Balanced Recommenders: A hybrid approach to improve and extend the functionality of traditional Recommenders

Javier G. Recuenco[2], David Bueno[1]

[1]Departamento de Lenguajes y Ciencias de la Computación. Universidad de Málaga
[2]AbyPersonalize
javier.recuenco@abypersonalize.com, bueno@lcc.uma.es

Abstract. The authors present a possible approach for a new general purpose recommender architecture, one which complements the current proven and tested techniques (User Model, Collaborative Filtering, Content Based Filtering), used in some everyday business scenarios, balancing with newly developed personalization procedures and methodologies. The overall objective is to try to tackle some of the typical shortcomings of traditional recommender systems (Cold start, dilution of the "personal color" in a sea of collective thinking…), by effectively balancing the amount of collective intelligence used against a more "personal affinity" score. This, the authors call PPM (Product Profile Matching), an approach which ignores collective results and relies mainly on the intrinsic affinity between the nature of both the subject and the item. Hence the use of the name "balanced", because of the balance struck between A.I. techniques and Applied Personalization Techniques used to make a better recommendation. The authors also focus on the need for proper self-fulfilling techniques in order to illustrate the paramount importance of improving and extending the control that existing recommender systems give users in order to optimize the user experience. An example based on the author's previous work in the field of TV content recommenders is presented to illustrate the validity of our approach.

1 Introduction

Information overload has become a problem in recent times. Increasingly, system users encounter difficulties in finding the information they need. Recommender Systems [1] have emerged as a way to reduce the amount of information users have to process in order to find something interesting. They have been applied to different areas of knowledge such as personalized newspapers (newsdude) [2], movie recommenders (movielens) [3], personal electronic programming guides ((PTV) [4]) or art recommenders [5]. In the rest of this article, the base element of the recommender system will be referred to as an item. Items may be documents, songs, news, TV programs, goods in a shop, pictures etc…

There are two main techniques used by existing recommender systems: content-based recommendations and collaborative recommendations [6]. In the first case, user recommendations are based on items similar to those s/he may have chosen in the past. An example of this is METIORE [7], which recommends publications; or myTV project [8] which is related to TV programming. In the second case, users are informed of recommendations based on similar users' preferences. A well-known example of this approach is Amazon.com [9] or Barnes&Noble both of which recommend books purchased by other clients with a similar profile. The Movielens recommender is also based on this technique. Ideally, the best solution benefits from both content and collaborative information. This is called the hybrid model and some interesting and relevant material can be found in [1] [10] [11].

Existing recommender systems have certain limitations, which although they do not hamper the overall usefulness of the system they prevent the "perfect recommendation" from being provided. The "perfect recommendation" is somewhat difficult to specify, but we define it as:

"*The result of ascertaining the exact desires of the individual using a recommender system, taking into account not only the knowledge of the whole network, but the particularities of the user AND the items available, which are relevant to the recommendation process*".

Some of the difficulties of recommenders are well known and are usually dealt with in different ways: "Cold start" is perhaps the best known one; clearly there is no real way for a recommender to provide useful recommendations from the start without an initial recommendation from other users. In Movielens different techniques are developed to select some items (films), shown to the user in order to create an initial model. One of the criteria to show initial items to the users is to rank items according to their particular relevance to these individuals. A good overview of the ranking algorithms is presented in [12] but most of these results are applied to queries made to documentary databases or to the Web like the popular PageRank ranking system [13]. We can also find ranking algorithms for blogs [14] to select the most popular ones according to the number of times they are read, the number of comments made and their voting average. Recent work [15] has tried to solve the cold start problem using the tied Boltzmann machine model, improved with content for collaborative recommendations. Another limitation is slightly more subtle in nature (dilution of the personal color in a sea of collective thinking): In a progressively personal world, where individual tastes are increasingly being better catered for, there is no such thing as the "perfect segment". Our aim for the recommender system is that it should approach as closely as possible the minimum segment size of 1. Segmentation is therefore a compromise between our ability to characterize a specific set of behaviors or attributes in order to define a user and the amount of available information and the real relevance and significance of those attributes connected to our context. So-called "Macrosegments" that can work correctly in a macro context (Women, Man 25-45..) are usually useless in terms of returning finely tuned recommendations. Each individual has a "color" of their own. Let us consider an example from a music recommender system, from the many currently available on the market (Pandora, Last.fm, Strands…) A hard rock music fan may also listen to a Synth Pop artist, and traditional recommenders will therefore associate that individual with a taste for BOTH kinds of music, so there will be a "poisoning" effect on future recommendations due to the apparent "anomaly", because the system does not handle "individual colors" but performs macrocluster mapping. The authors in [16] propose to solve this issue of different user 'faces' using a goal oriented recommendation, which keeps a common model and also a specific partial model for each of the user "goal/objectives". There is a risk that users may end up "belonging" to a specific cluster instead of what should really happen: A distinctive, unique personality should be matched to the shape of well known, well characterized "macroclusters", and the best fit selected. The current approach however could be compared to the process of making a random shape using paper and scissors and then trying to compare it with well known polygonal shapes: Circle, Pentagon, Octagon.., and then deciding which one fits best.

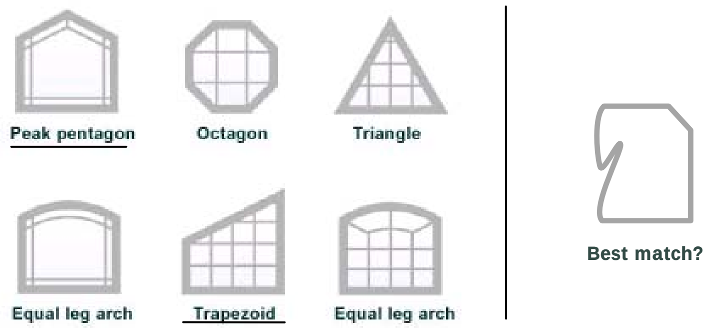

Fig. 1. The proposed shape shares some characteristics with the underlined pre-established shapes, but we cannot make a direct association to any of them beyond some shared characteristics

What we found interesting is that while we aim to achieve perfection in terms of pattern recognition and other such mathematical delicacies, we ignore as "non manageable" the capability to effectively and precisely draw a unique, non clustered, image of our user. This is where user modeling comes to the rescue: the main idea behind user modeling is to produce a "model" that tries to identify the key attributes belonging to a specific domain (in the case of Pandora, musical tastes) which can truly identify the user.

The problem with user modeling is a simple one: The model is produced (as accurately as possible), but it does not provide a suitable technique to ensure that several objectives are achieved. These objectives, fundamental in the overall process to guarantee a perfect personal recommendation experience, are the following:

a) To take the user from a "dummy" experience (i.e., one where they have had no involvement in the recommendation process) to being fully in control (fully tuning all the parameters included in the recommendation process) in a smooth and logical transition,
b) To provide an effective way of interacting with the user in order to engage him/her to produce more and more explicit feedback and profile detail.
c) To provide an effective framework for creating a constant "quid-pro-quo" scenario between provided data and improved responses from the recommender

User modeling provides a framework, but does not resolve the problem entirely. The user is not naturally enticed to cooperate, because there is no real incentive. In most approaches to recommendation engines there is one notable flaw: The systematic relying on Machine Learning and non-explicit feedback from the user to create the user model, where the possibility of truly engaging the user in the construction of their own profile is practically nonexistent. Why does this happen?

Mainly for one reason; most current approaches to recommender systems come from the "hard sciences" i.e., those related with knowledge based on rigid disciplines with fixed definitions such as mathematics and engineering Most of the sciences addressing the question of personalization however deal with "lighter" disciplines like etiology, psychology, marketing and so on , i.e., disciplines with a somewhat laxer approach to definitions and even contradictory solutions for identical problems. Therefore it seems that there is no possible way to rationalize and approach these disciplines systematically so it is best to rely on tried and tested scientific approaches. Unfortunately, although some work is emerging in this area [5], there is as yet very little literature existing on this matter[1]

2 Personalization: A Framework

Given the fact that "personalization" is a fairly vague word, which encompasses a lot of different definitions and approaches, with varying degrees of depth and no common consensus on the definition, we provide a series of basic components for our framework, dealing with the Personalization aspects of our work. Our proposal is a Balanced Recommender System defined as follows:

"A balanced recommender system is a approach which combines a recommender algorithm based on implicit, collective and behavioural data with a user's, explicit, user-centric and specific user model. The system uses additional tools and techniques provided to manipulate, enrich and fine tune the final recommendation."

The specific user model is not a generic one but depends entirely on the type of recommender involved (tv recommender, book recommender etc). Also the overall degree of involvement of the user in the creation of his/her profile has a significant impact on the quality of the final recommendation.

In this work we illustrate how we concluded that there was a need for this new type of recommender and describe the logic used to build our system.

2.1 Current use of the term "Personalization" in Recommender Systems

Supposedly, recommender systems - even the least sophisticated ones- deliver personalization. They deliver "personalized" recommendations, make "personalized" offers and deliver "personalized" messages. In our opinion however, this is not entirely the case. A detailed definition of personalization has been included in a previous reference[1] but for the purposes of this paper we will try to provide a less complex explanation:

"Personalization is a process which basically tries to adapt as closely as possible a product/message to a customer/speaker. The more accurate the analysis, the more accurate will be the recommendations. If we manage to grab the interest of our user

[1] One of the authors of this paper has published a book and several papers on a systematic approach for handling this problem, from which we have taken some definitions and some basic building blocks. Unfortunately, to date it is currently only available in Spanish: "Personalización" – Pearson Financial Times 2004 ISBN 9788420543543

and to obtain/understand their preferences, we will be more successful in our selling or communication proposition."

Therefore, we need to ascertain user preferences on the aspects relevant to our proposal (i.e.,, if we are trying to sell a Chinese cookery book, it is irrelevant to know the customer's hair color, but it is important to know that s/he is fond of cooking). Equally it is vitally important to know which communication channel the user is more receptive to. Adaptation involves a continuous process. Let us imagine for example that our objective is to paint a whole wall black. There is no such thing as "instantaneous wall painting" but rather it must be achieved one brush stroke at a time. In our case, the trivial data (i.e. name, address etc.,) are the equivalent of the brush strokes. When we use the word "personalization", the problem is that the verb "personalize" is like a kind of light switch, either it is on or off. Either you personalize or you do not. What really happens however is that there is a continuous process involved: It may be not be possible to personalize, it may be possible to personalize a little, it may be possible to have more or less accurate personalization, or have a completely tailor-made personalization. Clearly, "real" personalization is the latter of these possibilities, those really relevant to the user. Besides increasing the potential success of every subsequent interaction with our customer, there is another positive collateral effect arising from the use of personalization: After several relevant communications have been made, the customer/user pays more and more attention to our messages, because he has perceived them as relevant, unlike most of the communications they receive, where he/she perceives him/herself as an anonymous receiver. This precision is quite important, as we feel that there are too many unfounded claims of "delivers personalized results". In reality, the results may differ according to the true use of personalization in each scenario.

The recommender presented here is adapted to different systems and has been adapted with new features such as the one presented for the first time in this article (see sections 2.2, 2.3 and 2.4). Basically our proposal is a hybrid recommender that combines different features and separates long term and short term assumptions of the user model as presented in [17] [18]:

- Collaborative recommender (slope one)
- Content based recommender (WNBM, fingerprinting, PPM)
- Social recommender (Tags)

By having multiple sources for recommendations the cold start problem that appears in purely single source systems and especially in collaborative recommenders is avoided. This problem arises when a new item arrives, and no one has evaluated it, making it difficult to know how to recommend it. In our case the content based recommendation can be used initially in conjunction with the top relevance algorithm (see 2.2) and the Product Profile Matching approach (PPM) (see 2.3).

Summarizing, we have different recommenders: content based recommendations, a short term and a long term model, an item-item collaborative recommendation, one based on tags and the PPM. Each of these recommender approaches produces a list of programs and in order to calculate the relevance of each one for the user, we compute a weighted sum, where $\alpha+\beta+\varphi+\delta+\omega=1$. This determines the importance that we give to any of the four recommenders mentioned above (short and long are based on the same content based recomender). See Eq. (1). These parameters have an initial value that is updated for each recommender according to the amount of data available for it (i.e. if the number of user tags grows, this recommender will be given more importance). Besides automatic adjustment, the user can also express his/her preferences using the self fulfilling technique explained in 2.4.

$$R(user, item) = (\alpha R_{u,i}^{short}\ \beta R_{u,i}^{long})\varphi R_{u,i}^{collab.}\ \delta R_{u,i}^{tags}\ \omega R_{u,i}^{PPM} \qquad (1)$$

2.2 Top Relevance Algorithm

For the cold start scenario we propose different solutions. One of these is to propose items based on their relevance for the users. As users evaluate the items (i.e., the book, tv program, artist etc) the relevance must take into account the number of evaluations of this item(FO_i), the quality of its evaluations($Av(O_i)$), and the total number of evaluations input into the system($|evO|$). It is important to make a good comb ination of these factora because if not we may find situations like the following in some systems:

$$ranking_i = Av(O_i) \qquad (2)$$

At first glance, this approach may *seem* logical, as it means that an item will obtain its relevance according to the average of its evaluations. Let us suppose for the following examples that our items can be evaluated from (1 meaning very bad to 5 meaning very good). With eq. (2) there may be some strange results: if a document has been evaluated 100 times with an average of 4.2 it will be less relevant than one that has been evaluated only once with 5. This solution benefits newcomers and makes the top list very changeable and unstable.

$$ranking_i = Av(O_i) \times FO_i \qquad (3)$$

On the other hand we could take equation (3). This would give the older items a better position in the ranking because they have been evaluated many times, even if the evaluations were not particularly good. So, how can we obtain the right solution? We wish to give reflect an appropriate value for well evaluated newcomers but also respect those items evaluated many times. If we analyze the information retrieval experience, a similar problem arises when trying to rank documents according to a query. The algorithm TF-IDF [19] with all its variants [20] tries to solve a similar problem associated to terms in documents. IDF gives more importance to a term if it appears a few times in all documents (similar to our newcomers that have been evaluated several times) whereas TF increases the importance of a term if it appears many times on one document (similar to our many times evaluated items). Therefore, inspired by IDF, the first serious approach to our algorithm was:

$$ranking_i = \frac{\log_2 \frac{|evO|}{FO_i}}{|evO| + 1} \times FO_i \times \mathrm{Av(0i)} \qquad (4)$$

This equation, (4) works quite well but the logarithmic function gives much more priority to the newcomers, and if an item has been evaluated many times independently of its evaluations it becomes less relevant because the logarithm approaches zero (these are extreme cases), also the difference between different evaluations is not taken into account. Finally, inspired by the modification of IDF by

Joachim [21], where he states "The second difference is that the square root is used to dampen the effect of the document frequency instead of the logarithm", we changed the logarithm for a square root and squared the average evaluation in order to clarify the differences. The final equation is the following:

$$ranking_i = \frac{\sqrt{\frac{|evO|}{FO_i}}}{|evO| + 1} \times FO_i \times \mathrm{Av}(\mathrm{O_i})^2 \quad (5)$$

To clarify with a simplified example let us suppose there are 7 items that users can evaluate in the range 1-5. We have the average of their evaluations $\mathrm{Av}(\mathrm{O_i})$, the number of times the items have been evaluated $\mathrm{FO_i}$. and the total number of evaluations done in the system $|\mathrm{evO}|=\mathrm{Sum}(\mathrm{FO_i})$. In Table 1 we can find on the left how the algorithm (4) sorts the items and on the right how algorithm (5) does so. We can observe that the results on the left may not be entirely accurate because for example an item evaluated 100 times as 2 is ranked better than another that has been evaluated 10 times as 4. The square root of the equation (5) solves this problem and its ranking looks much more realistic. The equation (5) can be used with different goals: 1) To create top lists, i.e. for the *top list of favorites* (items are sorted because users have selected them as favorites) or the *popular items* . (sorts the items according to popular user evaluations). It could be used for example to obtain the most recent and popular selections 2) To tackle the Cold start problem. New users could obtain recommendations of the most popular items in the system as other personalized recommendations cannot be calculated yet or 3) To have initial estimations if the Personal and Explicit Profile (explained in the following section) has not yet been created.

Table 1. Comparison of the Ranking using a) the Logarithmic eq. (left) and b) the Square eq.(right)

$\mathrm{Av}(\mathrm{O_i})$	$\mathrm{FO_i}$	Log(eq.(4))	$\mathrm{Av}(\mathrm{O_i})$	$\mathrm{FO_i}$	Square(eq.(4))
5	40	2,21336768	5	40	10,5408205
4	40	1,77069415	5	20	7,45348565
5	20	1,55311241	4	40	6,74612512
2	100	1,03307474	4	10	3,37306256
4	10	0,79981639	5	3	2,88672258
5	3	0,41624581	2	100	2,66664009
2	10	0,3999082	2	10	0,84326564

2.3 Product Profile Matching

We understand PPM as a continuous process that involves the following elements:

A) A detailed User Explicit Profile (usually considered the user model), regarding the specific domain that in each system is being covered.

B) A product item, (which we can associate to something called the item model). This would involve the characteristics of the item relevant to the decision making process
C) A complete detailed model of the application of both group A and B, which could predict individual affinity between the specific user profile and the item model, not on a cluster basis but on an individual basis.

It is a continuous process because all three models are subject to continuous improvement, and a possible initial approach could yield some information helpful for improving every model. The key here is relevance. The criteria of inclusion /exclusion of attributes in these models is not to do with how easily they can be obtained but their relevance in the Product-Profile relationship. The design of both the attributes and the relationship must be done independently of the feasibility or any other factors that could hamper the creation of the best possible affinity mechanism model. Compromises can be made later but the model should take into consideration every single cause-effect that could influence the affinity model.

PPM involves a dedicated effort to create a taxonomy that must be addressed in a professional way, by people with knowledge of both business fields (User model – Item Model). Let us imagine a PPM model created for an online bookstore: There should be a clear customer expert behind the creation of the customer model, a librarian perhaps, and some kind of product manager behind the creation of the product mode. The combination of these, perhaps someone from a commercial department, should be behind the affinity model.

Product characterization does not need to be extensive if the relevance prerequisite previously mentioned is fulfilled – the authors have produced a paper on a process of PPM from the Product side [22] in which there is considerable compacting of an exhaustive product characterization (TV content, made by Anytime TV) into a more compact, easy to manage form and they present a taxonomy which would be a perfect product model for a PPM scenario, along with a complete TV user model and a complete Affinity model (More on this in [17] [18]).

How does PPM relate to balanced recommenders? A Proper PPM schema should be included in balanced recommenders for the following reasons:

- It must deal with the “individual” aspects of the recommendation, like the rest of the aforementioned techniques discussed previously.
- It provides a strong initial starting point thereby avoiding the Cold Start scenario (working in conjunction with the aforementioned solutions), translating the responsibility of preventing the cold start to the user providing detailed info on his model (as the product model has been previously covered, as well as the affinity application model).
- It offers a strong model to refine the overall recommender results.

2.4 Self Fulfilling Capabilities

Another key component of a Balanced Recommender system is the existence of Self fulfilling capabilities and a proper Self Fulfilling strategy must be in place. Let us try to develop this. Most recommenders have adopted an approach that we strongly discourage – that of keeping the user away from the underlying algorithm used. This is like telling users “Trust us, we are really smart” – not the best approach for a supposedly "personalized" approach. Users do not have a real sense of being in control and we think that it is important to allow people to decide to what degree outside or collective intelligence should play a part in the suggestion / decision provided by the recommender. As our current proposal involves several different features and it is highly likely that different users will have different degrees of collaboration, the following steps should be taken:

- A step-by-step system should be developed to educate the customer on how to move from a fully automated to a fully (user) controlled contribution for every factor
- The degree of precision in the recommendation should be directly linked to the following factors:
  - How well the user understands the underlying model and how this affects their input and fine tuning,
  - The degree of fulfillment of the data user model proposed, through a clear "tit for tat" proposition – you provide me with better data and a better recommendation will be the result.
  - The degree of precision must not be related to external factors such as intrinsic data quality or the degree of training of the recommender network
- All the contributions involved in the recommendation algorithm should be shown for the customer to fine tune and adjust their preferences once they have received appropriate training on the matter: i.e., they should be informed of how much weight was given to content based evaluations, how much to collaborative, how much to PPM etc…

With some kind of visual metaphor and some easy feedback procedures we are sure that people will have a much better experience with recommenders than has been the case up to now (see Fig. 2).

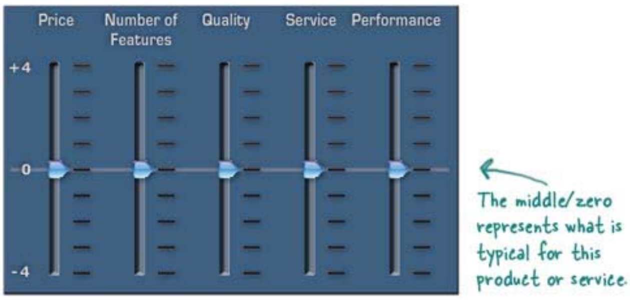

Fig. 2. The user can adjust the recommender parameters

3 Mirotele: A Balanced Recommender at work

Mirotele[2] is a joint venture between the authors in which we have been involved [17] [22] [18] for some time and we are using it to test all our theories at the moment. We have created a whole User model (representing what we consider to be the

[2] http://www.mirotele.com

relevant attributes involved in TV preferences), a whole self fulfilling environment (where advanced users can start to tinker with the data involved in the algorithm once they begin to appreciate how it works). We have developed a whole PPM schema - combining the aforementioned user model with a fingerprinting method for TV Programs [22]. In addition, we incorporate all the improvements made to the existing algorithms mentioned previously. There is as well a whole social networking schema, a cross between a wiki and a folksonomy approach, where the social network is used not go generate, but to filter and gather a huge amount of content related information that has been produced via an automatic information gathering and classification system (using web services from Google, YouTube, and other TV related services). Unfortunately we have not yet collected data on a real implementation for the current version of this paper (April 2009).

4 Conclusions and Future Work

Our current conclusions are basically the following:

- Although our Balanced Recommenders incorporate a hybrid approach, they are not the same as hybrid recommenders. This is not merely a question of semantics. We are mixing personalization techniques and classic recommender techniques. In our recommenders, the knowledge domains are quite different and the result is much more intuitive than in the hybrid approach. Personalization is a science in itself and trivial approaches must be avoided. We have found several personalization techniques like PPM that are appropriate for expanding the current recommender schema. We do not discard the possibility of enriching our schema with other techniques in the future, and have found that "balancing" a purely scientific approach with personalization techniques has produced an extremely good/promising result.
- In the future there will be a systematic shift from current recommender schemas to balanced approaches like the one we present here.
- We foresee a new golden age in the use of recommenders systems as they gradually become important information-organizers, substituting those currently in existence (mostly Search engines).

We are currently working on the implementations of our schemas and algorithms and plan to continue researching the area of balanced recommenders, in particular dealing with the less documented and structured aspects of personalization techniques. At the same time we will continue to improve our tools and attempt to determine as much as possible the correct combination of every factor considered here in order to achieve "the perfect recommendation". Perhaps it is as elusive as the perfect cocktail, but our ultimate goal is to improve current.

5 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," Knowledge and Data Engineering, IEEE Transactions on, vol. 17, no. 6, pp. 734-749, 2005.

2. D. Billsus and M. Pazzani, "A Hybrid User Model for News Story Classification," Banff, Canada: 1999.
3. N. Good, Schafer J., J. Konstan, A. Borchers, B. Sarwar, J. Herlocker, and J. Riedl, "Combining collaborative filtering with personal agents for better recommendations," 1999.
4. P. Cotter and B. Smyth, "WAPing the Web: Content Personalisation for WAP-Enabled Devices," Springer-Verlag, 2000.
5. H. Cramer, V. Evers, S. Ramlal, M. van Someren, L. Rutledge, N. Stash, L. Aroyo, and B. Wielinga, "The effects of transparency on trust in and acceptance of a content-based art recommender," User Modeling and User-Adapted Interaction, vol. 18, no. 5, pp. 455-496, Nov.2008.
6. M. Balabanovic and Y. Shoham, "Combining Content-Based and Collaborative Recommendation," Communications of the ACM, vol. 40, no. 3 1997.
7. D. Bueno and A. A. David, "METIORE: A personalized information retrieval system," User Modeling 2001, Proceedings, vol. 2109, pp. 168-177, 2001.
8. M. Pogacnik, J. Tasic, M. Meza, and A. Kosir, "Personal Content Recommender Based on a Hierarchical User Model for the Selection of TV Programmes," User Modeling and User-Adapted Interaction, vol. 15, no. 5, pp. 425-457, Nov.2005.
9. G. Linden, B. Smith, and J. York, "Amazon.com recommendations: item-to-item collaborative filtering," IEEE Internet Computing, vol. 7, no. 1, pp. 76-80, 2003.
10. R. Burke, "Hybrid Recommender Systems: Survey and Experiments," User Modeling and User-Adapted Interaction, vol. 12, no. 4, pp. 331-370, Nov.2002.
11. L. Candillier, K. Jack, F. Fessant, and F. Meyer, "State-of-the-Art Recommender Systems," in Collaborative and Social Information Retrieval and Access: Techniques for Improved User Modeling. M. U. o. T. Chevalier, Ed. 2009, pp. 1-22.
12. A. Trotman, "Learning to Rank," Information Retrieval, vol. 8, no. 3, pp. 359-381, Jan.2005.
13. S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web search engine," Computer Networks and ISDN Systems, vol. 30, no. 1-7, pp. 107-117, Apr.1998.
14. Klaus, "Algoritmo de Popularidad Actualizado," in http://blogs.gamefilia.com/markus/10-02-2008/854/algoritmo-de-popularidad-actualizado.
15. G. Asela and M. Christopher, "Tied boltzmann machines for cold start recommendations," in Proceedings of the 2008 ACM conference on Recommender systems Lausanne, Switzerland: ACM, 2008, pp. 19-26.
16. D. Bueno, "Recomendación Personalizada de documentos en sistemas de recuperación de la información basada en objetivos." Ph.D. Universidad de Málaga, 2003.
17. D. Bueno, R. Conejo, and J. G. Recuenco, "An architecture for a TV Recommender System," International Workshop on Personalization in iTV.Euro ITV 2007 Amsterdam: 2007, pp. 117-122.
18. D. Bueno, R. Conejo, D. Martín, J. León, and J. Recuenco, "What Can I Watch on TV Tonight?," 2008, pp. 271-274.
19. J. Rocchio, "Relevance Feedback in Information Retrieval," The SMART Retrieval System: Experiments in Automatic Document Processing, pp. 313-323, 1971.
20. S. Robertson, "Understanding inverse document frequency: on theoretical arguments for IDF," Journal of Documentation, vol. 60, no. 5, pp. 503-520, 2004.
21. T. Joachims, "A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization," 1997, pp. 143-151.
22. J. Recuenco, N. Rojo, and D. Bueno, "A New Approach for a Lightweight Multidimensional TV Content Taxonomy: TV Content Fingerprinting," in Changing Television Environments 2008, pp. 107-111.