Towards TV Recommender System: Experiments with User Modeling

Milan Bjelica, *Member*, IEEE

Abstract — *As the number of cable TV programs grows, it becomes more difficult for the viewers to find the right one. This calls for specialized recommender systems, often in a form of electronic program guides, which should provide unobtrusive assistance. In this paper, we analyze such recommender system design under the broadcast scenario, where uplink connection to the network center is not available. We put special emphasis on user modeling algorithm that would be able to efficiently learn the user's interests. Our proposal applies the elements of machine learning and pattern recognition, as well as the information retrieval theory, like vector spaces and cluster hypothesis. The derived algorithm is computationally simple, while experimental results show high acceptance ratio of the proposed recommendations.*

Index Terms — Digital TV, personalization, program recommendation, user modeling.

I. Introduction

To the majority of users, the most important feature of digital television is probably a large number of available programs. Indeed, cable operators often advertise their services by stating the number of programs they offer. Surprisingly, the users do not necessarily benefit from this overabundance of available content; instead, they repeatedly find it difficult to retrieve the right program at a given time and consequently watch only few programs from the available myriad on a regular basis. Clearly, this outcome satisfies neither service/content providers nor the users.

Until the fully personalized TV, where each user would be able to create program on her own becomes available, the researchers and providers focus their attention to the methods of assisting the TV viewers in finding the programs they might like. To help their users, many cable operators nowadays offer electronic program guides (EPGs). These, however, usually only list all the available programs and therefore represent another example of "one size fits all" approach that does not solve the problem. What the users would really need is some sort of personalized program guide which would recommend them only those programs that they are likely to be interested in.

In this paper we discuss how to provide the right recommendations in the digital TV environment. We restrict our considerations to the currently dominant broadcast application scenario. Here, a TV receiver can access only the downstream data from the center and the return (uplink) channel is unavailable. We discuss in detail the question of proper user modeling that will enable us to learn the patterns of user's preferences. We show that by applying some results from the field of information retrieval, it is possible to build a simple yet robust system that could provide helpful recommendations to its users.

The rest of this paper is organized as follows. Section II gives the overview of the related work. Section III discusses some specific features of the recommender systems for use in digital TV. Section IV describes our proposal which is then tested in Section V. Experimental results are discussed in Section VI, while Section VII concludes the paper.

II. Background and Related Work

Recommender systems are adaptive systems that deliver suggestions to their users about the content that matches their estimated interests. These systems are designed to operate in the environments where the amount of available content overcomes the user's capability to browse it in time-efficient manner.

A comprehensive overview of recommender systems is given in [1]. Two major strategies are content-based and collaborative recommendations. *Content-based systems* recommend those items that resemble the ones the user liked in the past, while *collaborative systems* recommend the items that the other users with similar tastes liked in the past. Both types of systems have good and bad sides. The most significant limitation that applies to both is the cold start or the new user problem, which reflects the fact that the system can hardly recommend an item to the user who has not generated enough interactions for her preferences to be learned. This problem could be solved by asking each new user to supply the system with the background data that could be used to derive the initial recommendation. This is known as explicit user modeling and because of its inherent intrusiveness should be avoided whenever possible. Amongst other limitations, the content-based systems tend to recommend only those items that are very similar to the recommendations the user has previously accepted, which is known as overspecialization; collaborative systems have problems with the users whose preferences fall outside the existing niches ("gray sheep") and in the environments with extremely large numbers of users and available items (sparsity). *Hybrid recommender systems* which are discussed in details in [2] try to combine the good sides from both content-based and collaborative techniques, while avoiding their limitations.

M. Bjelica is with the Communications Department, School of Electrical Engineering (ETF), Belgrade, Serbia (e-mail: milan@etf.rs).
Contributed Paper
Manuscript received 07/06/10
Current version published 09/23/10
Electronic version published 09/30/10.

Several user modeling techniques for use in specialized TV recommender systems are described in [3]. The authors state that although collaborative systems suite web-based applications in an excellent way, the personalized EPGs should rely on techniques that could be applied locally in the user's TV receiver. Clearly, only content-based methods meet this demand.

The proposal found in [4] has many common points with the theory of information retrieval. TV program is described with a genre specifying vector, which resembles the feature vector from information retrieval. User's preferences are represented by another vector which describes the user's interest in certain genres. The initial value of this vector is determined from the stereotypical model for Japanese TV audience and is later updated according to the user's interactions. The appeal of certain TV program to the observed user is computed as dot product of the corresponding program and user vectors.

Vector space model is also used in [5]. The authors apply cosine similarity function to computing the appeal of a program to a user. Program features are extracted from the metadata information, while program description vector is separately derived for each user. User model is formed by explicit, implicit, or hybrid profiling. Explicit profiler relies on "Display" and "Remove" buttons in the user interface; by using them, the user can state her opinion on the current recommendation. Implicit profiler observes the fraction of the program duration time the user has actually watched it.

Two more recent papers fall outside the scope of our consideration scenario. AVATAR [6] is a TV recommender system designed for an environment in which the receiver (or set-top-box) is permanently connected to a broadband network. This assumption matches the technologies like Internet TV. In [7], a web application that provides personalized hybrid TV recommendations is described.

The systems discussed so far were designed to provide the assistance only to individual users. However, a TV receiver is usually shared by multiple users, like members of a household, and the program is sometimes viewed by a group (family or friends). System described in [8] is intended to provide recommendations to the group of users. Group profile is formed by combining the individual user's profiles, but the authors miss to explain how these are initially obtained.

III. Personalized Recommender Systems for Digital TV

General purpose recommender systems are often derived under some idealized assumptions, such as cooperative and patient users, well-defined and distinctive items, or specialized hardware. These are seldom met in real world and do not apply to the broadcast TV scenario that we consider. In this section, we discuss in more details some questions related to the TV recommender system design.

A. Hardware

The hardware needed to support the underlying algorithms on the user side must be compatible with the existing equipment (i.e. TV receiver or set-top-box) in both size and user interface.

Ideally, the additional hardware would be small enough to fit into the TV or set-top-box; in our opinion, this could be easily accomplished by careful PCB design.

The demand of the user interface needed to gain knowledge on user's interests and display the recommendations is far more serious. This interface must not obtrusively interfere with the usual way TV is watched. Since even in the near future the features like computer (alphanumeric) keyboard, voice command, or touch screen are not likely to become widely available for home TV receivers, the user interface will have to use the existing functionalities, i.e. receiver or set-top-box remote control as input and receiver display as output device.

The demand of unobtrusiveness and the limited features of user interface jointly influence both the way a dialogue between the user and the system is performed and also the way the system learns the patterns of user's preferences. As an example, one cannot expect the user to enter her detailed personal data using a standard TV remote; what can be performed instead is a series of short yes/no or multiple choice questions, organized in menu-like structure.

B. Return Channel

Return channel that would enable data transmission from the receiver to the center is not available in the traditional broadcast technologies[1]. As a consequence, locally gathered personal information cannot be sent to the content provider and all the recommendation-related activities (like user modeling and program retrieval) are to be performed only within the user's TV receiver.

Let us note that recommender design in the absence of return channel is also discussed in [9], but with the initial assumptions different from our scenario.

C. Recommendation Techniques

By elaborating the previous subsection further, we can state that *content-based techniques should be dominant for digital TV recommender systems*. However, this does not mean that we exclude the possibility of collaborative recommendations as these could be carried out within single receiver unit, i.e. for the members of the same viewing group. In that case, the users could log in to the system by entering their personal codes on the remote.

There are certainly many users who might be primarily interested in watching only few types of content. For example, after returning home from a day in the office, they might want to relax and watch a movie. In this context, an overspecialization of content-based recommenders might not

[1] Technologies like DVB-H or IPTV cannot be regarded as traditional and therefore are outside of the scope of this paper.

be undesirable, but a valuable feature instead. To provide certain degree of diversity, system output could consist of few (e.g. not more than three) programs of expected interest within a given category.

Some authors solve the problem of cold start by using stereotype user profiles. What remains questionable is if the systems that rely on the user-behavior data gathered within certain geographic territory would perform equally well in other countries. We are not convinced in this and therefore avoid the use of stereotypes. We also argue that it might be better to leave the system gradually learn the user's habits and then start providing the recommendations, based on the learned patterns.

D. Program Features

Content-based techniques ask that the descriptive features are explicitly associated with the available items [1]. To ease their further processing, it is desirable that the features are expressed in numeric form. Quantified features could be ordered in a vector structure that would then serve as a unique descriptor for the observed item; this is the concept of the so-called item surrogate and the vector spaces from the information retrieval and pattern recognition.

Although the program features might be abstract in their nature, it is convenient that they relate to some practical aspects of a TV program. Assignment of features to the multimedia items by humans might be biased [10], and the automatic feature extraction is still under development. What we propose is to obtain a set of TV program features by use of the metadata information from MPEG-7 standard or extended text tables (ETTs) from MPEG-2, which are part of digital TV streams.

While some metadata, like program air time or duration are already numerical, the others, like title, genre, and cast are textual and need to be converted to numerical equivalents. The solution is once again easily found in the information retrieval: It is possible to define a set of relevant features that would apply to all available items, e.g. 256 TV program genres, as in [4]. This leads to the high-dimensional vector spaces, where each coordinate (i.e. basis vector) represents one program feature. The values of the vector coordinates assigned to the observed TV program can be determined in two ways. The feature vectors might be binary, where one would simply mean the presence of the current feature, and zero would mean its absence. A subtler solution is that the value of the coordinate reflects the relative amount of the related feature in the observed program, which implicitly introduces the elements of fuzzy sets. Term frequency-inverse document frequency (TF-IDF) or similar methods from information retrieval could be used for this purpose.

To lower the computational cost, it might be convenient to reduce the dimension of the original vector space by means of linear transform, such as principal component analysis (PCA). As an example, the 256-dimensional space in [4] was transformed into 3-dimensional one in which the basis vectors have the following meanings: relaxing, informative, and emotional.

E. User Feedback

Finding a balance between explicit and implicit user modeling is an imperative. Many recommender systems ask their users to provide explicit feedback on suggested items in terms of ratings. In the context of digital TV, we find this approach obtrusive and annoying. On the other hand, attempts to obtain implicit feedback by observing how long the user has watched certain program might lead to misconclusions. For example, the user might be dressing up to go out and then turn on the news channel for few minutes just to see the weather forecast. This certainly does not mean that she is not interested in the news at all, as one might conclude by measuring the viewing time. An acceptable approach to obtaining relevant and yet user-friendly feedback could be with "Like/Dislike" buttons on the remote, as in [5].

We use these guidelines to design a new TV recommender system which is described in the following section.

IV. Our Proposal

Our proposal relies on the theory of information retrieval, from which we adopt the vector space model. It is the extension of the general telecommunications service retrieval framework that we presented in [11].

We apply the user modeling and program retrieval algorithm shown in Fig. 1 to find the program(s) of expected interest to the observed user.

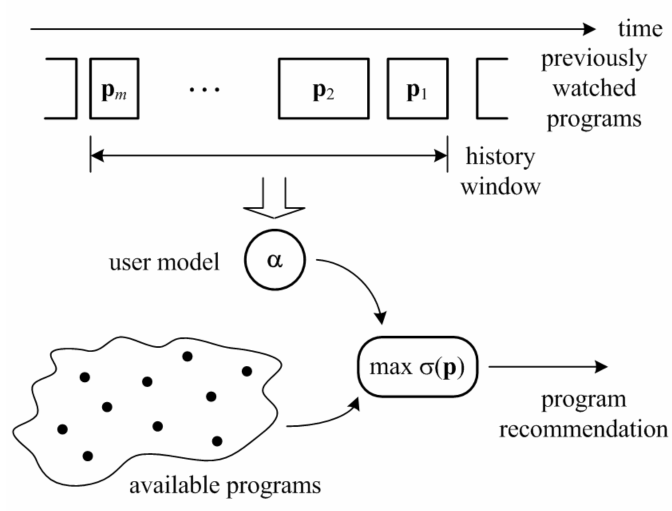

Fig. 1. User modeling and program retrieval.

TV program $\mathbf{p}_i$ is represented by n-dimensional vector $(p_{i1}, p_{i2,}\ldots, p_{in})$ whose coordinates are program features. The user model is given by $\boldsymbol{\alpha} = (\alpha_{1,} \alpha_{2,}\ldots, \alpha_n)$ and it could be regarded as target program in the observed vector space Ω.

The correspondence of some program $\mathbf{p}_i$ to the user model $\boldsymbol{\alpha}$ is given with similarity function

$$\sigma(\mathbf{p}_i) = \frac{\boldsymbol{\alpha}\cdot\mathbf{p}_i}{|\boldsymbol{\alpha}||\mathbf{p}_i|} = \frac{\sum_{j=1}^{n}\alpha_j p_{i,j}}{\sqrt{\sum_{j=1}^{n}\alpha_j^2}\sqrt{\sum_{j=1}^{n}p_{i,j}^2}}, \quad (1)$$

where · denotes scalar product and | | magnitude of the vector. Let us note that $\sigma(\mathbf{p}_i)$ equals cosine of the angle between the considered vectors; therefore (1) has unit maximum value for $\mathbf{p}_i = \boldsymbol{\alpha}$.

Recommendation agent, stored within user's TV receiver or in the set-top box, keeps the record on m programs $\mathbf{p}_1, \ldots, \mathbf{p}_m$ that the user has previously watched, $\mathbf{p}_1$ being the most recent one. It uses these data to estimate the optimal user model $\boldsymbol{\alpha}$; in terms of machine learning and pattern recognition, the agent uses the programs from the history window as a training sequence. After estimating the user model, the agent then searches over the available programs to find few with the greatest similarities to $\boldsymbol{\alpha}$. These programs are finally recommended to the user.

Let us now discuss how to compute the optimal user model. At first, one might expect the evolution patterns of user's interest to form a curve when represented in the vector space model. If this would hold true, then the optimal user model could be estimated by extrapolation, as shown in Fig. 2 a). Contrary to this, cluster hypothesis from the information retrieval states that consistent user queries form a cluster, not a curve [10]. Therefore, the optimal user model should be estimated by averaging, as a centroid of the cluster of vectors that correspond to the programs the user has previously watched, which is shown in Fig. 2 b).

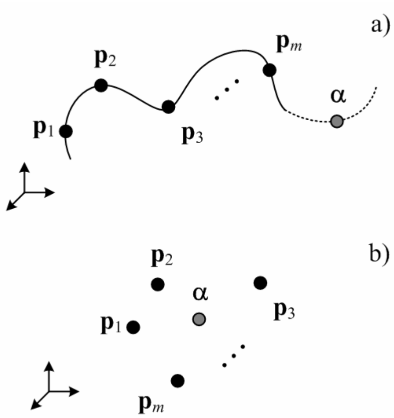

Fig. 2. Optimal user model estimation in vector space: a) extrapolation, b) clustering.

This can be formally written as

$$\boldsymbol{\alpha}_{opt} = \arg\max_{\boldsymbol{\alpha}\in\Omega} \sum_{i=1}^{m} w_i\sigma(\mathbf{p}_i), \quad (2)$$

where w_i are optional weighting coefficients that assign unequal importances to the watched programs.

Without loss of generality we can further assume unit magnitudes of program vectors in Ω. The coordinates of the optimal user model will then be given with

$$\alpha_j = \frac{S_j}{\sqrt{\sum_{k=1}^{n} S_k^2}}, j = 1, \ldots, n, \quad (3)$$

where

$$S_j = \sum_{i=1}^{m} w_i p_{i,j}, j = 1, \ldots, n. \quad (4)$$

Derivation details are given in the Appendix. Let us note that complexity of computing each α_j is $O(m \cdot n)$, where O is Landau O-symbol.

As illustrated in Fig. 3, clustering could be used to model the users whose interests are heterogeneous or inconsistent, with models assigned to each cluster. Similarly, "partial" user models could be also derived for different parts of a day, days in a week etc. Moreover, multiple users who share the same receiver could be identified by their distinct clusters without logging in.

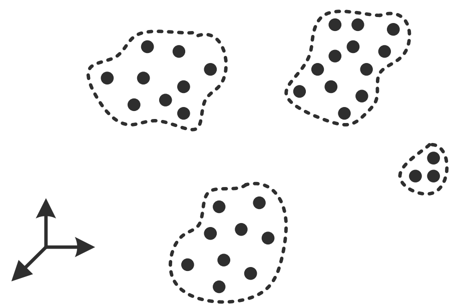

Fig. 3. If the user is interested in multiple heterogeneous areas, non-overlapping clusters will exist in the vector space.

The estimated user model is now used to retrieve the programs of expected interest to the observed user. The recommendation agent calculates the similarities of the available programs with the current user model and presents a few top-ranked programs to the user. An example interaction with the recommender system could go as follows:

User: (presses the "Recommend" button on her remote)
System: (displays on the TV) "Would you prefer a movie or news?"
User: (selects movie)
System: (recommends three movies).

Should the user accept any of the recommended programs, it would be considered as positive feedback and the current user model would be updated to include that program. The same applies if the user decides not to ask for recommendation, but finds a program by herself. Otherwise, if the user is not satisfied with the program she is currently watching, she would press the "Dislike" button on the remote control and that program would not be included in her profile; negative feedback is therefore stated explicitly. In the cases where the same TV receiver is used by multiple users, negative feedback could be used to protect one's privacy, since these programs would not be included in the profile.

V. Performance Evaluation

In the absence of reliable large-scale data regarding TV viewing habits, for the experimental validation of our proposal we used the datasets with movie recommendations that were gathered at the GroupLens Research Project, a research group in the Department of Computer Science and Engineering at the University of Minnesota, USA (http://www.grouplens.org). The "million-ml-data" dataset consists of 1000209 ratings for 3900 movies, given by 6040 users.

We randomly chose 80 users who generated 22402 interactions with the system. The distribution of these ratings is given in Table I. The average rating was 3.6.

TABLE I
Distribution of Ratings Within Selected Sample

Rating	1	2	3	4	5
Percentage	5%	11%	25%	36%	23%

For each movie, a genre description according to the data from Internet Movie Database (http://www.imdb.org) is provided. We use this description to generate feature vectors as follows. We construct an 18-dimensional vector space, where the coordinates have the following meanings: action, adventure, animation, children's, comedy, crime, documentary, drama, fantasy, film noir, horror, musical, mystery, romance, Sci-Fi, thriller, war, and western, respectively. Should any of these features exist in the description of the observed movie, a "1" is assigned to the matching coordinate; otherwise, its value is "0". Finally, the feature vectors are normalized to unit magnitudes.

Let us illustrate this on an example. 920th movie in the dataset is "Gone with the Wind" and its description is "drama-romance-war". The initial vector that we assign to it is

$$(0, 0, 0, 0, 0, 0, 0, \mathbf{1}, 0, 0, 0, 0, 0, \mathbf{1}, 0, 0, \mathbf{1}, 0),$$

with ones on the eighth (*drama*), fourteenth (*romance*) and seventeenth position (*war*). After the normalization, this gives

$$\mathbf{p}_{920} = (0, 0, 0, 0, 0, 0, 0, 0.577, 0, 0, 0, 0, 0, 0.577, 0, 0, 0.577, 0).$$

The user interactions were sorted in chronological order and the series of Monte Carlo simulations was then conducted. These included user modeling, according to (3) and (4), making a list of three recommendations and observing the user's reaction. This reaction is considered to be positive should the user's rating for any recommended item be at least 3. Otherwise, if the rating was 1 or 2, or if the user did not rate any of the recommended items at all, the recommendation was considered to be unsuccessful.

Four weighting schemes were considered:

1) $w_i = 1$, which is equivalent to no weighting;
2) users' ratings (1-5) were used as weights;
3) $w_i = 0.8^{m-i}$ (m is the size of history window), which means that the older interactions were favored, and
4) $w_i = 0.8^{i-1}$, so that the newer interactions were favored.

Throughout the available literature, the opinions on metrics that should be used to evaluate the performances of recommender systems are confronting; consequently, it is difficult to compare different results. Some authors use metrics adopted from information retrieval, like precision, recall, or F-measure. However, we find this approach inadequate for the following reason: The main goal of information retrieval is finding *all* the items that match the user's query, while the recommender system should find only *few* items the user is expected to like. Speaking of TV, we do not intend to list all the movies that are aired at a given time, but rather to select three of them that the user is likely to be interested in. It would hardly make any sense to define precision or recall over a small sample like this. Instead, we chose to observe the recommendation success rate, defined as the percentage of the interactions that were finished successfully (user's rating was at least 3). Fig. 4 shows how the success rate depends on the window size m.

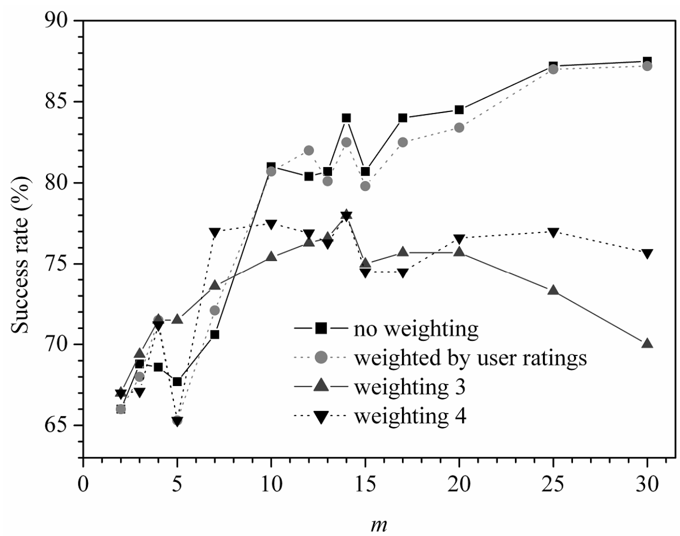

Fig. 4. Simulation results: The dependence of success rate (in %) on history window size, *m*, for four weighting schemes.

These results are discussed in the following section.

VI. Discussion

Our proposal falls under content-based recommendation strategies. Therefore, it is no surprise that it does not perform well for small sizes of history window. This is the clear manifestation of cold start problem that is especially visible for $2 \le m \le 5$.

For m between 5 and 15, the history window is probably still not of the enough size to encompass the dominant cluster of user's interests, but captures its different regions instead. Therefore, the performances rapidly change and it is not possible to identify single best weighting strategy.

For values of m greater than 15 our four strategies form two distinct groups of lines. The success rates of the first two schemes (no weighting and weighting by the ratings) continue to improve. We believe that this is the area where the effect of overspecialization is dominant, and the system enters the "steady state" with success rate of about 87%. At the same time, the performances of weighting strategies number 3 (favors the old interactions) and 4 (favors the new ones) start worsening; perhaps the weighting now overdistorts the cluster.

These results show that "no weighting" strategy applied for $m \geq 25$ is clear candidate for practical applications. This justifies our opinion that there is no need for the users to express their feedback by rating the recommendations, but rather by much simpler (and more comfortable) use of "Dislike" key on the remote instead. The optimal value for m in our experiments is 25, since greater values do not seem to bring significant improvement in performance.

VII. Conclusion

The most important factors that determine the design of TV recommender systems are demand of unobtrusiveness, scarce user interface, and the absence of return channel. In this paper we discuss their impact and show that it could be compensated through proper modeling of the user's viewing interests. The demand of unobtrusiveness and the scarce user interface jointly advocate the use of simple feedback, while the absence of return channel implies the use of content-based recommendation techniques. The overspecialization of these systems is elsewhere considered as their serious limitation, but here it might be a valuable feature. With these postulates accepted, the next steps in our work were to apply the vector space model and clustering hypothesis from the information retrieval. We proposed an analytical model for estimation of user's interests and ran a series of Monte Carlo simulations to verify its performances. The best tested strategy achieved success rate of 87% for history window size m = 25. The success rate this high proves the quality of the proposed user modeling and program recommendation procedure.

Our future work in this area will include the examination of more advanced clustering techniques that would be able to model heterogeneous areas of user interests. For the time being we solely use "positive" or "white" user profile, which includes those programs the user has liked, while the disliked ones are simply ignored. It remains for some future research to examine if the use of separate "white" and "black" profiles would be better. We are also looking forward to testing our system under real-life deployment scenario, as a part of digital TV service.

Appendix

Here we will show how coordinates of the optimal user model, given with (3) and (4), are obtained.

We choose to restrict our considerations to the unit hyperspheric surface:

$$(\forall \mathbf{x} \in \Omega) \quad |\mathbf{x}| = 1 . \tag{5}$$

Under this assumption, (1) reduces to scalar (dot) product,

$$\sigma(\mathbf{p}_i) = \sum_{j=1}^{n} \alpha_j p_{i,j} . \tag{6}$$

Our task is to maximize the sum of weighted similarities

$$f = \sum_{i=1}^{m} w_i \sigma(\mathbf{p}_i) = \sum_{i=1}^{m} w_i \left(\sum_{j=1}^{n} \alpha_j p_{i,j} \right) \tag{7}$$

under the constraint $|\boldsymbol{\alpha}| = 1$, which is equivalent to

$$\sum_{j=1}^{n} \alpha_j^2 = 1 . \tag{8}$$

Let us form the goal function

$$\begin{aligned} g(\boldsymbol{\alpha}, \beta) &= f - \beta(|\boldsymbol{\alpha}|^2 - 1) = \\ &= \sum_{i=1}^{m} w_i \left(\sum_{j=1}^{n} \alpha_j p_{i,j} \right) - \beta \left(\sum_{j=1}^{n} \alpha_j^2 - 1 \right), \end{aligned} \tag{9}$$

where β is Lagrange multiplier. After finding its derivatives with respect to α_j, j = 1, …, n and setting them equal to zero, we get

$$\alpha_j = \frac{\sum_{i=1}^{m} w_i p_{i,j}}{2\beta} . \tag{10}$$

Substituting in (8) yields

$$2\beta = \sqrt{\sum_{k=1}^{n} \left(\sum_{i=1}^{m} w_i p_{i,k} \right)^2} . \tag{11}$$

Finally, from (10) we get

$$\alpha_j = \frac{\sum_{i=1}^{m} w_i p_{i,j}}{\sqrt{\sum_{k=1}^{n} \left(\sum_{i=1}^{m} w_i p_{i,k} \right)^2}} = \frac{S_j}{\sqrt{\sum_{k=1}^{n} S_k^2}} , \tag{12}$$

where

$$S_j = \sum_{i=1}^{m} w_i p_{i,j} . \tag{13}$$

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", *IEEE Trans. Knowledge and Data Eng.*, vol. 17, no. 6, pp. 734-749, June 2005.

[2] R. Burke, "Hybrid recommender systems: survey and experiments", *User Modeling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331-370, November 2002.

[3] L. Ardissono, C. Gena, P. Torasso, F. Bellifemine, A. Difino, and B. Negro, "User modeling and recommendation techniques for personalized electronic program guides" in L. Ardissono et al. (eds.) *Personalized Digital Television*, Kluwer Academic Publishers, 2004, pp. 3-26.

[4] T. Isobe, M. Fujiwara, H. Kaneta, T. Morita, and N. Uratani, "Development of a TV reception system personalized with viewing habits", *IEEE Trans. Consumer Electron.*, vol. 51, no. 2, pp. 665-674, May 2005.

[5] Z. Yu and X. Zhou, "TV3P: an adaptive assistant for personalized TV", *IEEE Trans. Consumer Electronics*, vol. 50, no. 1, pp, 393-399, February 2004.

[6] Y. B. Fernandez, J. J. P. Arias, M. L. Nores, A. G. Solla, and M. R. Cabrer, "AVATAR: an improved solution for personalized TV based on semantic inference", *IEEE Trans. Consumer Electron.*, vol. 52, no. 1, pp. 223-231, February 2006.

[7] A. B. B. Martínez, J. J. P. Arias, A. F. Vilas, J. G. Duque, and M. L. Nores, "What's on TV tonight? An efficient and effective personalized recommender system of TV programs", *IEEE Trans. Consumer Electron.*, vol. 55, no. 1, pp. 286-294, February 2009.

[8] C. Shin and W. Woo, "Socially aware TV program recommender for multiple viewers", *IEEE Trans. Consumer Electron.*, vol. 55, no. 2, pp. 927-932, May 2009.

[9] M. L. Nores *et al.*, "Receiver-side semantic reasoning for digital TV personalization in the absence of return channels", *Multimedia Tools Appl.*, vol. 41, no. 3, pp. 407-436, February 2009.

[10] T. Volkmer, J. A. Thom, and S. M. M. Tahaghoghi, "Modeling human judgement of digital imagery for multimedia retrieval", *IEEE Trans. Multimedia*, vol. 9, no. 5, pp. 967-974, August 2007.

[11] M. Bjelica and Z. Petrović, "A novel service retrieval scheme", *IEEE Commun. Lett.*, vol. 11, no. 7, pp. 637-639, July 2007.

[12] C. J. Van Rijsbergen, *Information retrieval*, 2nd Ed. Butterworths: London, 1979, ch. 3.

BIOGRAPHY

Milan Bjelica (M'10) became a Member (M) of IEEE in 2010. He was born in Mostar, Bosnia and Herzegovina in 1977. He received B. Sc. (dipl. ing.) degree in 2000, M. Sc. degree in 2003, and Ph. D. degree in 2009, all from School of Electrical Engineering (ETF), University of Belgrade, Serbia. He is currently with the ETF Communications Department, working as assistant professor. His research interests include recommender systems, broadband communication networks, communication protocols for energy-efficient networking, multimedia, simulation, and modeling.