An investigation into the effect of recommendation thresholds on item diffusion in homophily-driven systems

Dominic Hunt

1st July 2011

Abstract

This study attempts to shed light on the influence of item recommendation thresholds in systems where the diffusion of items is driven by homophily. A simulation is constructed to look at the diffusion of objects, identified by their attributes, across a directed network of nodes, with comparable attributes. The diffusion of objects is governed by the expected similarity of nodes and the similarity of an object to the recommending node. Variations in similarity thresholds are studied. Preliminary findings show that the cluster size per object follows a power law, the coefficient of which varies with respect to the recommendation threshold.

The diffusion of an item or idea through a group has become a popular subject of analysis in the past 20 years with the availability of large datasets of user interactions. These can range from individual shopping data for a supermarket to the set of political blogs[1][2]. Besides any pure interest in understanding how diffusion occurs in society, understanding the properties of these interactions will be useful in bringing awareness of products to potential customers as well as in the feasibility of information control.

When modelling a property of a system, it can be considered as being based on some global parameters or as a series of interactions between nodes with different properties. The former approach is used when considering network diffusion as an epidemic. Both biological and computer virus propagation can be modelled by considering each node as either infected or susceptible. The node has a probability of changing from one state to the other, producing cascades of infection. This approach has had some success in modelling the networks produced by viral marketing[3][4] as well as to the spreading and persistence of computer viruses[5].

Models with varying node parameters are generally used when analysing social networks or recommendation communities such as *aNobii*. The site provides the facilities for people to list the books they wish to read, are reading and have read, and then rate them. A user can also see the profiles of other users and see what they think of the books they have read. One recent study[6] found users not only tended to create links with those whose were close geographically, in interests or in the network, but that their similarity increased over time.

One of the difficulties in studying recommendations is the identification of which ones were listened to. When a user recommends an item and another user, to whom he/she is already socially linked, begins recommending this item has there been a *direct recommendation*? While the two recommendations are correlated their emergence may be due to *homophily*, the tendency for people who are similar to perform the same actions. Another difficulty is that the second recommender may well have multiple associates who had previously recommended this item and could potentially have been the cause of the recommendation. A series[7][8] of recent studies attempt to distinguish between homophily, and contagion. In so doing they highlight the significant impact homophily has on the perceived propagation of items through a social network.

Recent work on aNobii by Camille Roth and Chih-Chun Chen, as yet unpublished, looks into the distributions of book ratings given by users. They showed that users socially linked to previous raters maintained the distribution of ratings for the book. Their results bring up the question of how people select books based on the ratings of friends.

This study looks at one aspect of this question; how the user chosen *recommendation thresholds* affect the book diffusion structure. This is achieved by way of simulations, allowing for a much broader range of information to be gathered on the system with no external influences.

In this study we first construct a network of *users* and *friends*. Users then 'read' an initial set of *books*, some of which they will recommend. For every following timestep a user is chosen to select and read a book based on their friends' recommendations. Recommendations will be accepted from friends expected to be similar to the user. This assumes that the book preferences of a friend can be assessed by looking at the attributes of previously recommended books. As the simulation progresses this assessment will change, with some friends no longer being considered suitable and others becoming suitable. The thresholds for book recommendation and friend similarity will be equivalent for this study. It is the properties of the network formed by the book *accepted recommendations* that we will be analysing.

Simulation details

A book has an *attribute-list* of size N that fully describe its similarity to other books. Users have an equivalent attribute-list that fully describes their similarity to other users and their affinity to books. These attributes are not associated with any real properties but are assumed to be a universal orthogonal basis by which all user preferences and book properties can be completely described. Initially, a user is aware only of their own properties. The number of attributes is fixed throughout the simulation and equal for books and users, allowing easy comparison.

In practice it is expected that the number of attributes necessary to describe an item will be very large. However, in these simulations computational constraints prevented the use of many attributes.

In the real world most attributes would be continuous. However, for ease of computation the values are being limited to 0 and 1. Any particular attribute-list can therefore be represented on a unit hypercube of dimension N.

From this construction a *distance* can be defined between attribute-lists X

and Y as

$$D_{XY} = \sum_{i=1}^{N} \| X\left[i\right] - Y\left[i\right] \|$$

We can define the complementary measure, *similarity*, as

$$S_{XY} = \sum_{i=1}^{N} \delta\left(X\left[i\right], Y\left[i\right]\right)$$

A book is a passive object, completely described by its attribute-list. The set of books available in the simulation, B, is generated at the start and not varied over the simulation. It is a set of unique books, each generated by randomly selecting from $\{0, 1\}$ for each attribute. For N dimensions this gives 2^N possible combinations of books. To try and get the most data from a simulation, all books will be used.

On 'reading' a book the user becomes aware of the attribute-list of that book. Until then the user has no information on a book, other than its existence.

Users

A user can read books, recommend books and follow the recommendations of selected users, called friends. However, a user can only see the recommendations and name of its friends. This is an approximation of what was seen on aNobii, where users can see the ratings on a scale from one to four that users provide for the books they have read.

Unlike the set of books, the set of users can contain attribute-list duplicates. The attribute-lists are generated by randomly selecting from $\{0, 1\}$ for each attribute. Like the books, once the system has been initialised no new users are added.

The number of users, U, for these simulations is taken to be $\lceil \frac{B}{N} \rceil$. This value has no justification other than being chosen to allow the system to have significantly more books than users, while still providing enough users for reasonable diffusion of books.

Recommendations A book that has been read by a user can be recommended if it appeals to the 'tastes' of the user. A recommended book is one that a user has read and whose similarity to the user is equal or above a given threshold value. Recommendations are then accessible to any user who has made this user their friend. Recommendations will be accepted from friends who are expected to have a similarity equal or above a threshold value. For the simulations considered here this value will be equal to the book recommendation threshold.

As this is the only way for a book to be recommended, the similarity level at which users are no longer prepared to recommend a book will have a significant impact on the diffusion of a book. This is due to two factors; a higher proportion of users being prepared to recommend a book and from a greater number of friends from whom recommendations can be considered. The combination of these effects will potentially change the distribution of accepted recommendations per user per book.

Currently the threshold is constant and equal for all users with a value of $\lceil \frac{2}{3} N \rceil$. This was chosen as an arbitrary starting point that could be varied later.

Friends A friend of a user is someone who can be seen by that user. The friends of each user are selected at the beginning and stay fixed. The creation of new friends on aNobii was found to be much slower than the reading of books. Allowing users to add a few friends during the simulation would have significantly increased the complexity of the simulation without any benefits to the analysis.

The information that a user has about a friend is limited to that friend's name and recommendations. Users has no information about friends of their friends. A friendship link is directional and therefore not necessarily mutual.

The network is constructed using a preferential attachment rule, with each user adding $\frac{U}{2}$ friend connections, starting with a non-zero probability of connecting to any user.

During the construction of the network the probability of user X connecting to user Y is calculated as:

$$P\left(\text{X connecting to Y}\right) = \frac{\text{Number of connections to } Y + 1}{\text{Total number of connections} + U}$$

Since the friend connections can be to any user it is possible when creating a new link for the user to try to connect to itself or to one of its previously selected friends. In these cases the connection is ignored, leading the users to have anywhere between 0 and $\frac{U}{2}$ friends. The maximum number of friends has been arbitrarily chosen to provide short paths between any two users while still keeping most of them separate. As a user only takes recommendations from friends who are similar, most friend connections will not be used to provide recommendations.

Assessing the similarity of friends A user assesses its similarity to each of its friends based on what the it thinks the likely attributes of its friends are. Feasible attribute sets are those which could give rise to the list of recommended books provided by the friend. However, the user does not necessarily have access to the attribute-lists of all their friend's recommended books. The user can therefore only calculate a superset of the feasible attribute sets based on a subset of the friend's recommended books.

Based on this feasible superset the user can calculate the probability of user Y being a given distance away from user X. This is done by looking at what proportion of the feasible superset is that distance away from the user. The expected distance for friend Y from user X can be calculated using a weighted sum of these ratios.

This method of calculating similarity assumes that the user's recommendations are based on what they would currently recommend if they were to read all their books now for the first time. This allows the simplifying assumption of having the same weight given to each recommendation, regardless of when it appeared.

Once a book has been read the user checks to see which of its friends have read this book and updates their similarity ratings. Since these predictions are based on the sets of feasible friend preferences this update will also incorporate the changes due to any new books the friends have read and the user had already read. This means when the user selects a book from their friends' recommendation lists the selection will be based on the predicted similarity calculated the last time the user read a book. The books the friend has read in the meantime will not be taken into consideration for the selection which user to select a book from. This is expected to slightly increase the dislike rate on selecting a book to read as friends who have a lower similarity than the threshold are more likely to be picked.

This system of only receiving recommendations from some of the friends in effect produces a secondary dynamic and far less connected network of users through which the actual recommendations pass. It is the cascades of recommendations through this network that we are actually interested in.

Initial books read Books are randomly selected from the whole distribution with each book having the same probability of being selected. The initial number of books read is selected randomly from a uniform distribution. The purpose of the distribution is to limit the overall impact of the random initial selection of books by keeping the set small while trying to keep it large enough that almost all users recommend at least one book. The latter constraint is achieved by setting the minimum number of books read m such that only 1% of users are expected not to be able to recommend a book with m books read. The maximum is set as the lower of $2m$ and B.

Reading new books A user takes reading recommendations from those friends who have an expected similarity to the user greater or equal to the recommendation threshold. Those friends who can recommend are each given a weight proportional to their expected similarity to the user. Based on these weightings one of the users is randomly selected and a book is randomly selected from their recommendation list. If no new books are in the recommendation list of that friend then the friend is removed from the weighted list and another friend is selected. This continues until there are no more friends who could give recommendations to the user. In this case the user reads a random book that they have not read before.

This method allows for an indirect weighting of books with regards to how many of a user's friends recommend a book. If a book is recommended by multiple friends the probability of selecting a friend who has read the book and then selecting the book itself will be higher.

The number of times a user will be forced to select a book at random should be inconsequential for a low enough recommendation threshold. However, for very high recommendation thresholds it will be very frequent that none of a users friends have any books to recommend. In part this will be due to each user not recommending many of the books they come across and in part due to the way in which users estimate the preferences of another user.

If we take the example where a friend is actually a distance of 1 from the user and the recommendation threshold is $N-1$. Then the friend would be one where the user should take recommendations from. The books that this friend could be recommending would either be a distance zero, one or two from the user. Under the current method of calculating the expected attribute set of a friend, both the friend and the user would have had to have read the book with the same attribute set as the user for the user to consider recommendations from that friend. Since this is very unlikely, diffusion is unlikely to occur.

The set of books that could potentially be recommended by a user's friends is of size $c\left(1, \min\left(2r, N\right)\right)$, where r is the difference between the number of attributes and the recommendation threshold and

$$c\left(a,b\right)=\sum_{k=a}^{b}\binom{N}{k}$$

Given all the books that could be recommended by friends only

$$\frac{c\left(1,r\right)}{c\left(1,\min\left(2r,N\right)\right)}$$

would be recommended by the user. Since we do not know what proportion of a given user's friends will fall within the recommendation threshold the actual fraction of books recommended by a friend that the user would also recommend, h, will be

$$1\leq h\leq\frac{\left(r+1\right)^{2}}{c\left(1,r\right)}$$

This assumes that the friend has read all the books that they would recommend. In other cases we have no information on the fraction of books.

Another way of approaching this is from the perspective of a book. The attractiveness of a given book will vary depending on the user, with a cut-off distance from its attribute point where it will no longer be recommendable. A book therefore has a fixed number of users who could potentially recommend it, which in this case is anywhere between the number of books a user could recommend and zero. How those users are connected by friend links will determine whether a book can propagate far or not.

Timesteps

A maximum useful number of simulation timesteps emerges based on the number of timesteps necessary for each person to read a bit more than every book they could recommend. The extra bit comes from the books read that have not passed the user's recommendation threshold, either due to them being 'poor' recommendations or from the random set of initial books read. Once this maximum useful number is reached there is no point in continuing the simulation as all users will have as much information about their friends as would be useful; no extra information would help with the assessment of who to take recommendations from.

Depending on the number of initial users many books that a user would recommend will only become accessible to them when randomly selecting a book. This leads to the useful number of timesteps being lower, as some books will be read only by people who would not recommend them. The end result will therefore not be a complete set recommended books but a distribution of books read.

The number of timesteps is therefore calculated as the number of users multiplied by the number of books they could potentially recommend.

While it is true that the network progresses as the timesteps increase it is not clear that there is any direct mapping between real-world time and timesteps. The network of recommendations is a useful snapshot of the propagation of a recommendation network, but it is unlikely that the changing state of the network during the simulation has real world parallels.

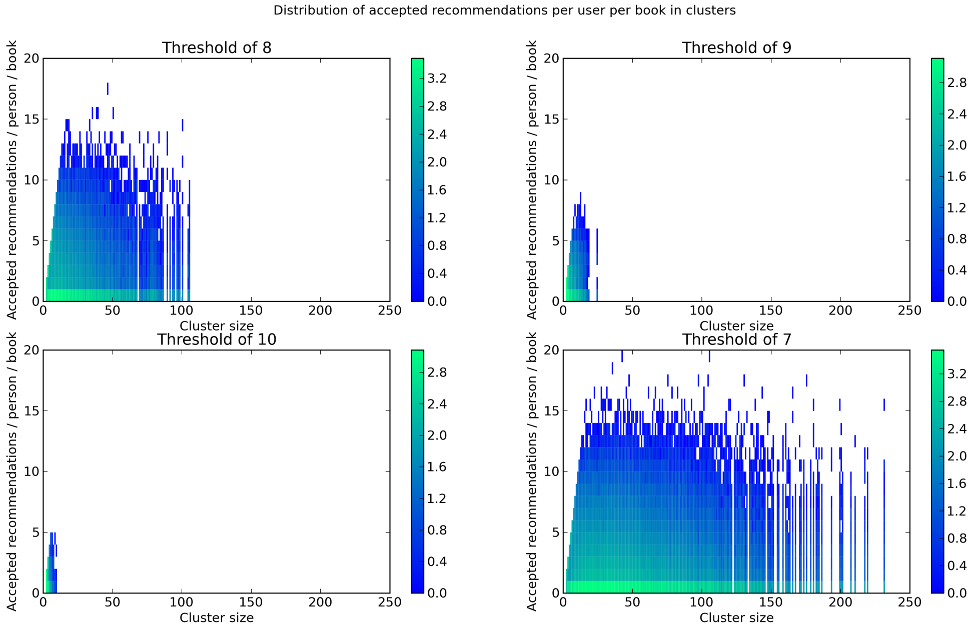

Figure 1: Heatmaps showing how the distribution of recommendation acceptances change with cluster size. The colour scale is $\log_{10}$ of the number of nodes fitting the two axis properties. Each cluster comes from a particular book.

Results from simulations

For computational reasons the simulations were performed with $N = 11$ and use the functions specified above to calculate all fixed variables.

When looking to see if one parameter, in this case the recommendation threshold, affects the eventual properties of the system it is necessary to rule out all other possible effects. For this reason simulations have been run looking at the evolution of the system as the number of timesteps and people. This was initially done by looking at the changes in book popularity distributions. Since these analyses are not directly related to this study their graphs have been placed in the appendix. When the length of the simulation is varied there are changes in the distribution of book popularity. When increasing the number of users there is no variation in the distribution when normalised by the number of users.

In the following graphs the results are aggregated from five different simulations with identical input parameters. The variation between simulations has been found to be small so the aggregation acts as a smoothing of the datasets.

To try and identify any changes in the structure of how books reading diffuses, a directed network of recommendation links was created for each book. This was used to identify *clusters* of book reading. A cluster is defined as a set of users who have either had their book recommendation accepted by another user in the set, have accepted the book recommendation of one of their friends in the set or both. In each cluster there will only be one user with only out links.

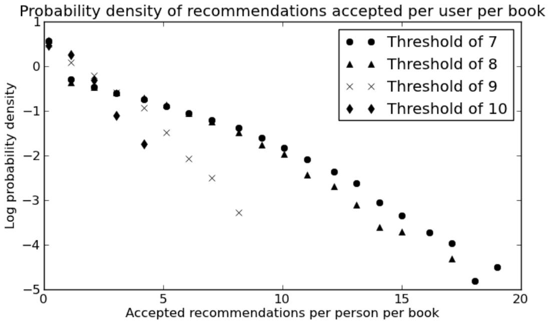

Figure 2: Histogram of the $\log_{10}$ probability density of recommendation acceptance per user per book

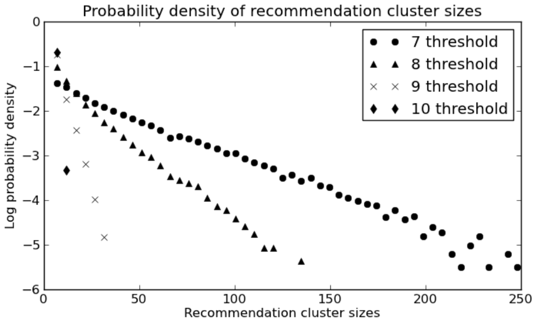

Figure 3: Histogram of the $\log_{10}$ probability density of cluster size. Each cluster comes from a different book network.

A book may well have quite a few distinct clusters of recommendations. Since the network properties are book agnostic we can collate these together and look at the distribution of out links across nodes for each cluster size. Figure 1 shows how these distributions vary across recommendation thresholds.

For a given recommendation threshold we can see there is a maximum number of recommendation acceptances, or out links, that a given node can expect to have for a given book. This appears to suggest that within a cluster there is a saturation point for local recommendations. One possible cause for this would be the number of times a given user is not only a friend, but also considered to be similar enough to provide recommendations.

The accepted recommendations across all cluster sizes in figure 2 we can see the recommendation thresholds change probability density gradient. From the cluster size distribution in figure 3 it is clear that this is a power law with a coefficient that appears to be dependent on the recommendation threshold. The cluster size per book is therefore scale free.

Each of these book networks are not independent as the relationships between users are defined across books. It therefore makes sense to look at the network as a whole. For the thresholds nine and ten the distribution of clusters shows a few very small clusters and one larger cluster. For the thresholds of seven and eight there is only one cluster containing all nodes. For this reason

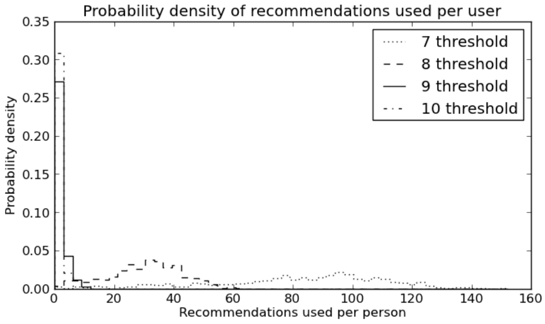

Figure 4: Histogram of the probability density of the recommendation acceptance per user for all books

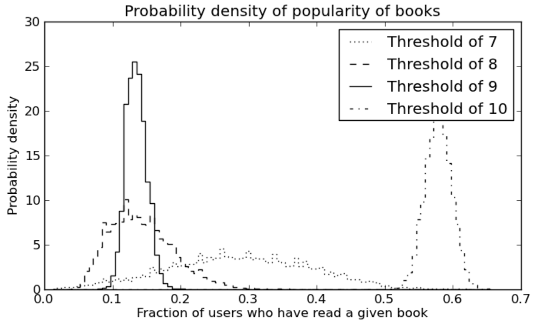

Figure 5: Histograms showing the popularity frequency of different books.

we will only look at the distribution of recommendation acceptances, as shown in figure 4.

The distribution of out links, with a slightly off centre mean and a negative skew shows once again the independence of book choice. The Gaussian-like distribution has an increasing mean as the recommendation threshold increases. Overall, the distributions would appear to suggest that the nodes with high recommendation acceptances for one book are not likely to be in that position for other books. Another way to put this is that users reading one book after most users is likely to be early in reading some other book.

The distributions for the thresholds studied are largely the same. However, as the number of timesteps a simulation runs for currently depends on the threshold, for the systems to be comparable they would need to reach the same state. This is brought into question by figure5, showing how the different book-popularity distributions vary. The variation would be an indication of changes due to the recommendation threshold were they not similar to those seen when varying the number of timesteps, as shown in figure 7.

Summary and where to go next

In this study we have looked at how modifying the recommendation threshold affects homophily-driven diffusion. A simulation was constructed that looked at the diffusion of objects with attribute-lists across a directed network of nodes with comparable attribute-lists. This has been tied to a book recommendation network between user friend links.

The networks show a scale free cluster size for individual books with the coefficient dependent on the recommendation threshold. The assumed relationship between the recommendation threshold and the number of timesteps is called into question by the different shapes of the distributions of popularity of books and their resemblance to the variations in distribution across different numbers of timesteps.

There is still much to be looked at in this simulation before any definitive answer can be given on the impact of the recommendation threshold.

An important priority would be finding methods of extending the simulation to work feasibly in higher dimensions. This can be achieved by finding a general solution for N-large dimensions or by finding an approximation to the brute force method that has been used so far, allowing a wider range of recommendation thresholds.

Breaking the link between the book recommendation threshold and the acceptance of recommendation threshold would allow more flexible control over how the roles of these two properties are related and what equivalences can be made in their values.

To generalise the results it would be useful to remove the constraint that attribute sets contain only zero or one, and instead allow for continuous variation between the two.

References

[1] Jean-Philippe Cointet and Camille Roth. Socio-semantic Dynamics in a Blog Network. *2009 International Conference on Computational Science and Engineering*, (6):114–121, 2009.

[2] Telmo Menezes and Camille Roth. Precursors and Laggards : An Analysis of Semantic Temporal Relationships on a Blog Network.

[3] Jure Leskovec, Lada A Adamic, and Bernardo A Huberman. The Dynamics of Viral Marketing . *Machine Learning*, 1(May):1–46, 2007.

[4] Esteban Moro. Information diffusion epidemics in social networks. pages 1–12, 2008.

[5] Romualdo Pastor-satorras and Alessandro Vespignani. Epidemic dynamics in finite size scale-free networks. *Networks*, pages 1–4, 2008.

[6] Luca Maria Aiello, Alain Barrat, Ciro Cattuto, Giancarlo Ruffo, and Rossano Schifanella. Link creation and profile alignment in the aNobii social network. *Proceedings of the Second IEEE International Conference on Social Computing SocialCom 2010*, pages 20–22, June 2010.

[7] Andrew C Thomas. Homophily and Contagion Are Generically Confounded in Observational Social Network Studies. pages 1–27, 2010.

[8] Sinan Aral, Lev Muchnik, and Arun Sundararajan. Distinguishing influence-based contagion from homophily-driven diffusion in dynamic networks. *Methods*, 2009.

Elimination of other possible influences

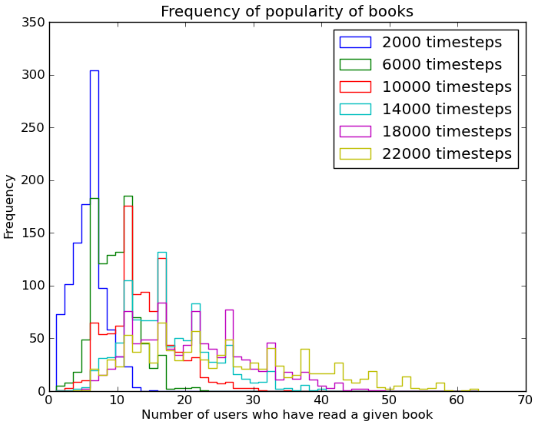

Figure 6: Histograms showing the popularity probability density of different books across a range of simulation lengths.

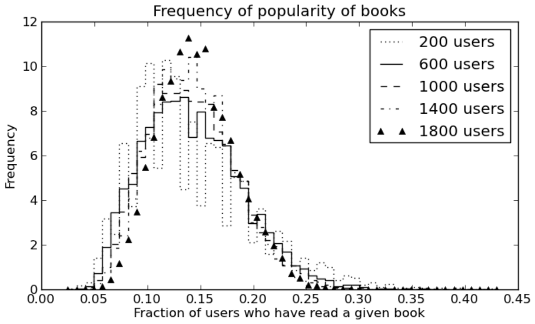

Figure 7: Histograms showing the popularity probability density of different books across a range of population sizes.