Proceedings of the Workshop Net-Works 2007,
Aranjuez, 10–11 September 2007, pp. 1–16.

Preferential attachment, aging and weights in recommendation systems

Massimiliano Zanin[1], Pedro Cano[1] and Javier M. Buldú[2][1]

[1] *Music Technology Group, Universitat Pompeu Fabra*

[2] *Nonlinear Dynamics and Chaos Group, Departamento de Física, Universidad Rey Juan Carlos*

Abstract

In the present work, algorithms based on complex networks applications are applied to *Recommendation Systems* in order to improve their quality of predictions. We show how some networks are grown under the influence of trendiness 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 an histogram of user ratings, using therefore a 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: Recommendation systems, preferential attachment, network evolution

1. Introduction

Since the experiment of Milgram in 1967 [1], the study of (social) networks have attracted the interest of many scientists from completely different fields. Boosted by the seminal paper of Watts and Strogatz [2], complex networks theory has become a strong utility to analyze different kinds of data structures. The application of complex networks to social problems has generated special interest, and it has given fruitful results in different subjects, raging from sexual disease control [3, 4] to music community identification [6, 7]. Another field

[1]emails: `massimiliano.zanin@hotmail.com`, `pedro@bmat.com`, `javier.buldu@urjc.es`

where complex networks have been successfully implemented is in 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 cares, and this has supposed a great interest in generating algorithms that recommend items to users that enter an “e-store”.

In the search for better recommendation algorithms using complex networks theory, properties of the system like Clustering Coefficient [13] or Jaccard’s Coefficient [12] have been explored, obtaining different results. When the growth of the recommendation system is considered, the *Preferential Attachment* strategy has been recently proposed [12], but without much consideration within this community.

In this paper, we want to go deeper in 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 [8, 11] 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 the most extended way of generate a recommendation matrix, we take 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 the others in the network, and select the most similar (see [8] for details). For this purpose, a *cosine-based similarity* is commonly used. For each item, a vector of length N is created, being N the total number of users. The vector accounts for the relation between items thanks to user choises: 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 if not). 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:

$$sim(i,j) = cos(\vec{i},\vec{j}) = \frac{\vec{i} \cdot \vec{j}}{|\vec{i}| \cdot |\vec{j}|} \tag{1}$$

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 [8].

In our experimental study we have used two datasets, each one with different characteristics, to observe results in different backgrounds.

The first dataset is the collection of ratings of NetFlix [9], 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 17770 items (movies), 2.6 millions users and more than 23 millions of operations (links).

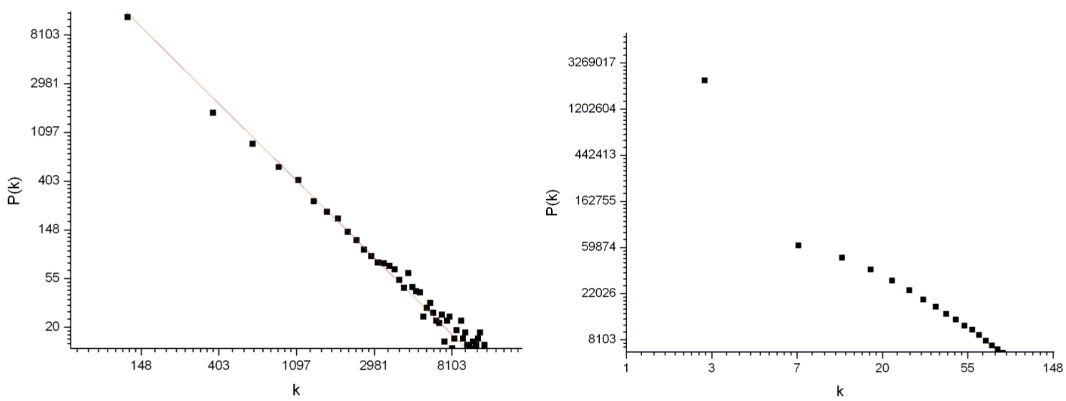

Figure 1: Degree distribution for items (a) (i.e. movies) and users (b) in the NetFlix dataset.

The second dataset, is from *Art Of The Mix* [10]. In this network, we have 90000 users, 472000 items (songs, in this case) and 1.3 millions of links. The *A*rt 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, somehow, fit in those 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 the list, whose origin is based on the musical taste of the playlist' author. Both datasets share the same structure: a line of the network file includes a connection between an user (specifically, the ID of the user) and an item (again, an anonimous ID defining the item), and 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 [12], where 10000 items and 2000 users where considered, or [13] with a dataset close to 40000 items.

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 in [16], the preferential attachment has become a paradigmatic growing algorithm in order to explain the structures and evolution of social networks.

The main idea in [16] 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:

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

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 be cases where the popularity of an item, or the existence of a certain fashion, do 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 consist on a bipartite network [14] with two different kind of nodes, users 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 must 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* $dis(j)$ with respect to the target user is calculated, and values are stored in a linear array:

$$dis(j) = cos(\vec{i}, \vec{j}) = \frac{\vec{i} \cdot \vec{j}}{|\vec{i}| \cdot |\vec{j}|} \tag{3}$$

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

- For each item l of the network, we define a *compatibility* value $comp(l, user)$ 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 $1 - dis$:

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

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

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 can be 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 this connections, its compatibility value (i.e. the compatibility between the selected and the target user) 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); this is the basis of preferential attachment: the more connections, the more the probability of being chosen by another user. On the other side, not only the number of links is considered: the compatibility is calculated, like in the basic algorithm, to be a representation of the user tastes; if an item is well-known, but is far from the tastes of the target user, its total compatibility value 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 the trendiness component is the preference for an user for items with high popularity: in the case of customers datasets, buying items in an e-store, as the NetFlix dataset, the more an item is known, the more is likely for that item to be chosen by the target user. Up to now, all data previous to prediction date has been considered. Except some works like [17] or [18], this has been the traditional approach, since it is a generalized opinion that the more data is used in calculation, the better the result will be. Nevertheless, trendiness of an item greatly depends on time: one item can have a high popularity on a time t_0, but it can lose all interest after a certain time t_1.

This fact can be observed in Fig. 2. The left plot shows an hypothetical evolution of the number of new links for an item A (i.e., the derivative of its degree): in time t_0 this item has a great instantaneous degree (i.e. a great popularity in a given moment, with many new users connecting to this item), while close to t_1 its number of new links decreases. On the other side, item B has an overall lower degree, with a greater degree close to time t_1. It is important to note, that item A has a greater 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,

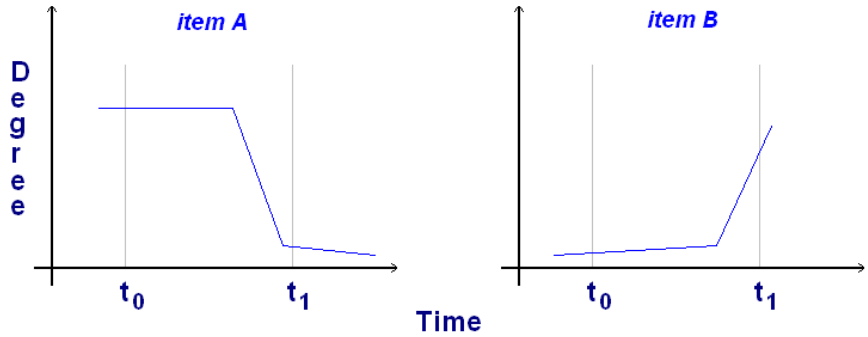

Figure 2: Example of degree evolution for two items; item on the left has a higher global degree, while item B has a higher degree in time t_1.

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 advantage 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.

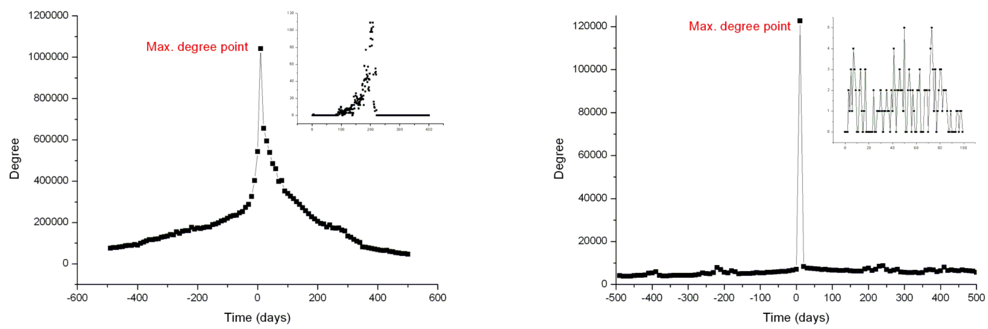

Figure 3: Global degree evolution for NetFlix (a) and Art Of The Mix (b) networks: the central point represents the moment of greatest degree of every item. In the insets, are represented the degree evolution of an example item for each network; note as for (b), the degree shows no clear peak: the mean degree evolution for that network is therefore flat.

In Fig. 3 we represent how the instantaneous degree of the items evolves in time. The instantaneous degree 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 evolution for a given item. In order to account for all items, we add the instantaneous degree of all items, but aligned at their absolute maxima. Fig. 3 shows how we obtain different results for both networks. For the NetFlix dataset, a great peak is observed, with the degree value increasing and decreasing continuously 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. Second, popularity depends on time, and therefore, we must take it into account at the moment of recommending an item.

The opposite case is Art Of The Mix, where the instantaneous degree level of the whole dataset is quite constant, with only a central delta-shaped peak. In fact, the central peak is an artifice: 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 adding all items. The absence of correlation in the degree evolution indicates that relations between users and items do not depend on time, as we don't have a transaction like structure, and that trendiness is not important to explain 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 Fig. 4 and Fig. 5 (NetFlix), and Fig. 6 (ArtOfTheMix). 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. When applying an aging filtering to Art Of The Mix network we do not obtain an improvement of the score (see Fig. 6): as degree evolution is not important in this kind of network, reducing the dimension of the 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. 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; moreover, we have previously seen how an user-based strategy is more efficient that an item-based one: once again, the speed of the new algorithm is more suitable for real-time implementations.

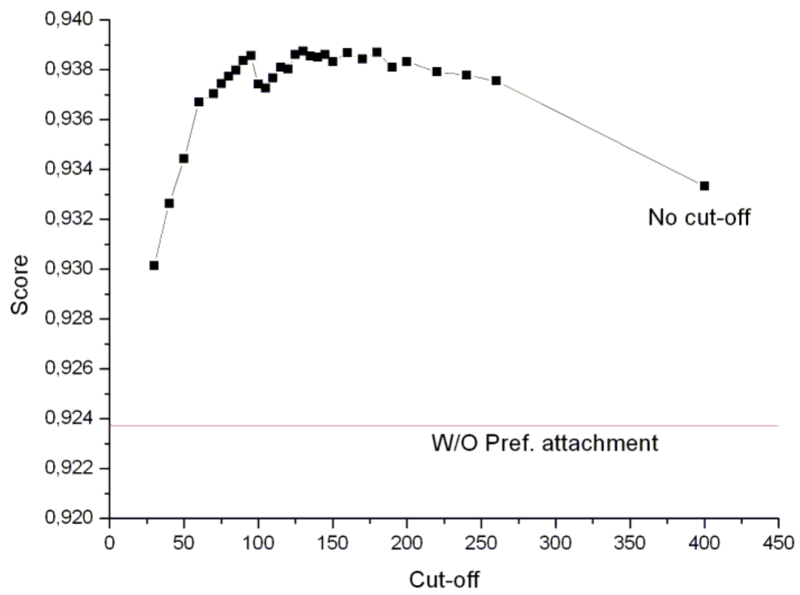

Figure 4: Recommendation score as a function of cut-off window dimension d, for NetFlix dataset. The horizontal line represent the score for the basic item-based algorithm, while the right point, marked with *no cut-off*, is the result of using the preferential attachment algorithm without filtering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

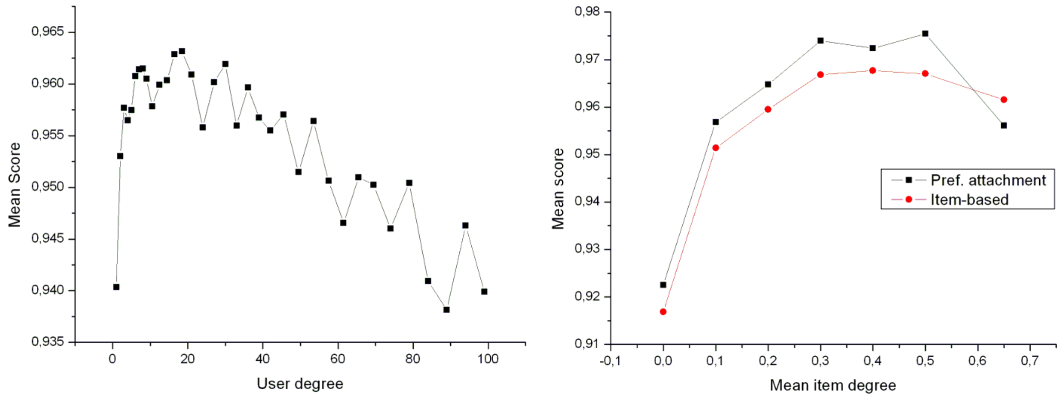

Figure 5: Recommendation score for NetFlix dataset. (a) Mean score obtained for users with different number of items; the system needs a minimum of 3 movies connected to an user to recommend to him with good results. (b) Mean score obtained as a function of the mean degree of the movies chosen by an user, for the standard and the preferential attachment algorithm.

4.. 3 Score calculation

In the previous section, he 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 would return 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:

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

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 (i.e. MEA, RMSE or

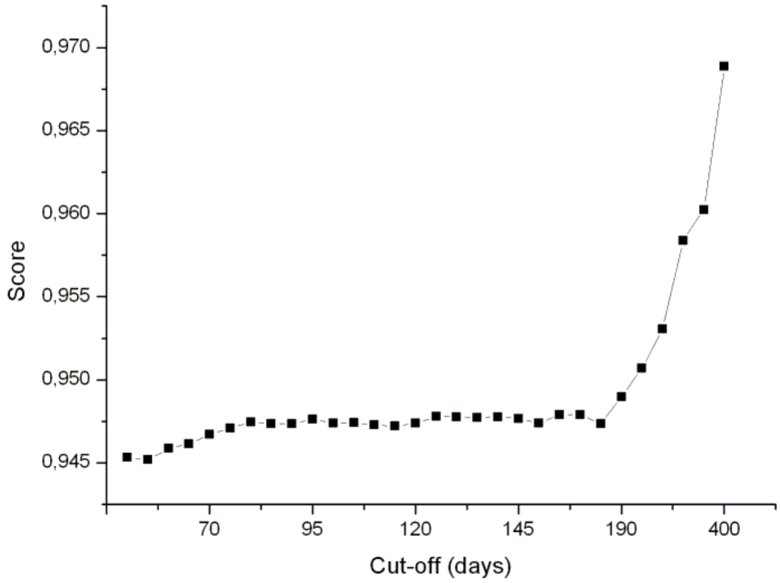

Figure 6: Recommendation score as a function of cut-off window dimension d, for Art Of The Mix dataset.

Precision/Recall/F-measure) are often taken as standard measures. To make an example, in [15] 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 $t_1 << t_0$, 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 α and β:

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

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

$$comp(l) = \sum_j (1 - dis(j))\, W(j \to l) \tag{5}$$

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

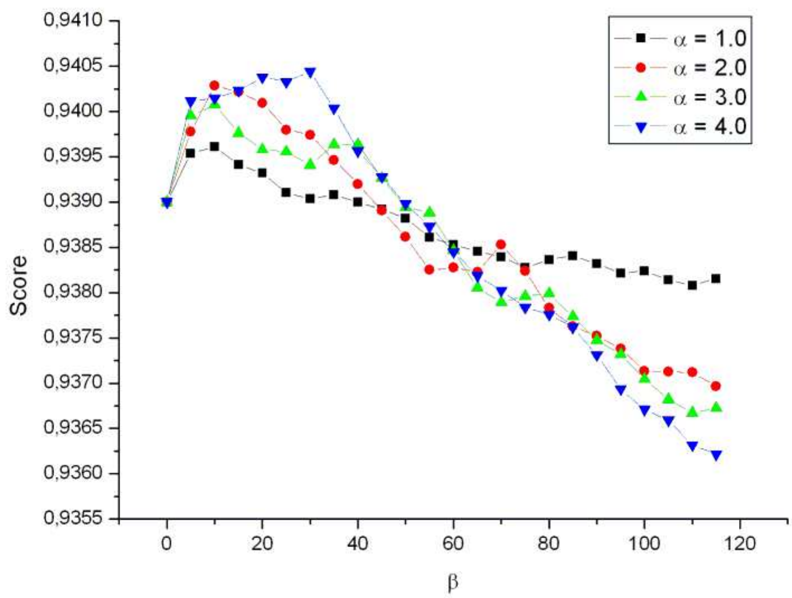

Figure 7: Effect of considering weighted links. Results refer to the NetFlix dataset for a window dimension of 120 days.

The obtained *score* for different values of α and β on the NetFlix data collection is shown in Fig. 7. A maximum is detected around $\beta = 20$ for different α, while large values of β lead to a reduction of the *score*. This behavior is expected since high values of β are equivalent to increase the importance of old links, a fact that is not favorable for a preferential attachment strategy. On the other side, low values of β are equivalent to include only very young links, excluding a great quantity of information, and making the *score* value to decrease.

6. Discussion

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 target item is item number 872 (for privacy issues, users and items are encoded with sequential numbers). Target user has links with 24 other items in the moment of the recommendation.

First, we calculate the *score* using the basic algorithm. After making the ranking ordered by compatibility, in firsts positions the followings items are founded, along with their compatibility score:

Item	(1^{st}) 7843	(2^{nd}) 5085	(3^{rd}) 11038	(4^{th}) 14241
Compatibility	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 17770 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 take 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 report what can be obtained with $d = 70$ days. In this case, after filtering the dataset, we have only 2.26 millions operations (about ten time less than original data), and target user has 3 more links to other items. Target item 872 is connected with 198 users in that interval of time, and their compatibility with target user are the following:

User	(1^{st}) 698478	(2^{nd}) 2081171	(3^{rd}) 1558760	...
Compatibility	0.04352	0.06337	0.05803	...

Summing up all 198 values give a total compatibility of 14.69987. In this example, we can see as the compatibility value 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:

Item	(1^{st}) 13728	(2^{nd}) 14240	(3^{rd}) 2782	(4^{th}) 11521
Compatibility	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	∞
Score	0.87530	0.97794	0.97766	0.97535	0.9740	0.97840

7. Conclusions

In recommendation systems, it is a common opinion that the bigger the dataset, the better the result will be. In this paper, we show that in certain case 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

[1] S. Milgram, "The small world problem", *Psychol. Today,* **1** (1967) 61–67.

[2] D. J. Watts and S. H. Strogatz, "Collective dynamics of small-world networks", *Nature* , **393** (1998) 440–442.

[3] R.M. May and A.L.Lloyd, "Infection Dynamics on Scale-free Networks", *Phys. Rev. E,* **64** (2001) 066112.

[4] R. Pastor-Satorras and A. Vespignani, "Epidemic dynamics in finite size scale-free networks", *Phys. Rev. E* **65** (2002) 035108.

[5] Peter S. Bearman, James Moody and Katherine Stovel, "Chain of Affection: The Structure of Adolescent Romantic and Sexual Networks", *AJS*, **110-1** (2004) 44–91.

[6] R. Lambiotte and M. Ausloos, "On the genre-fication of music: a percolation approach", *Eur. Phys. J. B* **50** (2006) 183–188.

[7] Juyoung Park, Oscar Celma, Markus Koppenberger, Pedro Cano and Javier M. Buldú, "The Social Network of Contemporary Popular Musicians", arXiv:physics/0609229v1 (2006).

[8] Badrul Sarwar, George Karypis, Joseph Konstan and John Riedl, "Item-based Collaborative Filtering Recommendation Algorithms" *In Proc. of the 10th International World Wide Web Conference (WWW10)* Hong Kong, May (2001).

[9] http://www.netflix.com

[10] http://www.artofthemix.org

[11] J. Wang, A.P. de Vries and M.J.T. Reinders "Unifying Userbased and Itembased Collaborative Filtering Approaches by Similarity Fusion" *Proceedings of SIGIR'06* Seattle (2006).

[12] Zan Huang, Xin Li and Hsinchun Chen, "Link Prediction Approach to Collaborative Filtering", *In Proc. of the 5th ACM/IEEE-CS joint conference on Digital libraries*, 141–142 (2005).

[13] Zan Huang, "Link Prediction Based on Graph Topology: The Predictive Value of the Generalized Clustering Coefficient", *LinkKDD* (2006).

[14] M.E.J. Newman, "The structure and function of complex networks", *SIAM Review*, 45 (2003) 67–256.

[15] Zan Huang, Daniel D. Zeng and Hsinchun Chen, "Analyzing Consumer-product Graphs: Empirical Findings and Applications in Recommender Systems", *SSRN September* (2005).

[16] A.-L. Barabási, R. Albert, "Emergence of Scaling in Random Networks", *Science* **286** (1999).

[17] Xiaodan Song, Ching-Yung Lin, Belle L. Tseng and Ming-Ting Sun, “Modeling Evolutionary Behaviors for Community-based Dynamic Recommendation”, *Proc. of the SIAM Intl. Conf. on Data Mining* (2006).

[18] J.L. Herlocker, J.A. Konstan, L.G. Terveen and J.T. Riedl, “Evaluating Collaborative Filtering Recommender Systems”, *ACM Transactions on Information Systems (TOIS)* (2004).