y_\theta^T\}^T$, where θ is the number of programs available, and y_{kj} receives the value 1 if the program k has the category j , 0 otherwise. The weight of the relation between programs and terms is given by the matrix $D = \{d_1^T, \dots, d_\theta^T\}^T$, each position d_{ki} represents the frequency of a term i in the description of a program k . The matrix D has dimension $\theta \times \alpha$.

The goal of this classification step is to construct the matrix W . To accomplish this task, we use the IMBHN algorithm [13], which allows inferring the influences of each term for program categories. The IMBHN algorithm performs the process by minimizing the cost function given by Equation 2 [13]:

$$\begin{aligned} Q(W) &= \frac{1}{2} \left(\sum_{j=1}^w \sum_{k=1}^{\theta} (\text{class}(\sum_{i=1}^{\alpha} d_{ki} w_{ij}) - y_{kj})^2 \right) \\ &= \frac{1}{2} \left(\sum_{j=1}^w \sum_{k=1}^{\theta} \text{error}_{kj}^2 \right), \end{aligned} \quad (2)$$

where,

$$\text{class}(\sum_{i=1}^{\alpha} d_{ki} w_{ij}) = \begin{cases} 1 & c_j = \underset{c_{j^*} \in c}{\text{argmax}} (\sum_{i=1}^{\alpha} d_{ki} w_{ij^*}) \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

The algorithm aims to minimize the squared error between the predicted and real values of the program categories. Gradient descent (Least-Mean-Square [13]) adjusts matrix W until a minimum error or a maximum number of iterations is reached (i.e., algorithm stop conditions¹).

C. Term Recommendation

Before generating the term recommendation, we analyze user and item profiles to process program recommendation, in order to identify appropriate items for users, this task is performed by the Program Recommender component (Figure 2).

In this work, we generate lists of recommended programs for users by using a hybrid recommendation system, which is generally a combination of collaborative filtering and content-based approaches, however, any recommendation techniques can be applied. For collaborative filtering we used Matrix Factorization² (MF), in which learning is performed by stochastic gradient descent [6]. Additionally, we analyze user profiles and include into MF recommended lists, programs with ratings greater or equal to 3 in user histories.

Finally, we generate term recommendation through the Term Recommender component (Figure 2). As each recommended program has a vector of terms extracted from its descriptions (Term Extraction (Figure 2)) and classified by EPG categories (Network Builder (Figure 2)), the weight of a recommended term t for a user u is given by:

$$\bar{r}_{tu} = \frac{1}{|P_u|} \sum_{p=1}^{|P_u|} f_{tp} \times r_{pu}, |P_u| > 0, \quad (4)$$

where $|P_u|$ is the number of recommended programs containing the term t to the user u , and f_{tp} is the number of occurrences of the term t on program p description and r_{pu} is the recommended rating to the program p for the user u . Thus, terms more frequent in recommended program descriptions tend to receive a higher weight. Using this weight, an application can use recommended terms according to its own requirements.

D. Recommendation Adapters

Recommendation Adapters (Figure 2) are responsible for generating the final recommendation (7), which is used by the Client Applications (Figure 2). Thus, for each application a corresponding adapter must be created, which uses (6) the Recommendation Service (Figure 4.2).

¹maximum number of iterations = 1000, minimum error = 0.01.

²We use MyMediaLite Recommender System Library - <http://www.mymedialite.net/>.

We create a recommendation adapter based on categories related to the corresponding application and its list of users (7), which are reported to the recommendation service (6), and then the lists of recommended terms to users are returned (6), such as, terms of action, comedy, sports, etc.

A possible technique to generate final recommendation is to compute the similarity between vectors of recommended terms and vectors of extracted terms from textual information of recommendable items (e.g., movies, books, news, etc.). Here, we use the cosine similarity [14], which compute the angle between two vectors (common terms) x and y of size m .

An illustration of the final recommendation generation is shown in Figure 3, where we want to select the most suitable drama movie to the user interest among three possible choices (i.e., Movie 1,2,3). The first step is to obtain the user recommended terms of drama (i.e., user feature vector for the drama category), then, we extract terms from movie descriptions, in order to create each movie vector of terms (i.e., movie feature vectors). Finally, we calculated the cosine similarity among the user feature vector and each movie feature vector. Thus, based on the calculated similarity, Movie 3 is the best related to the user interest, since it achieved the greatest similarity, which means that Movie 3 feature vector and the user feature vector present a greater number of common terms.

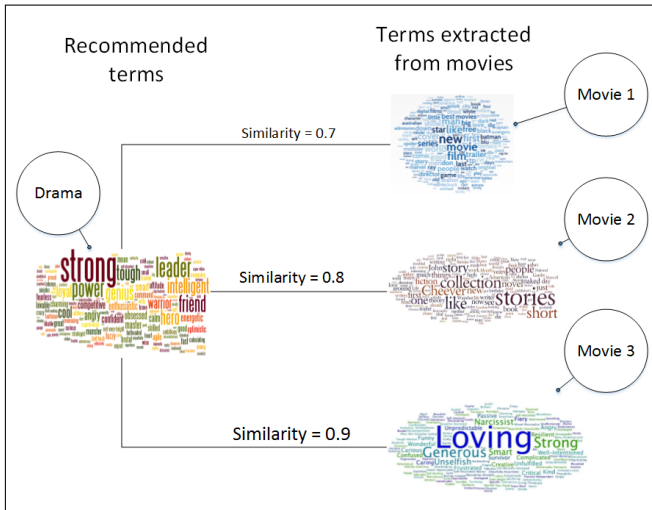


Fig. 3: Example of final recommendation generation.

IV. VALIDATION

In order to validate our solution we used a real dataset, which consists of information about TV users and their viewing histories. To collect the data we conduct a survey, where the participants filled a form³ with information about the programs they usually watch and their corresponding frequency view during a week. The dataset is composed of 63 users, 112 programs and 29 types of program categories.

The main goal of the validation is to proof the interoperability of recommendations, i.e., recommend different items from the term recommendation. Thus, we developed a prototype, which consists in creating two recommendation adapters

to recommend for the 63 users two different items, books and movies. The proposed architecture should be integrated into a DTV architecture. However, for matters of validation the process took place in a desktop environment.

To generate the final recommendation (books and movies), we create two recommendation adapters, one for each type of item, where the following steps were performed:

- 1) We specified EPG categories related to the book and movie applications (e.g., action, adventure, comedy, drama, police and thriller);
- 2) We determined the list of users of each application, here we considered the 63 participants of the survey;
- 3) Then, we processed term recommendation by category, i.e., for each user we got recommended term of action, adventure, comedy, etc.;
- 4) After obtaining the recommended terms, we generated the final recommendation of books and movies. The process was performed as follows: first, we mined item descriptions to extract representative vectors of terms. Then, we calculated the similarity (i.e., cosine similarity) between recommended terms by category for each user and extracted term vectors, recommending items with greater similarity.

The proposed approach can be used to recommend different types of items (such as videos, news, products, etc.), since they have textual description and related categories. Therefore, the same way the movie and book recommendations were processed it could be done for other types of items.

In order to evaluate our solution, we compare it with a non-personalized approach (i.e., without classification and recommendation), in which the user feature vector consists of the occurrence of terms extracted from user histories, i.e., the weight assigned to each term here is given by its number of occurrences in program descriptions. The main research question we want to answer is the following:

P1 - The use of our personalized approach (PA) improves the precision of the final recommendation compared to the non-personalized one (NP)?

To answer that, we formulate the following hypotheses:

H_0 : The precision of PA is greater then NP for category c and item i .

Where c is a category of type action, adventure, comedy, drama, police or thriller, and i is an item of type movie or book. So that, we have 12 hypotheses.

Since the assumption of normality was not met based on Shapiro-Wilk test, we used the non-parametric test of Wilcoxon signed-rank (95% confidence, i.e., $\alpha = 0.05$).

In Table II can be seen the results of movie recommendation precisions, i.e., the mean of the precisions ($P@5^4$) calculated for each user, and the p-value related to the hypothesis tests, where we conclude that our solution outperformed the NP approach for the categories action, police and thriller. For the remainder, the two approaches are statistically equal.

³<http://goo.gl/kEDKWt>

⁴Given a category, how many of the top 5 recommended movies are of that same category?

TABLE II: Movie recommendation precisions

	Categories					
	Act.	Adv.	Com.	Dra.	Pol.	Thr.
PA	66%	44%	34%	23%	44%	38%
NP	50%	45%	34%	26%	20%	30%
p-value	1e-06	0.44	0.22	0.82	4e-11	7e-06

In Table III can be seen the results of book recommendation precisions, i.e., the mean of the precisions ($P@4^5$) calculated for each user, and the p-value related to the hypothesis tests, where we conclude that our solution outperformed the NP approach for all studied categories, except for drama.

TABLE III: Book recommendation precisions

	Categories					
	Act.	Adv.	Com.	Dra.	Pol.	Thr.
PA	47%	33%	20%	7%	44%	74%
NP	42%	25%	14%	29%	23%	38%
p-value	0.04	0.001	0.01	1	1e-09	8e-11

Finally, we can conclude that the use of our solution of extraction, classification and recommendation of terms achieved better results for both, movie and book recommendations.

V. CONCLUSION AND FUTURE WORK

In this paper, we presented a recommendation architecture based on the extraction and classification of terms from TV program descriptions, applied to Digital TV domain. The proposed approach for the term extraction is important to overcome problems arising from the interaction constraints between users and TV. Another significant contribution is the term classification and recommendation approach, which allows different applications to use the generated recommendations, which is not possible in approaches that recommend items.

The prototype showed the feasibility of the proposed solution, ensuring that is possible to recommend different items using the term recommendation approach. Thus, to generate the final recommendation to a given application, it is only necessary to create a corresponding recommendation adapter, which uses the term recommendation based on the categories from the EPG closely related to the application. So that, only the recommended terms that have higher weight for these categories are returned.

As future work, a study about different ways to extract terms will be carried out. Some works have focused on the extraction of product attributes [5], [17]. The proposed approaches in these studies can be evaluated and integrated to the current proposal with the aim to obtain a better representation of TV programs, and hence, to get more significant recommendations.

As the solution proposed in this paper is based on the extraction of EPG content, which in some cases may contain reduced information [10], a possible future work is to investigate ways to obtain data from different sources to enrich the EPG information, and improve the textual representation of TV programs, for example, using information from Wikipedia⁶ [10].

⁵Given a category, how many of the top 4 recommended books are of that same category?

⁶<http://www.wikipedia.org/>

ACKNOWLEDGMENT

The authors would like to thank CAPES for support this work.

REFERENCES

- [1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *TKDE, IEEE*, 17(6):734–749, 2005.
- [2] R. Bambini, P. Cremonesi, and R. Turrin. A recommender system for an iptv service provider: a real large-scale production environment. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, *Recommender Systems Handbook*, pages 299–331. Springer, 2011.
- [3] N. Chang, M. Irvan, and T. Terano. A tv program recommender framework. *Procedia Computer Science*, 22(0):561 – 570, 2013. KES2013.
- [4] A. Fariña, N. R. Brisaboa, G. Navarro, F. Claude, A. S. Places, and E. Rodríguez. Word-based self-indexes for natural language text. *ACM Trans. Inf. Syst.*, 30(1):1:1–1:34, Mar. 2012.
- [5] R. Ghani, K. Probst, Y. Liu, M. Krema, and A. Fano. Text mining for product attribute extraction. *SIGKDD Explor. Newsl.*, 8(1):41–48, June 2006.
- [6] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, Aug. 2009.
- [7] C. Krauss, L. George, and S. Arbanowski. Tv predictor: Personalized program recommendations to be displayed on smarttvs. In *Proceedings of the 2Nd International Workshop on BigMine, BigMine '13*, pages 63–70, New York, NY, USA, 2013. ACM.
- [8] G. Linden, B. Smith, and J. York. Amazon.com recommendations: item-to-item collaborative filtering. *Internet Computing, IEEE*, 7(1):76–80, 2003.
- [9] A. Martinez, J. Pazos Arias, A. Vilas, J. Duque, and M. Nores. What's on tv tonight? an efficient and effective personalized recommender system of tv programs. *Consumer Electronics, IEEE Transactions on*, 55(1):286–294, 2009.
- [10] C. Musto, F. Narducci, P. Lops, G. Semeraro, M. Gemmis, M. Barbieri, J. Korst, V. Pronk, and R. Clout. Enhanced semantic tv-show representation for personalized electronic program guides. In J. Masthoff, B. Mobasher, M. Desmarais, and R. Nkambou, editors, *User Modeling, Adaptation, and Personalization*, volume 7379 of *Lecture Notes in Computer Science*, pages 188–199. Springer Berlin Heidelberg, 2012.
- [11] F. Ricci, L. Rokach, and B. Shapira. Introduction to recommender systems handbook. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, *Recommender Systems Handbook*, pages 1–35. Springer, 2011.
- [12] R. Rolim, F. Barbosa, A. Costa, G. Calheiros, H. Almeida, A. Perkusich, and A. Martins. A recommendation approach for digital tv systems based on multimodal features. In *Proceedings of the 29th Annual ACM SAC, SAC '14*, pages 289–291, New York, NY, USA, 2014. ACM.
- [13] R. Rossi, T. de Paulo Faleiros, A. de Andrade Lopes, and S. Rezende. Inductive model generation for text categorization using a bipartite heterogeneous network. In *12th ICDM, 2012*, pages 1086–1091, 2012.
- [14] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th WWW, WWW '01*, pages 285–295, New York, NY, USA, 2001. ACM.
- [15] X. Shi and J. Hua. An adaptive preference learning method for future personalized tv. In *KIMAS'05, 2005. International Conference on*, pages 260–264, 2005.
- [16] Y. Song, L. Zhang, and C. L. Giles. Automatic tag recommendation algorithms for social recommender systems. *ACM Trans. Web*, 5(1):4:1–4:31, Feb. 2011.
- [17] T.-L. Wong, W. Lam, and T.-S. Wong. An unsupervised framework for extracting and normalizing product attributes from multiple web sites. In *31st annual international ACM SIGIR, SIGIR '08*, pages 35–42, New York, NY, USA, 2008. ACM.