Exploiting Social Tagging in a Web 2.0 Recommender System

Recommender systems help users cope with information overload by using their preferences to recommend items. To date, most recommenders have employed user ratings, information about user profiles, or metadata describing the items. To take advantage of Web 2.0 applications, the authors propose using information obtained from social tagging to improve recommendations. The Web 2.0 TV program recommender queveo.tv currently combines content-based and collaborative filtering techniques. This article presents a novel tag-based recommender to enhance the recommending engine by improving the coverage and diversity of the suggestions.

Ana Belén Barragáns-Martínez
Centro Universitario de la Defensa en la Escuela Naval Militar de Marín, Spain

Marta Rey-López
Consellería de Educación e O.U., Spain

Enrique Costa-Montenegro, Fernando A. Mikic-Fonte, Juan C. Burguillo, and Ana Peleteiro
University of Vigo, Spain

The number of channels and programs available to digital TV consumers is constantly increasing. To tackle this information overload problem, researchers have begun developing TV recommendation mechanisms for digital TV systems. In this context, we've designed queveo.tv (www.queveo.tv), an innovative Web 2.0 application that learns about individual users' TV viewing preferences to provide them with highly customized daily TV content recommendations. Queveo.tv combines the two most successful techniques in the area of recommender systems — collaborative filtering (CF) and content-based filtering (CBF) — as well as a well-known matrix factorization technique in the implementation of the item-based CF algorithm.

Using queveo.tv as a starting point, we propose taking advantage of the system's social tagging capabilities (where users can add tags to describe items) to enrich the quality of the recommendations. The tags that users apply to items describe both the items and the users themselves. Such social tagging also lets us create a *folksonomy*, which shows the relationships between the different tags.[1] Using all this information, we can measure the similarity between users and items, as well as between potential and past users, with the aim of recommending items to users. This type of recommendation considers the user rather than the content creator viewpoint and lets us work with a more detailed item description rather than taxonomies.

1089-7801/10/$26.00 © 2010 IEEE

Such an approach would improve the quality of recommendations twofold. First, the system would gain more semantic interconnections among the items themselves and, consequently, enable more relevant recommendations, which in turn would improve both the coverage and recall of the recommendations. Additionally, we could also increase the diversity because a tag-based recommender would access items not recommended with the current implementation (because they belong to genres outside the user-specified preferences).

Tag-based recommendation techniques enable a further improvement specific to this application domain (TV shows). Because user ratings are assigned to tags rather than to the items themselves, this information is valuable even when these items are no longer available. This is particularly important in a TV recommendation system, where many items only stay in the system for a short time.

Recommender Systems

Recommendation techniques come in two basic flavors. CF approaches rely on the availability of user ratings information and make suggestions for some target user *u* from the items preferred by similar users (user-based) or from items that received similar ratings to the items that *u* likes (item-based).[2] In contrast, CBF techniques rely on item descriptions and generate recommendations from items that are similar to those the target user has liked in the past.[3]

The standard CF approach presents some well-known problems:

- The *gray-sheep problem* involves users that have uncommon preferences that don't overlap much with those of other system users.
- The *cold-start problem* occurs when new system users haven't submitted any ratings yet.
- The *first-rater problem* arises when new items can't be recommended until some users have taken the time to evaluate them.

These problems with CF systems can be solved by using CBF techniques, which let users encounter new content that hasn't been rated before. To do so, the system must be able to match an item's characteristics against relevant features in the user's profile. In the same way, CBF techniques also let new users find interesting content. This is the technique typically applied in TV programming recommendations through personalization of the electronic program guide (EPG) using only CBF techniques.

The primary drawback to CBF systems is their tendency to overspecialize item selection because they only recommend items similar to those the user has previously liked. We can eliminate this problem by combining CF and CBF techniques.

For these reasons, we previously proposed adopting a hybrid approach for the TV recommender domain.[4] In the initial stage, this system can infer program recommendations from viewer preferences before a user's viewing history is available. As the system gathers more information about a user's viewing history and the given ratings, it can fine-tune program recommendations to match personal preferences using the CF approach.

Moreover, users generally only rate a small fraction of the total amount of available programs, making it difficult to confidently identify similar users and items. Another important limitation of CF systems is their lack of scalability: as the number of users and items increases, this approach can lead to unacceptable latency for providing recommendations during user interaction. To eliminate both problems, we proposed using singular value decomposition (SVD) technology[5] to reduce the dimensionality of the recommender system database.[6]

Tag-Based Recommenders

Tag-based recommender systems include those that use tag information to enhance traditional algorithms used in recommender systems and those built from scratch by using the information associated with tags.

Among the approaches belonging to the first category, the most relevant one to our work presents new tag-based similarity measures to compute the user and item neighborhood and applies them to improve the standard user- and item-based CF approach.[7] Social ranking approaches find content that's relevant to a user's query, extending traditional recommender system techniques with tags.[8] However, this system doesn't account for user ratings. (A full survey of tag-based recommendation techniques is available elsewhere.[9])

Our hybrid system belongs to the second category because it's completely based on tags, folksonomies, and ratings. However, we can also

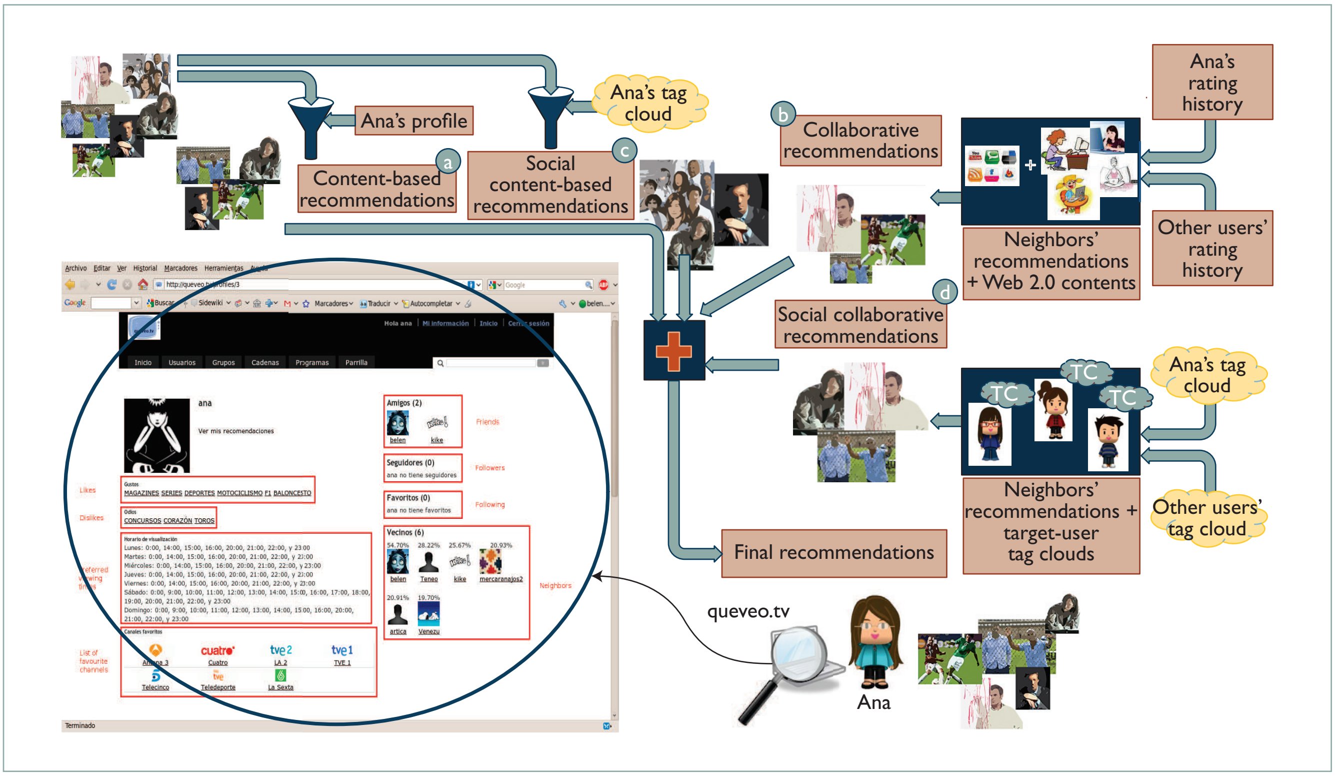

Figure 1. Overview of our hybrid recommendation system. The four recommendation techniques are (a) content-based, (b) collaborative, (c) social content-based, and (d) social collaborative recommendations.

make an analogy with traditional classification of recommendation techniques. Although most approaches in tag-based recommender systems use a direct relationship to measure the similarity between a set of tags (which considers the coincident tags),[7,9] our approach accounts for the relationships between those tags, thus finding relationships even when there aren't common tags. Our system also exploits collaboration among users in a generic way, not based on particular users and items.

Overview of Queveo.tv

The recommendation component is the intelligent core of queveo.tv; it takes the user profile information (including likes, dislikes, ratings history, recent activity, and so forth) and selects new programs to recommend.

On one hand, the system uses three components for content-based recommendation (see Figure 1a):

- content descriptions for all TV programs,
- a compatible content description of each user's profile, and
- a procedure for measuring the similarity between a program and a user.

We use the vector-space model to generate content-based recommendations, representing programs and user preferences as vectors of attributes. This model's basic concept is the selection of the item vectors that are the most similar to a user's preferences vector. In a TV program recommendation system, there are as many item vectors as the number of programs and only one user's preferences vector. The vector-space model selects a desired item by calculating the similarity between each of the item vectors and the user's preferences vector, in our case, employing the cosine measure.

On the other hand, the CF module (see Figure 1b) generates the collaborative recommendations for the user Ana from the ratings history of all the users in the system and Ana's ratings history. Research shows that item-based nearest-neighbor algorithms are more accurate in predicting ratings than their user-based counterparts.[10] Thus, we only use the user-based CF approach to generate the set of neighbors for the active user (using the cosine similarity measure to find the top-*M*-most similar users) but not to generate the recommendations. That set of neighbors appears in the public user profile (see the left side of Figure 1), enabling users

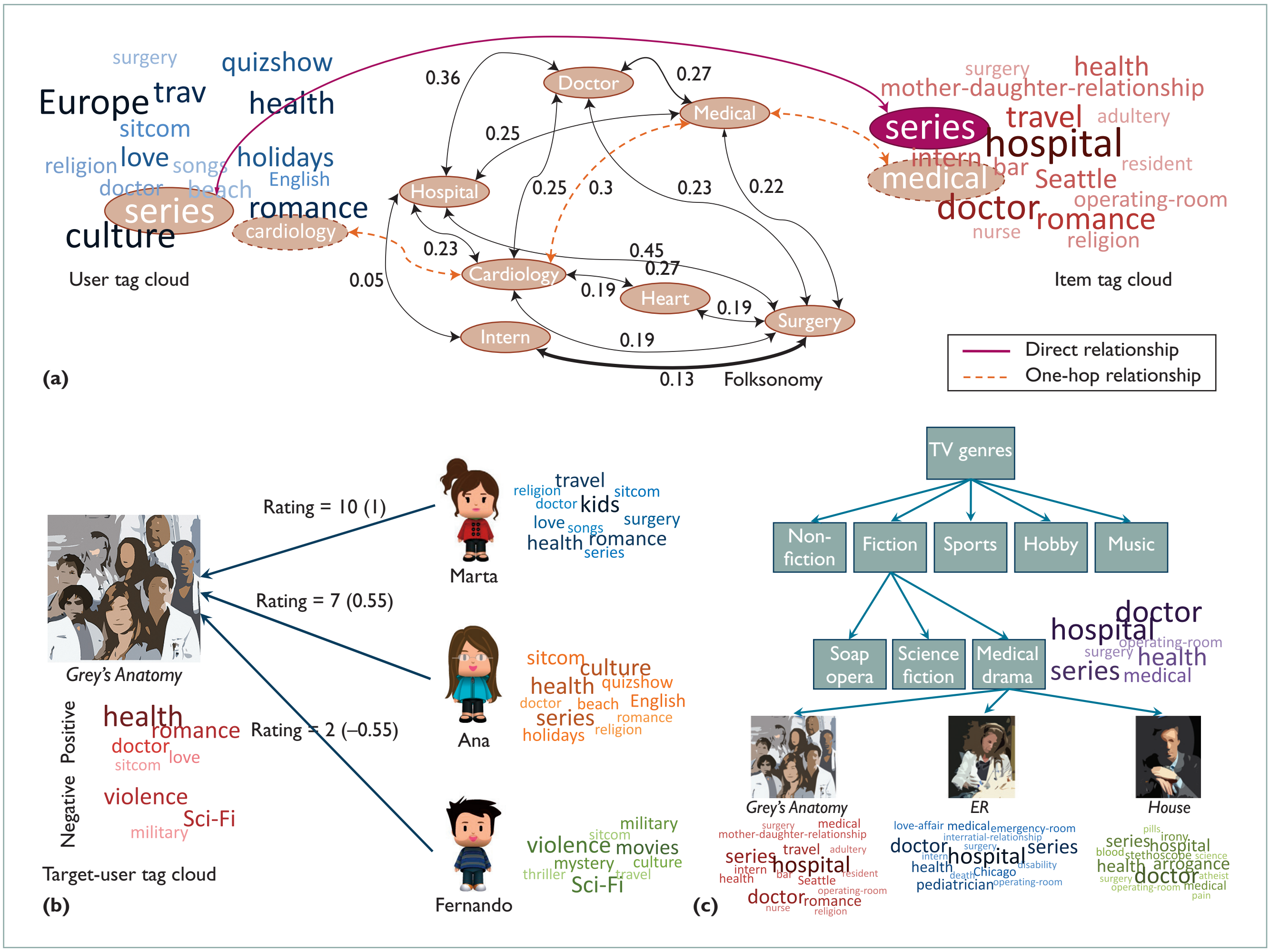

Figure 2. Social recommendation. Our process includes (a) obtaining the relationships, (b) creating the target-user tag cloud, and (c) creating the stereotype.

to find the people who most closely share their tastes and to navigate a potentially immense social network. (A more rigorous mathematical description is available in related work.[4])

Queveo.tv uses an item-based CF approach to predict the level of interest for each item and to calculate the items' neighborhood. The algorithm we use in queveo.tv uses the matrix factorization SVD technique to reduce the dimension of the active item's neighborhood, and then it executes the item-based filtering with this low rank representation to generate its predictions.

The hybrid proposal works well because the algorithms complement each other; CBF recommends the usual programs, and CF helps discover new ones. This lets us enrich our recommender system with two new recommendation techniques (see Figures 1c and 1d), both based on social tagging information.

A Tag-Based Recommender System

To enrich our system's recommendations, we propose using *social recommendation* (see Figure 2). As we previously mentioned, queveo.tv lets users give items tags to describe them. We use these tags to build both user and item tag clouds. User clouds consist of tags users have never assigned, whereas an item tag cloud includes the tags users have assigned to it (see Figure 2a). In both cases, the weight of the tags — its importance — is proportional to the number of times they have been assigned (by the user or to the item). In user tag clouds, a tag's weight is also proportional to the ratings the users gave the items. As users indicate their ratings in queveo.tv from one to 10, we normalize this value to get values from –1 to 1 for positive and negative ratings, respectively. Therefore, the weights of the tags in user tag clouds can have positive or negative values.

Item tag clouds also reflect the relationships between the system's tags. We can represent this structure, called a folksonomy,[1] as an undirected graph in which nodes are the system tags and arcs represent the relationships between the tags they link.[11] The weights of these relationships (r_{jk} in Equation 3) are proportional to the number of times tags appear together in an item description (see Figure 2a).

The simplest way of recommending items to users is by directly comparing their tag clouds. In this manner, we provide CBF based on tags, or social CBF (SCBF, Figure 1c). To obtain the similarity between user and item tag clouds, we measure the number of coincident tags of both tag clouds (direct relationship, R_0) as well as the relationships between the user's and item's tags (one-hop relationship, R_1), as Figure 2a shows. Hence, we can calculate the relationship between the item i and the user u as follows:

$$R(i,u) = \alpha R_0(i,u) + (1-\alpha) R_1(i,u), \quad (1)$$

$$R_0(i,u) = \sum_{\forall t \in |T_i \cap T_u|} \sqrt{w(t,T_i)\, w(t,T_u)}, \quad (2)$$

$$R_1(i,u) = \sum_{\substack{\forall j / t_j \in T_i \\ \forall k / t_k \in T_u}} \sqrt{w(t_j,T_i)\, w(t_k,T_u)}\; r_{jk}, \quad (3)$$

where T_i is the set of tags in the tag cloud of item i, T_u is the set of tags in the tag cloud of user u, $w(t, T)$ is the weight of tag t in the tag cloud T, α is the parameter used to indicate the importance of each type of relationship ($\alpha \in [0, 1]$), and r_{jk} is the relationship in the folksonomy between the tags t_j and t_k.

However, as we explained earlier, SCBF selects items that are similar to the ones the user has consumed before. For this reason, we improve social recommendations by complementing SCBF with a social collaborative filtering (SCF) technique (see Figure 1d). For this to be possible, we need a new tag cloud for the items, called the *target-user tag cloud*, which describes the user for whom the item is appropriate and is made from the tag clouds of the users who have viewed the item and the ratings they have provided for it. As Figure 2b shows, users assign a rating to the items they've viewed. Hence, the user tag clouds are weighted by their ratings to compose the target-user tag cloud. Once this new tag cloud is created, the system compares it with the potential users' tag clouds to obtain their similarity (see Figure 2a). This similarity is used to obtain the SCF recommendations.

Both SCF and SCBF suffer from first-rater and cold-start problems. When new users or items enter the system, they have neither a tag cloud nor a target-user tag cloud (the latter applies only for items). To alleviate this problem, we chose a hybrid approach for item description that combines the social tagging approach with taxonomies. We use the taxonomy for TV programs provided by TV-Anytime[12] to classify items according to their TV genre. Using the item tag clouds of all the programs of each genre, we can obtain a stereotype tag cloud for this TV genre (Figure 2c) using an approach similar to the one we used for the target-user tag clouds. We do this for two reasons: a tag cloud is assigned to new content until users tag the content, and we use it to compose the initial user tag cloud. For the latter, we ask users about their favorite TV genres when they join the system. Combining the stereotype tag clouds with the relevant TV genres, we compose the provisional user tag cloud, which the system uses until users have their own.

Because this social-recommendation algorithm is based on user participation, it won't work for passive users who don't supply item tags. With the aim of mitigating this drawback, we compose tag clouds for these users by combining the viewed items' tag clouds weighted by the ratings they provide, in an approach similar to the one in Figure 2b but the other way around. (More detailed formulas and technical specifications are available elsewhere.[13])

Illustrative Example

The following example helps illustrate the advantages of adding the tag-based recommender to queveo.tv. For the sake of clarity, we use a simple example with only five users and six TV programs (see Table 1).

Suppose that a new user Juan is entering queveo.tv. When registering and filling out his profile, he selects the categories "Documentaries: General and Medical" and "Sports: Basketball" as his likes because he's especially interested in documentaries about medical topics, such as surgery. Immediately, the recommender can suggest some TV programs from all the shows that will be broadcast in the next three days on any channel that Juan can access (he has previously specified this information

Users	Apollo 13	More than a Game	NBA Action	The Operation: Surgery Live	House	Nip/Tuck
Ana	9	1	2	–	10	–
Marta	–	9	–	10	–	–
Enrique	7	–	10	–	8	3
Fernando	2	9	8	9	8	8
Juan	–	10	9	–	–	–

*Table 1. User-program matrix A.**

* Items with a – value have not been rated yet.

in his profile). This can be done thanks to the CBF algorithm, which suffers from neither cold-start nor first-rater problems. The CBF algorithm's output consists of those TV programs (sorted according to relevance) that match his likes — in our example, *More Than a Game* and *NBA Action*, which are a documentary and a sports program, respectively, both about the US National Basketball Association (NBA).

Initially, the CF algorithm was unable to give any recommendation because Juan hadn't rated any programs. After watching the queveo.tv suggestions, let's suppose Juan rates the documentary with a 10 (on a scale from one to 10) and the sports program with a 9, as Table 1 shows.

Now our three recommenders can offer Juan suggestions, upon request. The CBF algorithm recommends another documentary, *The Operation: Surgery Live*, where top surgeons carry out life-changing operations in front of a studio audience. This recommendation fits perfectly with Juan's main interest.

After Juan has rated some programs, the CF technique must process the user-item matrix that contains the ratings each user gives for all programs on a scale from one to 10 (see Table 1). Matrix A (Table 1) is processed as we previously explained. Again, the main goal of our item-based CF approach is to precisely fill these empty values with predictions. The result of these computations is the prediction matrix in Table 2. Consequently, and for a recommendation threshold of seven, the CF algorithm recommended the documentary *The Operation: Surgery Live* and the TV series *House*. The CF algorithm returns the programs that are similarly rated to those highly rated by Juan — for example, *More Than a Game* (which Juan gave a 10) was rated with the same pattern as *The Operation: Surgery Live* — that is, both Marta and Fernando gave a similar rating to both. In the same way, Enrique and Fernando gave similar ratings to *NBA Action* and *House*. Without the tag-based recommender, the final results would be *The Operation: Surgery Live* (both CF and CBF recommend it) and *House*.

Using our tag-based recommender strategy, the system can now use the information from both Juan's tag cloud and the tag clouds from each program. Because Juan has used terms such as surgery and doctor to tag the documentary *The Operation: Surgery Live*, the SCBF algorithm finds new relevant content to recommend: the TV series *Nip/Tuck* (focused on a plastic surgery practice) that other users have previously tagged with the same tags or with tags related to the former through the folksonomy (a one-hop relationship).

Additionally, the SCF algorithm also recommends to Juan the movie *Apollo 13* because its target-user tag cloud contains tags that are also in Juan's tag cloud or related to them through the folksonomy. Even though the tags such as doctor or surgeon appear on the *Apollo 13* target-user tag cloud, that doesn't necessarily mean that the movie deals with these subjects. It reflects that users who liked items with these tags also liked *Apollo 13*.

Both *Nip/Tuck* and *Apollo 13* hadn't been suggested before because the CBF technique doesn't recommend TV series or movies (because Juan didn't choose these genres in his profile). The CF approach didn't find that *Nip/Tuck* or *Apollo 13* had a similar rating pattern to any program Juan rated highly. SCBF didn't recommend *Apollo 13* because it doesn't contain any coincident or related tags to Juan's in its tag cloud (only in its target-user tag cloud).

Table 2. User-program matrix A after being filled in with the predicted values.*						
Users	**Apollo 13**	**More than a Game**	**NBA Action**	**The Operation: Surgery Live**	**House**	**Nip/Tuck**
Ana	9	1	2	9.5	10	–
Marta	6	9	7.25	10	8.66	5.5
Enrique	7	7.25	10	9.5	8	3
Fernando	2	9	8	9	8	8
Juan	6	10	9	9.5	8.66	5.5

* Cells bordered in orange indicate predictions obtained for user Juan.

From this illustrative example, we clearly see that using tag-based recommendation techniques (both SCBF and SCF) lets queveo.tv gain more semantic interconnections thanks to the use of folksonomies. They also obtain greater coverage because additional relevant items are now included among the recommendations.

Implementation Details

We developed the queveo.tv website using open source technologies, specifically the scripting language Ruby and the framework Rails. We also implemented the databases of users' profiles, tag clouds, programs, broadcastings, groups, ratings, comments, and TV channels in MySQL. The queveo.tv server runs on Ubuntu Jaunty Linux on an Intel Core2 Quad CPU Q9300 at 2.50 GHz with 4 Gbytes of RAM. We stress tested this system beyond 20,000 average hits per month, without any performance degradation.

To gather the complete listings of all TV channels, the application uses XMLTV (http://xmltv.org/wiki), which is a set of programs that processes TV listings and stores them in an XMLTV format. To compute the SVD, we used linalg (http://rubyforge.org/projects/linalg), which is a fast Lapack-based library for real and complex matrices. To alleviate the response time invested in computing new recommendations for a given user, we update the predictions matrix offline.

We're currently validating the whole application using undergraduate students. We carefully evaluated the previous version of queveo.tv with good results in terms of accuracy (see our mean absolute error [MAE] results[4]) and had a great user acceptance, which leads us to expect the same success in this highly improved version.

As we noted earlier, one of the main advantages of the system is making tags independent of the items because many TV program items are only available for a short time. CF program ratings are valid only while the programs are available, but with the proposed tag extension, the ratings are related to the tags rather than to the items themselves, which makes them valuable even when the items are no longer in the system. In the future, we will study the possibility of reducing the weights of the tags as they get older.

Our illustrative example helps indicate the potential effectiveness of this work, but we're aware that it isn't a substitute for an evaluation phase. Hence, it's our immediate goal to carry out a complete assessment of our research results. Our evaluation will consist of two phases:

- using well-known metrics such as the MAE value and the F1-measure and comparing the results with other previous works, and
- surveying users to determine how they perceive the recommendations' quality.

We also plan to design a system- and user-centric questionnaire.

In addition, we're currently working on creating a hybrid approach that combines the strategies we describe here, instead of obtaining separate recommendations for each one. Moreover, it would be interesting to improve social recommendations by providing queveo.tv with new features that let users judge other users' descriptions or identify their friends to create a circle of trust in which the ratings and tags friends provide are more trustworthy.

References

1. T. Vander Wal, "Folksonomy Coinage and Definition,"

2007; www.vanderwal.net/folksonomy.html.
2. P. Resnick et al., "GroupLens: An Open Architecture for Collaborative Filtering of Netnews," *Proc. ACM Conf. Computer-Supported Cooperative Work* (CSCW 94), ACM Press, 1994, pp. 175–186.
3. M.J. Pazzani and D. Billsus, "Content-Based Recommendation Systems," *The Adaptive Web*, P. Brusilovsky, A. Kobsa, and W. Nejdl, eds., LNCS 4321, Springer, 2007, pp. 325–341.
4. A.B. Barragáns-Martínez et al., "What's on TV Tonight? An Efficient and Effective Personalized Recommender System of TV Programs," *IEEE Trans. Consumer Electronics*, vol. 55, no. 1, 2009, pp. 286–294.
5. G. Golub and C.V. Loan, "Matrix Computations," *Johns Hopkins Studies in Mathematical Sciences*, 3rd ed., 1996.
6. M.G. Vozalis and K.G. Margaritis, "Applying SVD on Item-Based Filtering," *Proc. Intelligent Systems Design and Applications* (ISDA 05), IEEE Press, 2005, pp. 464–469.
7. H. Liang et al., "Collaborative Filtering Recommender Systems Using Tag Information," *Proc. IEEE/WIC/ACM Int'l Conf. Web Intelligence and Intelligent Agent Technology*, vol. 3, IEEE Press, 2008, pp. 59–62.
8. V. Zanardi and L. Capra, "Social Ranking: Uncovering Relevant Content Using Tag-Based Recommender Systems," *Proc. Conf. Recommender Systems* (RecSys 08), ACM Press, 2008, pp. 51–58.
9. A. Klasnja Milicevic, A. Nanopoulos, and M. Ivanovic, "Social Tagging in Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," *Artificial Intelligence Rev.*, vol. 33, no. 3, 2010, pp. 187–209.
10. B.M. Sarwar et al., "Item-Based Collaborative Filtering Recommendation Algorithms," *Proc. 10th Int'l Conf. World Wide Web* (WWW 01), ACM Press, 2001, pp. 285–295.
11. E. Michlmayr, S. Cayzer, and P. Shabajee, "Add-A-Tag: Learning Adaptive User Profiles from Bookmark Collections," *Proc. 1st Int'l Conf. Weblogs and Social Media* (ICWSM 07), 2007; www.icwsm.org/papers/4--Michlmayr-Cayzer-Shabajee.pdf.
12. TV-Anytime Forum, *Broadcast and On-line Services: Search, Select and Rightful Use of Content on Personal Storage Systems*, ETSI TS 102 822, 2004.
13. M. Rey-López et al., "T-Learning 2.0: A Personalised Hybrid Approach Based on Ontologies and Folksonomies," *Computational Intelligence for Technology Enhanced Learning*, Springer-Verlag, 2010.

Ana Belén Barragáns-Martínez is an associate professor at the Centro Universitario de la Defensa en la Escuela Naval Militar de Marín (Pontevedra). Her research interests include recommendation algorithms, using social networks, and Web 2.0 technologies. Barragáns-Martínez has a PhD in computer science from the University of Vigo. She's a member of IEEE, the IEEE Consumer Electronics Society, the ACM, and ACM Sigsoft. Contact her at belen.barragans@uvigo.es.

Marta Rey-López is a secondary school teacher in the area of electronics and telecommunications at Consellería de Educación e O.U., Spain. Her research interests include recommendation systems and the application of Web 2.0 technologies in the field of reasoning. Rey-López has a PhD in personalized learning through interactive digital TV from the University of Vigo. Contact her at research@martarey.es.

Enrique Costa-Montenegro is an associate professor in the Department of Telematic Engineering at the University of Vigo. His research interests include wireless networks, car-to-car communication technologies, multiagent systems, peer-to-peer systems, and recommendation technologies. Costa-Montenegro has a PhD in telecommunication engineering from the University of Vigo. Contact him at kike@det.uvigo.es.

Fernando A. Mikic-Fonte is an assistant professor in the Department of Telematic Engineering at the University of Vigo. His research interests include intelligent agents, distributed artificial intelligence, e-learning, telematics services, and recommendation systems. Mikic-Fonte has a PhD in telecommunication engineering from the University of Vigo. Contact him at mikic@det.uvigo.es.

Juan C. Burguillo is an associate professor in the Department of Telematic Engineering at the University of Vigo. His research interests include intelligent agents, multiagent systems, evolutionary algorithms, game theory, and telematic services. Burguillo has a PhD in telematics engineering from the University of Vigo. Contact him at jrial@det.uvigo.es.

Ana Peleteiro is a PhD student in the Department of Telematics at the University of Vigo. Her research interests include intelligent agents, multiagent systems, self-organization, evolutionary algorithms, and game theory. Peleteiro has an MSc in telecommunications engineering and an MSc in telematics engineering from the University of Vigo. Contact her at apeleteiro@det.uvigo.es.