



# PREFERENTIAL ATTACHMENT, AGING AND WEIGHTS IN RECOMMENDATION SYSTEMS

MASSIMILIANO ZANIN, PEDRO CANO\*  
and OSCAR CELMA

*Music Technology Group, Universitat Pompeu Fabra,  
08003 Barcelona, Spain*

*\*BMAT, Barcelona Music and Audio Technologies,  
Llacuna 162, 08018 Barcelona, Spain*

JAVIER M. BULDÚ

*Complex Systems Group, Universidad Rey Juan Carlos,  
Tulipán s/n, 28933 Móstoles, Madrid, Spain*

Received March 14, 2008; Revised May 26, 2008

In the present work, algorithms based on complex network theory are applied to *Recommendation Systems* in order to improve their quality of predictions. We show how some networks are grown under the influence of trendy forces, and how this can be used to enhance the results of a recommendation system, i.e. increase their percentage of right predictions. After defining a base algorithm, we create recommendation networks which are based on a histogram of user ratings, using therefore an underlying principle of *preferential attachment*. We show the influence of data aging in the prediction of user habits and how the exact moment of the prediction influences the recommendation. Finally, we design weighted networks that take into account the age of the information used to generate the links. In this way, we obtain a better approximation to evaluate the users' tastes.

*Keywords:* Complex networks; recommendation algorithms; aging; music networks.

## 1. Introduction

Since the experiment of Milgram [1967], the study of (social) networks has attracted the interest of many scientists from completely different fields. Boosted by the seminal paper of Watts and Strogatz [1998], complex networks theory has become a strong utility to analyze different kinds of data structures [Boccaletti *et al.*, 2006]. The application of complex networks to social problems has generated special interest, and it has given fruitful results in different subjects, ranging from sexual disease control [May & Lloyd, 2001; Pastor-Satorras & Vespignani, 2002] to music community identification [Lambiotte & Ausloos, 2006; Park *et al.*, 2007]. Another field

where complex networks knowledge is leveraged is the design of *Recommendation Systems*. In the last years, developments in computer and information technologies have created new channels of commerce, mainly electronic, where millions of customers are served each day, generating an enormous quantity of information about their habits. On the other hand, this innovation has created the need for personalization in customer care, and this has led to a great interest in generating algorithms that recommend items to users entering an “e-store”.

In the search for better recommendation algorithms using complex networks theory, properties of the system like Clustering Coefficient [Huang, 2006] or Jaccard's Coefficient [Huang *et al.*, 2005]

have been explored, showing different results. When the growth of the recommendation system is considered, the *Preferential Attachment* strategy has been recently proposed [Huang et al., 2005], but without much consideration within music technology community.

In this paper, we wish to go deeper into the idea of applying preferential attachment to a recommendation system: after defining a base algorithm, we study the effect of time in the network evolution, and find a better approximation to evaluate the users' tastes.

## 2. Preparing the Ground

The item-based strategy [Sarwar et al., 2001; Wang et al., 2006] is one of the most popular in recommendation systems: it presents interesting advantages, like short computation time and low sensitivity to network sparsity. Since it is a very extended method for recommendation, we will choose this algorithm as the ground to compare with any other results.

The basic idea behind an item-based strategy is to look into the set of items related with the target user, to compute the similarity of these items with others in the network, and select the most similar (see [Sarwar et al., 2001] for details). For this purpose, a *cosine-based similarity* is commonly used. For each item, a vector of length  $N$  is created,  $N$  being the total number of users. The vector accounts for the relation between items thanks to user choices: for example, if the  $n$ th element of the vector has a value of 1, it means that the user

number  $n$  has selected that item (or 0 otherwise). In some datasets, moreover, each element can represent the rating of a given user for an item: e.g. a value between 1 and 5. After creating those vectors, the similarity between two items  $i$  and  $j$  is defined as:

$$\text{sim}(i, j) = \cos(\mathbf{i}, \mathbf{j}) = \frac{\mathbf{i} \cdot \mathbf{j}}{|\mathbf{i}| \cdot |\mathbf{j}|}. \quad (1)$$

In this paper, we will only use this measure of similarity, for being well-known and easy to implement; nevertheless, other ways to calculate this characteristic have been developed in the past: the Correlation-Based Similarity (by computing the *Pearson-r* correlation) and the Adjusted Cosine Similarity [Sarwar et al., 2001].

We have used two datasets in our experimental study, each one with different characteristics, to analyze different backgrounds and compare results. The first dataset is the collection of ratings of Netflix [Netflix], a web page of movie renting where users can also evaluate movies (from 1 to 5). In order to work with a network of simple (unweighted) connections, we filter ratings different from 5 (the highest mark), so that we only keep users connected with their top-rated movies. The result is a set of 17 770 items (movies), 2.6 millions users and more than 23 millions of operations (links).

The second dataset is obtained from *Art Of The Mix* [AOM]. In this network, we have 90 000 users, 472 000 items (songs, in this case) and 1.3 million links. The Art Of The Mix is a project started at the end of 1997 and consists of a web site where users upload and interchange playlists of their favorite music. The songs of a playlist, somehow, fit in those

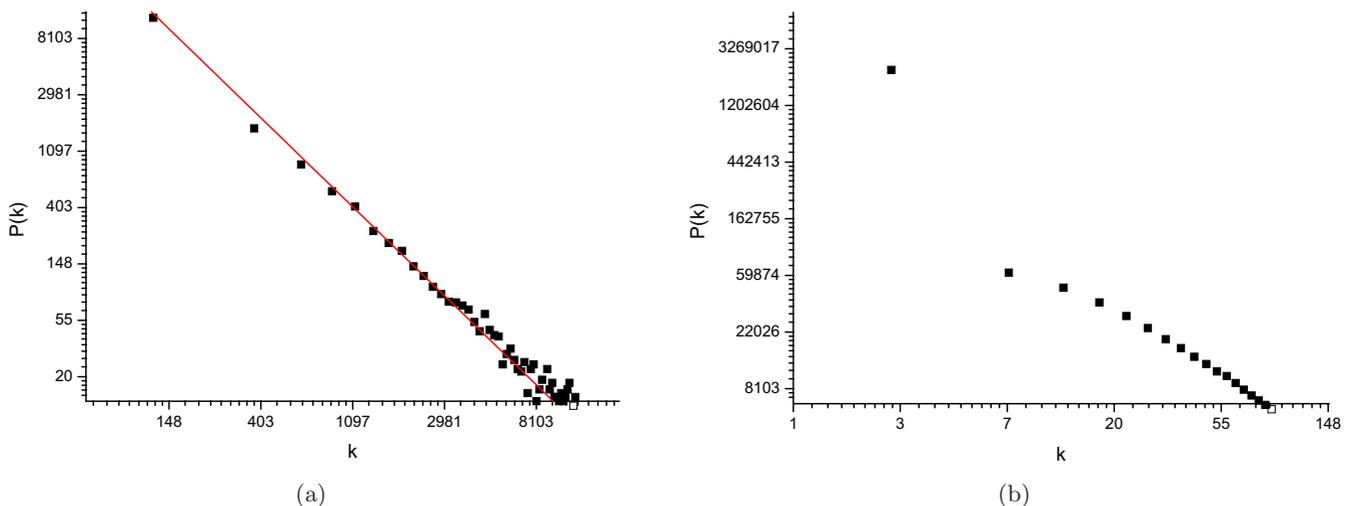


Fig. 1. Degree distribution for (a) items (i.e. movies) and (b) users in the NetFlix dataset.

lists, even though they do not need to belong to the same country, decade or musical genre. In this way, a certain connection results between songs of a certain list, whose origin is based on the musical taste of the playlist’s author. Both datasets share the same structure: each line of the network file includes a connection between a user (specifically, the ID of the user) and an item (again, an anonymous ID defining the item), and also the timestamp of the connection.

Once networks are defined, it is worth noting that the size of the present datasets is much higher than previous results in other networks, like [Huang *et al.*, 2005a], where 10 000 items and 2000 users were considered, or [Huang, 2006] with a dataset close to 40 000 items.

Concerning the network structure, we can observe in Fig. 1 that the degree distribution (i.e., number of connections) of both users and items have a power law decay at their tails, a fact that indicates that both networks are scale-free [Barabási & Albert, 2000].

### 3. Preferential Attachment

The initial step to improve a recommendation algorithm by taking advantage of complex networks theory is to use the concept of preferential attachment. First introduced by Barabási and Albert [1999], the preferential attachment has become a paradigmatic growth algorithm in order to explain the structures and evolution of social networks.

The main idea in [Barabási & Albert, 1999] is that nodes with higher degrees (i.e. with more links) acquire new links at higher rates than low-degree nodes; the probability that a link will connect a new node  $j$  with another existing node  $i$  is linearly proportional to the actual degree of  $i$ :

$$p(j \rightarrow i) = \frac{k_i}{\sum_{j=1}^N k_j} \quad (2)$$

where  $k_i$  is the degree of node  $i$  and  $N$  is the total number of nodes. When defining a recommendation algorithm, this is equivalent to suppose that a given user has a higher probability of selecting a popular item than an unknown one. Intuitively, it may be clear that in some cases it will be right: every time the algorithm is applied to a selling system, where goods being sold depend on trendiness, items that are well-known will have a higher probability of being bought. Nevertheless, there can exist cases

where the popularity of an item, or the existence of a certain fashion, does not affect the creation of new links, and users make their choices only following personal criteria.

As we will see, both considerations should be taken into account and some kind of balance between them should also be included. Another interesting point is that the initial dataset consists of a bipartite network [Newman, 2003] with two different kind of nodes, users (people) or items (movies/songs). The bipartite network could be projected in two different networks; one with users being the fundamental nodes and other with movies/songs being the nodes. Nevertheless, both projected networks disregard part of the information when they are considered independently and we should define a way of accounting for all the information within the dataset.

At this point, let us explain the way of implementing a preferential attachment strategy in our recommendation algorithm, i.e. an algorithm that favors the recommendation of the most connected items. The procedure can be summarized in four steps:

- First, we define a distance between a target user and any other user. As in the case of items, a vector is created for each user, accounting for his/her selected items. The vector has length  $M$  which corresponds to the total number of items, and it will have a value of 1 at position  $m$  if the  $m$ th item has been chosen by the user. Next, the *cosine-distance*  $\text{dis}(j)$  with respect to the target user is calculated, and values are stored in a linear array:

$$\text{dis}(j) = \cos(\mathbf{i}, \mathbf{j}) = \frac{\mathbf{i} \cdot \mathbf{j}}{|\mathbf{i}| \cdot |\mathbf{j}|} \quad (3)$$

where  $i$  is the target user, and  $j$  is the other user of the network. As before, other measures can be chosen, but this particular one has been selected for simplicity.

- For each item  $l$  of the network, we define a *compatibility* value  $\text{comp}(l, \text{user})$  between an item and the target user, which is calculated as the sum of the *closeness* of users related with that item; *closeness* is defined as  $1 - \text{dis}$ :

$$\text{comp}(l, \text{user}) = \sum_j (1 - \text{dis}(j)) \quad (4)$$

where  $l$  is the item, and  $j$  accounts for users that have connections with  $l$ .

- Finally, items are ordered according to their *compatibility*, in descending order. Items in the

beginning of the list are the more compatible, i.e. more suitable for recommendation. In this way, items at the top of the list are the best for the target user, and should be submitted to his/her attention.

Note that within this context *compatibility* is related to how good an item is for a target user according to his/her tastes and to the network created with other users. The value of *compatibility* is bounded in the range  $0 \leq \text{comp} \leq \infty$ . As we are summing up  $(1 - \text{dis})$  other users connected with that item, the result considers the size of the community of users with the same tastes (and, of course, users who have bought the target item). The bigger this community is, the higher the similarity between our user and the community, which leads to an increase of the *compatibility*.

Two important features of this approach need to be explained in detail. First of all, this scheme has a very small calculation time; the most expensive operation, i.e. the calculation of distance between users, is executed only one time: this leads to a complexity function of  $O(m)$ , where  $m$  is the number of users. On the contrary, for the basic item-based scheme, the algorithm should calculate the compatibility between an item and each one of the items connected to the target user. This is equivalent to carry out this calculation  $u$  times, where  $u$  is the number of items related with the target user; or, in other words,  $O(l \cdot u)$ , with  $l$  being the number of items. As a result, the computational cost of the basic algorithm is up to 100 times worse for the NetFlix dataset: of course, this point is an important feature when working with large datasets and real-time recommendations.

Second, unlike the basic algorithm, now we see that the global *score* (the measure of the quality of the recommendation) of an item depends on how many users have a connection with it: for each one of these connections, its compatibility value (i.e. the compatibility between the selected and the target users) is summed up, and the result of the sum is the global compatibility of that item. This means that an item with many links will have a higher compatibility value than another item with only a few links (because of the higher quantity of values summed up), which is the basis of preferential attachment: the more connections, the more the probability of being chosen by another user. On the other hand, not only the number of links is considered if an item is well-known, but it is far from the tastes of the

target user, its total compatibility will be small, and that item will not be recommended.

## 4. Aging Effect

### 4.1. Trendiness in real networks

As explained before, it makes sense that preferential attachment may improve the quality of recommendations when the underlying network has a strong trendiness component, where trendiness is the preference of a user for items with high popularity: in the case of customer datasets, buying items in an e-store, as the NetFlix dataset, the more an item is known, the more it is likely that the item will be chosen by the target user. Up to now, all data previous to the prediction date have been considered. Except in some works like [Song *et al.*, 2006] or [Herlocker *et al.*, 2004], this has been the traditional approach, since it is a generalized opinion that the more data used in calculation, the better is the result. Nevertheless, trendiness of an item greatly depends on time: one item can have a high popularity at a time  $t_0$ , but it can lose all interest after a certain time  $t_1$ . This fact is schematically plotted in Fig. 2. The left plot shows a hypothetical evolution of the number of new links of an item  $A$  (i.e. the derivative of its instantaneous degree): at time  $t_0$  this item has a high probability (i.e. considerable popularity in a given moment, with many new users connecting to it), while close to  $t_1$  its number of new links decreases. On the other hand, item  $B$  has an overall lower degree, but a greater instantaneous degree close to time  $t_1$ . It is important to note, that item  $A$  has a higher number of connections if we consider the global data, while  $B$  wins in instantaneous degree after time  $t_1$ . A simple recommendation algorithm, like the one exposed before, would consider all data of the network, resulting in a greater probability for item  $A$ ; nevertheless, if we want a *real-time* suggestion, e.g. just after  $t_1$ , the recommendation algorithm should be advantageous to  $B$ .

The example above explains the importance of the link aging: when the global network is used in calculations, many data that are not strictly necessary are included; sometimes, that unwanted data can lead to mistakes, and in addition they always increase the calculation time.

In Fig. 3 we represent how the instantaneous degree of all items which takes into account the number of new links per day. We can see in the inset of Fig. 3(a) an example of the instantaneous degree

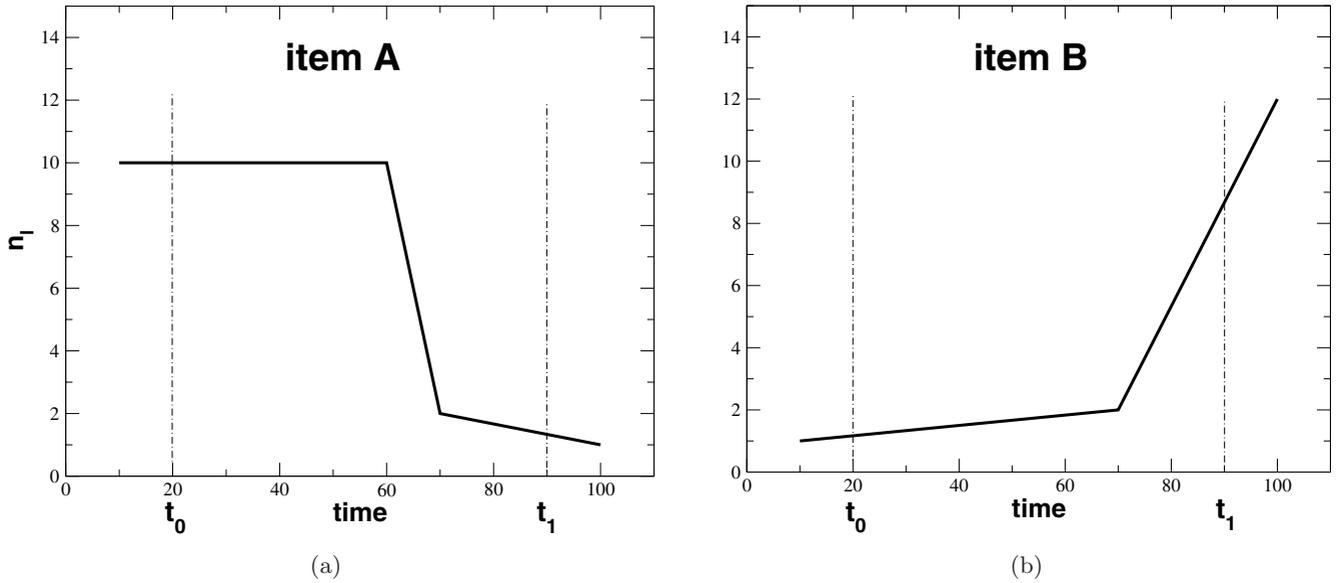


Fig. 2. Qualitative example of the degree evolution for two items, with the ordinate axis indicating new links  $n_i$  acquired at a given time. If a recommendation would be done at  $t_1$ , item on the left (A) has a higher global degree (which would correspond to the area of the  $n_i = f(t)$  function), while item B has a higher acquisition rate.

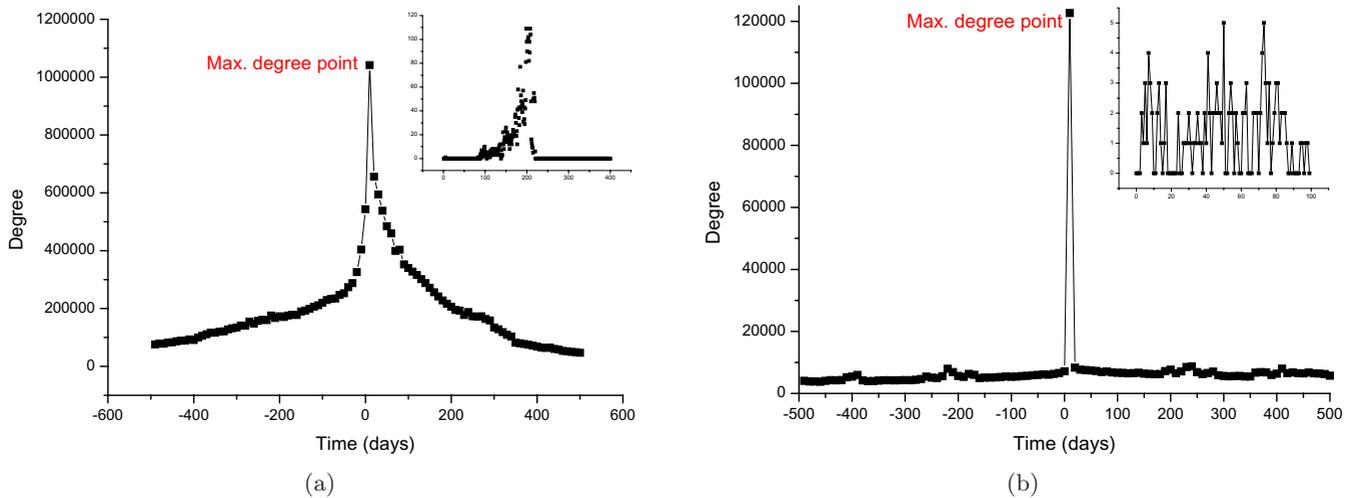


Fig. 3. Global degree evolution for (a) NetFlix and (b) Art Of The Mix networks: the central point represents the moment of greatest degree of every item. In the insets, we plot the instantaneous degree of an example item for each network; note that in (b), the degree does not show a clear peak: the mean degree evolution for that network is therefore flat.

evolution for a given item. In order to account for the whole dataset, we add the instantaneous degree of all items, which are aligned at their absolute maxima. Figure 3 shows the results for both networks. For the NetFlix dataset, a great peak is observed, with the degree value increasing and decreasing monotonically around the central point: from the aging point of view, it means that, first, there is a certain correlation time in the process of achieving the highest popularity, and second, popularity depends on time, and therefore, we must take it

into account at the moment of recommendation of an item.

The opposite case is Art Of The Mix, where the level of the instantaneous degree for the whole dataset is nearly constant, with only a central delta-shaped peak. In fact, the central peak is an artifact: since we align all items at their absolute maxima, we will always have the highest value at time zero. Nevertheless, the flat spectrum of the rest of the series indicates that fluctuations of the instantaneous degree are filtered when considering all items.

The absence of correlation in the degree evolution indicates that the relation between users and items does not depend on time, as we do not observe a transaction-like structure, and that trendiness is not fundamental in order to explain the network growth: aging should not help in improving results.

#### 4.2. The cut-off time

Starting from the above considerations, we define an improvement of the basic preferential attachment algorithm: before calculating the result, the network is filtered to include only data (i.e. links) enclosed in a time window. We assign a cut-off time  $d$  to the window, and for a given time  $t_1$  and a target item, only links within the window  $t_1$  and  $t_1 - d$  are considered.

Results of applying aging-based filtering to both networks are shown in Figs. 4 and 5 (NetFlix), and 6 (Art Of The Mix). In order to evaluate the recommendation algorithm we compute the *score* of the predictions, which will be explained in detail in the next section. For the time being, the score must be taken as an indicator of the quality of the recommendation. As expected, thanks to the strong trendiness in the NetFlix dataset, the cut-off dimension of the window results in an improved score. Obviously, when the window is too small, there is not enough information to perform a good recommendation and the score decreases. On the other hand, when applying an aging filtering to

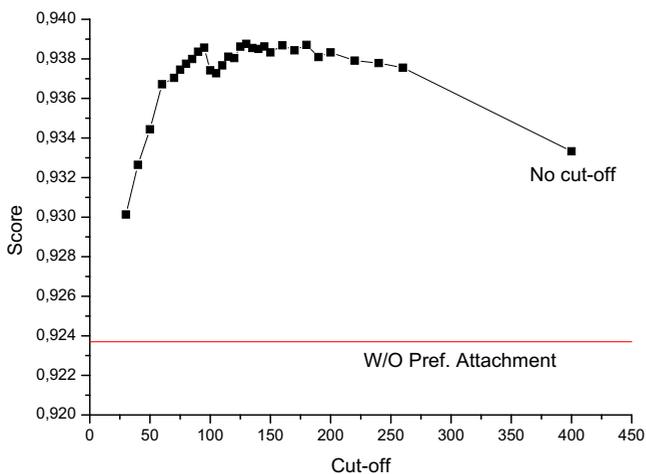


Fig. 4. Recommendation score as a function of cut-off window dimension  $d$ , for NetFlix dataset. The horizontal line represents the score of the basic item-based algorithm, while the right point, marked with no *cut-off*, is the result of using the preferential attachment algorithm without filtering data (as if  $d = \infty$ ).

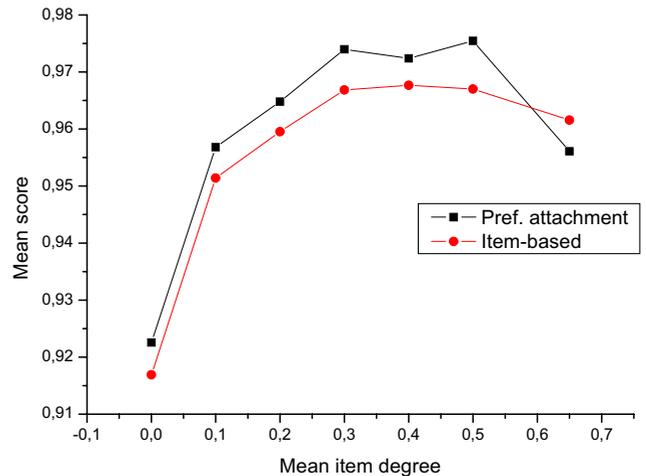


Fig. 5. Mean score obtained as a function of the mean degree of the movies chosen by a user, for the standard and the preferential attachment algorithm.

the Art Of The Mix network we do not obtain an improvement of the score (see Fig. 6): as there is no correlation in the evolution of the instantaneous degree, the reduction of the time window excludes important data from the analysis, and therefore the score decreases.

When network growth is based on rules that are equivalent to preferential attachment, an important improvement in recommendation results can be achieved; we go from the 0.924 of the item-based algorithm, to 0.933 of the preferential attachment algorithm without aging, and finally to 0.939 when link aging is considered. Although, according to the score, this improvement is only of 1.5%, it has great relevance from a commercial perspective: in fact,

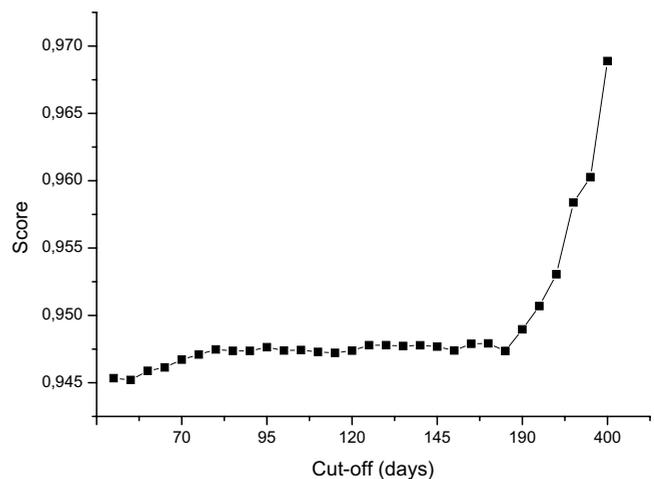


Fig. 6. Recommendation score as a function of cut-off window dimension  $d$ , for Art Of The Mix dataset.

due to the great quantity of items in the dataset, small variations in the score lead to significant variation in the user perception of the effectiveness of the algorithm. At the same time, calculation time has been optimized: when window size is small, there is less information to be processed and the recommendation speeds up. For example, for a time window of 120 days, the mean number of transactions that has to be analyzed is around 25% of the whole dataset, which means a computational cost reduction of more than five times. Moreover, we have previously seen how a user-based strategy is more efficient than an item-based one: once again, the speed of the new algorithm is more suitable for real-time implementations.

### 4.3. Score calculation

In the previous section, we have used a *score* value to compare results coming from different algorithms: it is time to explain how it is calculated, and moreover, why we have used this strategy.

When we evaluate a recommendation system, we randomly choose a target user and a target item already selected by this user: that item should be recommended by the algorithm for the given user, using only data prior to link date and time. No restriction is applied to links position: it can be at the beginning of the dataset (thus, only a few data can be used), or it can be at the end (improving the amount of information available, but also increasing the computational cost). The recommendation algorithm returns a list of items, ordered by compatibility, so that the items on the top of the list should be the best for the target user.

The *Score* value is simply calculated as a function of the position of the target item in that list:

$$\text{Score} = 1 - \frac{\text{Pos}_{\text{item}}}{\#\text{items}}$$

The more the target item is in the upper part of the recommendation list, the more *score* approximates to 1.

In the past, other algorithms have been defined to check the performance of recommendation algorithms, and some of them (e.g. MEA, RMSE or Precision/Recall/F-measure [Herlocker *et al.*, 2004]) are often taken as standard measures. As an example, in [Huang *et al.*, 2005] a great part of the dataset is used for training the system, while the last part is the testing period; using data of the first set, the algorithm should generate a ranked

list of recommendations for each user, and the quality of the recommendation system is then measured using the number of *hits* and their position in the ranked list. This method of evaluation is not suitable when preferential attachment is used, and even more when an aging effect is applied, due to the fact that time has a great influence in calculations. When we choose a time  $t_0$  and a given user for evaluating the recommendation, all data related with item's rank depend on  $t_0$ . If an item  $i$  is a *hit* at a distant time  $t_1$ , let us say  $t_1 \gg t_0$ , we should disregard that result.

## 5. Links Weight

Finally, let us mention some details about the link heterogeneity. When defining recommendation algorithms, links between users and items are normally identical, and the network is defined as unweighted. In our case, we have a parameter that can be used to discriminate the importance of each connection: the age of that link.

For a given link, we can assign a weight that is defined as a function  $W$  of the number of days passed since its creation. Although any function can be used for this purpose, we have chosen a piecewise linear function, that can be tuned by two parameters  $\alpha$  and  $\beta$ :

$$W(i) = \begin{cases} 1, & a_i > \beta \\ 1 + \frac{\beta - a_i}{\beta}\alpha, & a_i \leq \beta \end{cases} \quad (5)$$

where  $a_i$  is the age of the link. In this way, we modify the compatibility of a given item  $l$ , which now reads:

$$\text{comp}(l) = \sum_j (1 - \text{dis}(j))W(j \rightarrow l) \quad (6)$$

where  $(j \rightarrow l)$  is the link connecting user  $j$  to item  $l$ .

The parameter  $\alpha$  represents the relative importance of recent links with respect to older ones; according to Eq. (5), a link that is created the same day (i.e.  $a_i = 0$ ) would have a weight of  $W(i) = 1 + \alpha$ , while an old link would have  $W(i) = 1$ . The value of  $W(i)$  decreases linearly with time from  $W(i) = 1 + \alpha$  when  $a_i = 0$ , to  $W(i) = 1$  when  $a_i = \beta$ . Therefore,  $\beta$  is the maximum number of days for an item to be weighted more than the unit.

The obtained *score* for different values of  $\alpha$  and  $\beta$  on the NetFlix data collection is shown in Fig. 7. A maximum is detected around  $\beta = 20$  for different

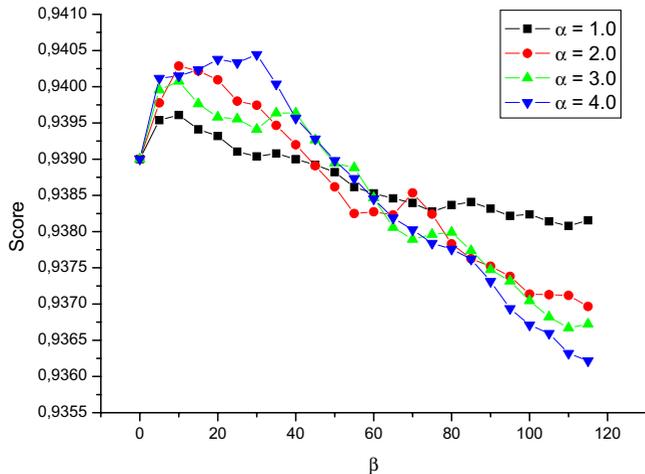


Fig. 7. Effect of considering weighted links. Results refer to the NetFlix dataset for a window dimension of 120 days.

$\alpha$ , while large values of  $\beta$  lead to a reduction of the *score*. This behavior is expected since high values of  $\beta$  are equivalent to increasing the importance of old

links, a fact that is not favorable for a preferential attachment strategy. On the other hand, low values of  $\beta$  are equivalent to include only very young links, excluding a great quantity of information, and decreasing the *score* value.

## 6. Let Us Recommend

In order to better explain how preferential attachment algorithm works, we report an example of recommendation for the NetFlix dataset. The target user, randomly chosen, is the user number 658088, and the item that is going to be chosen at the next time step is item number 872 (for privacy). Target user has links with 24 other items at the moment of the recommendation.

First, we calculate the *score* using the basic algorithm, which only considers the cosine-distances between items. After computing the ranking, ordered by compatibility, we obtain the following items in the first positions:

<b>Item</b>	(1 <sup>st</sup> ) 7843	(2 <sup>nd</sup> ) 5085	(3 <sup>rd</sup> ) 11 038	(4 <sup>th</sup> ) 14 241
<b>Compatibility</b>	0.16734	0.14864	0.14591	0.14381

Target item is in position 830, with a compatibility of 0.04993: that is, we get a *score* of 0.95329 ( $Score = 1 - 830/17770$ , where 17 770 is the total number of items) for this case.

Next step is executing the preferential attachment algorithm with aging on the same user and item. The dimension of the window  $d$  used for data filtering can have different values, and for each value the results obtained (i.e. number of connections of

the target user, rankings, *score*) are different. To show an example, we set the time window to  $d = 70$  days. In this case, after filtering the dataset, we have only 2.26 millions operations (about ten times less than the original data), and the target user has three more links to other items. Target item 872 is connected with 198 users in that interval of time, and their compatibility ratings of the target user are the following:

<b>User</b>	(1 <sup>st</sup> ) 698 478	(2 <sup>nd</sup> ) 2 081 171	(3 <sup>rd</sup> ) 1 558 760	...
<b>Compatibility</b>	0.04352	0.06337	0.05803	...

Summing up all 198 values, we obtain a total compatibility of 14.69987, which is greater than the one obtained with the basic algorithm: this is because we are summing up hundreds of values, so the system must work with wider ranges. For this value of  $d$ , the ranking obtained starts with the following values:

<b>Item</b>	(1 <sup>st</sup> ) 13 728	(2 <sup>nd</sup> ) 14 240	(3 <sup>rd</sup> ) 2782	(4 <sup>th</sup> ) 11 521
<b>Compatibility</b>	756.14	165.59	160.98	146.96

Target item is at position 357, that represent a *score* of 0.9799: comparing the result of the item-based algorithm, target item climbed 515 positions.

Scores obtained with different values of  $d$  are shown below:

$d$	30	50	100	140	180	$\infty$
Score	0.87530	0.97794	0.97766	0.97535	0.9740	0.97840

Although in this example a higher value of  $d$  (i.e. the dimension of the cut-off window) is related with a higher score value, this is just a single case: for the whole dataset, the best value of  $d$  is the one shown in Fig. 4. That value of  $d$  is the mean result of the dataset, which is not excluded from stochastic fluctuations. The forces inducing these variations and the connection with some user characteristics still remain unclear, and will be the subject of future works.

## 7. Conclusions

In recommendation systems, it is a common opinion that the larger the dataset, the better the result will be. In this paper, we show that in certain cases this reasoning is not true. When recommendation systems refer to networks with strong trendiness component, a preferential attachment strategy can improve results, while at the same time, smaller computational cost is required. This fact is due to the aging of the existing information, which can be crucial in certain kind of networks. We demonstrate that, when fashion or trends are present in the evolution of a given network, the age of the links must be taken into account when developing a recommendation algorithm. Moreover, we have seen that weighted links, based on its age, are suitable for discriminating between recent and old information, increasing the quality of the prediction in trendiness networks.

## References

AOM: <http://www.artofthemix.org>  
 Barabási, A.-L. & Albert, R. [1999] “Emergence of scaling in random networks,” *Science* **286**, 509–572.  
 Boccaletti, S., Latora, V., Moreno, Y., Chavez, M. & Hwang, D.-U. [2006] “Complex networks: Structure and dynamics,” *Phys. Rep.* **424**, 175–308.  
 Herlocker, J. L., Konstan, J. A., Terveen, L. G. & Riedl, J. T. [2004] “Evaluating collaborative filtering recommender systems,” *ACM Trans. Inform. Syst. (TOIS)*.

Huang, Z. [2006] “Link prediction based on graph topology: The predictive value of the generalized clustering coefficient,” *LinkKDD*.  
 Huang, Z., Li, X. & Chen, H. [2005a] “Link prediction approach to collaborative filtering,” *Proc. 5th ACM/IEEE-CS Joint Conf. Digital Libraries*, pp. 141–142.  
 Huang, Z., Zeng, D. D. & Chen, H. [2005b] “Analyzing consumer-product graphs: Empirical findings and applications in recommender systems,” *SSRN*, September (2005).  
 Lambiotte, R. & Ausloos, R. [2006] “On the genre-fication of music: A percolation approach,” *Eur. Phys. J. B* **50**, 183–188.  
 May, R. M. & Lloyd, A. L. [2001] “Infection dynamics on scale-free networks,” *Phys. Rev. E* **64**, 066112.  
 Milgram, S. [1967] “The small world problem,” *Psychol. Today* **1**, 61–67.  
 Netflix: <http://www.netflix.com>  
 Newman, M. E. J. [2003] “The structure and function of complex networks,” *SIAM Rev.* **45**, 67–256.  
 Park, J., Celma, O., Koppenberger, M., Cano, P. & Buldú, J. M. [2007] “The social network of contemporary popular musicians,” *Int. J. Bifurcation and Chaos* **17**, 2281–2288.  
 Pastor-Satorras, R. & Vespignani, A. [2002] “Epidemic dynamics in finite size scale-free networks,” *Phys. Rev. E* **65**, 035108.  
 Sarwar, B., Karypis, G., Konstan, J. & Riedl, J. [2001] “Item-based collaborative filtering recommendation algorithms,” *Proc. 10th International World Wide Web Conference (WWW10)*, Hong Kong, May.  
 Song, X., Lin, C.-Y., Tseng, B. L. & Sun, M.-T. [2006] “Modeling evolutionary behaviors for community-based dynamic recommendation,” *Proc. SIAM Intl. Conf. Data Mining*.  
 Wang, J., de Vries, A. P. & Reinders, M. J. T. [2006] “Unifying userbased and itembased collaborative filtering approaches by similarity fusion,” *Proc. SIGIR’06*, Seattle.  
 Watts, D. J. & Strogatz, S. H. [1998] “Collective dynamics of small-world networks,” *Nature* **393**, 440–442.