Overcoming Data Sparsity & Bias in Order to Recommend from the "Long Tail"

Problem presented by

Benjamin Dias (benjamin.dias@unilever.com)

Unilever UK

Executive Summary

Unilever is currently designing and testing recommendation algorithms that would make recommendations about products to online customers given the customer ID and the current content of their basket. Unilever collected a large amount of purchasing data that demonstrates that most of the items (around 80%) are purchased infrequently and account for 20% of the data while frequently purchased items account for 80% of the data. Therefore, the data is sparse, skewed and demonstrates a long tail. Attempts to incorporate the data from the long tail, so far have proved difficult and current Unilever recommendation systems do not incorporate the information about infrequently purchased items. At the same time, these items are more indicative of customers' preferences and Unilever would like to make recommendations from/about these items, *i.e.* give a rank ordering of available products in real time. Study Group suggested to use the approach of bipartite networks to construct a similarity matrix that would allow the recommendation scores for different products to be computed. Given a current basket and a customer ID, this approach gives recommendation scores for each available item and recommends the item with the highest score that is not already in the basket. The similarity matrix can be computed offline, while recommendation score calculations can be performed live. This report contains the summary of Study Group findings together with the insights into properties of the similarity matrix and other related issues, such as recommendation for the data collection.

Version 2.0
5 June 2008

Report coordinator

Vera Hazelwood (Industrial Mathematics KTN, vh@industrialmaths.net)

Contributors

Rosemary Apple (University of Edinburgh, r.apple@sms.ed.ac.uk)
Chris Cawthorn (University of Cambridge, c.j.cawthorn@damtp.cam.ac.uk)
Kwan Yee Chan (University of Manchester,k.chan@maths.manchester.ac.uk)
Oded Lachish (Univerisity of Warwick, oded@dcs.warwick.ac.uk)
Achim Nonnenmacker (University of Edinburgh, A.Nonnenmacher@ed.ac.uk)
Mason A. Porter (University of Oxford, porterm@maths.ox.ac.uk)
Sylvain Reboux (University of Nottingham, pmxsr@nottingham.ac.uk)

ESGI64 was jointly organised by
Heriot-Watt University
The Knowledge Transfer Network for Industrial Mathematics
The International Centre for Mathematical Sciences

and was supported by
Engineering and Physical Sciences Research Council
The London Mathematical Society
The Institute of Mathematics and its Applications
The European Journal of Applied Mathematics

Contents

1 Introduction

The Mathematical And Psychological Sciences (MAPS) group, at Unilever Corporate Research have been investigating various personalisation algorithms in order to understand how their performance varies according to different data sets and application scenarios.

Over the past few years, researchers at MAPS have collaborated with several retailers, including the Swiss online supermarket LeShop (www.LeShop.ch), in analysing individual shopping basket (cf. loyalty card) data. As part of these collaborations, MAPS group have developed and deployed online personalised retail recommender systems, which serve as a test-bed in which they can evaluate the performance of Unilever's personalisation algorithms [1].

2 The nature of the problem

A key challenge Unilever face in this area is that the data is sparse and skewed. This affects the performance of most (if not all) personalisation algorithms. Typically, retail shopping data has a distribution similar to that illustrated in Figures 1 and 2. The phenomenon seen in Figures 1 and 2 is known as the "Long Tail" effect, where few items (20%) are bought very frequently and most items (80%) are bought very few times (the term "The Long Tail" was popularized by author Chris Anderson in his book [2]). Furthermore, the pair-wise co-occurrence matrix generated from this data is very sparse, as not many items ever occur together in the same basket. Therefore, Unilever are currently unable to make any meaningful recommendations from the long tail (*i.e.* from the many items that aren't bought very frequently). Whenever they include the items in the long tail in the modelling, the signal to noise ratio decreases significantly. This leads to a significant decrease in model performance. Therefore, as per the current common practice, Unilever currently ignore the long tail and only model the remaining more frequently purchased items.

Although Uniliver collaborators' data are confidential, the properties of this data are very similar to publicly available retail shopping datasets (*e.g.* [3] and [4]). Therefore, in order to overcome confidentiality issues, Unilever propose the use of one or more of the publicly available datasets during this study group.

Although the transactions relating to each individual item in the long tail is small in absolute numbers, collectively they cover a substantial fraction of all transactions (and hence sales). A shopper's rarer purchases are also more informative of their tastes and preferences than their purchases of very popular items such as bananas and toilet roll. Furthermore, in the context of a recommender system, it makes more sense to recommend more personally relevant, personally appealing, serendipitous, *etc.* items, instead of the most popular items such as bread and milk, which the shopper would buy anyway. Hence, being able to use the items in the long tail both as inputs to and outputs of the models is very important.

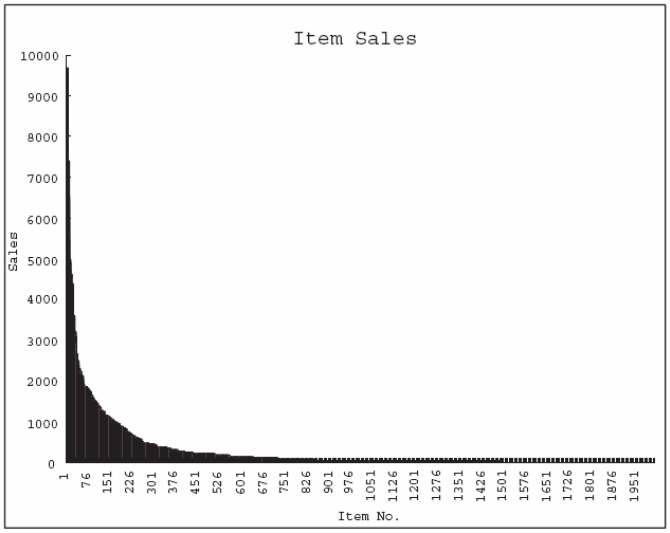

Figure 1: Data Skewness: Items ordered by their frequency of purchase.

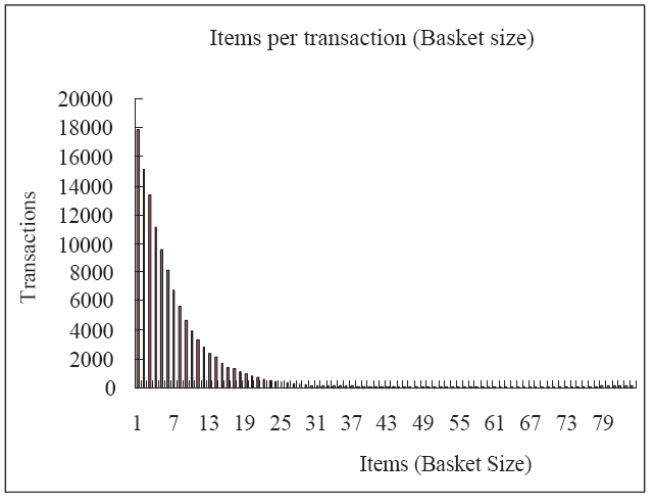

Figure 2: Data Sparsity: Items per transaction.

3 Unilever's request to Study Group participants

The main goal is to develop a probabilistic model that is able to assign a probability of purchase $P(i|s, B)$ to each item i, for shopper s given the current contents of their basket B. The recommender system will then be assumed to recommend the n items most likely to be purchased next. As Unilever's live system currently provides each shopper with three recommendations, they currently set $n = 3$, in the analysis.

Ideally the participants would shortlist their proposed techniques for dealing with the Long Tail effect and, implement and test the most promising ones on the

data provided. Unilever hope to implement and deploy at least the three most promising techniques and approaches resulting from this study group on their live system, following successful further specific testing on collaborator's data. Any insights into consumer shopping behaviour, which naturally fall out from the work or analysis carried out by the participants of the study group, would be considered as a very valuable bonus outcome.

4 Study Group Work

4.1 The Data

Two datasets were made available to Study Group participants. The first is Belgian Retail Data Set [3] that consists of approximately five months of data covering three non-consecutive periods between December 1999 and November 2000. It contains data for $5,133$ Customers, $16,470$ Items and $88,163$ Baskets. The second dataset is Ta Feng retail dataset that consists of four months of data covering the consecutive period between November 2000 and February 2001. It contains data for $32,266$ distinguishable Customers, $2,012$ Sub-Categories and $119,578$ Baskets. For the purpose of this study, Ta Feng dataset is more appropriate for several reasons. This dataset is larger and contains more information, such as it allows items to be arranged according to shoppers rather than to baskets easily and includes the demographic data.

In the rest of the report, Ta Feng data is used for the analysis. Each record in this dataset consists of nine attributes: 1: Transaction date and time 2: Customer ID 3: Age: 10 possible values, 4: Residence Area: 8 possible values, 5: Product subclass 6: Product ID 7: Amount 8: Asset 9: Sales price. Shopping records with the same customer ID and the same shopping date are considered as a transaction or basket. In this report, we only included categories 2, 6 and 7 in order to make the initial analysis simpler.

For the purposes of this study the format of the data had to be changed to the format compatible with Matlab.

4.2 Previous efforts

It is estimated that frequenly bought items account for 20% of all items but data describing purchases of freguently bought items makes 80% of all data. At the same time, rarely bought items account for 80% of all items but only 20% of data describes purchasing history for these items. This 80/20 split leads to the "long tailed" distribution shown in Figure 1. Current recommendation algorithms either ignore the information about rarely purchased items and only keep 80% in doing this, or approximate the item sales distribution by a shorter tailed distribution.

Unilever have previously tried the Bayesian analysis based on conditional probability-based similarity. This is the way of computing similarity between each pair of items i and j by using a measure that is based on the conditional probability of purchasing one of the items given that the other has already been purchased. Unilever have tried several methods of variuos complexity [1], [5] but they did not perform well if the long tail data was included.

Study Group participants also tried cosine similarity method to rank similar items "in the short head" [6]. Unilever had previously applied it to 'short head' only, while we tried applying it to the full dataset. However, this method produces a symmetric similarity matrix, which we believe does not take the information from the long tail into account in a proper manner. To illustrate this point, let us consider two items i and j, and take i from the bulk of the frequently purchased items, while suppose j to be a rarely purchased item from the long tail. Therefore, item i has been puchased significantly more frequently then item j and the number of times that i and j are purchased together is much smaller than them number of times that i is puchased alone. If we now want to calculate the similarity between i and j, then from j's point of view it will be low, because only a small fraction of all purchases of j occurs with i. However, from i's point of view, the similary may be high because it has been puchased so few times and significant fraction of this purchases may have occured with j. Therefore, $\mathrm{sim}(i,j) \neq \mathrm{sim}(j,i)$, and the similarity matrix should not be symmetric. In the next section we investigate how to build an assymetric similarity matrix from a dataset.

4.3 Bipartite networks

In order to produce an assymetric similary matrix, we suggest to use the approach of Bipartite networks [7], [8]. The relevance distribution method was suggested in [9] to project a bipartite network onto a unipartite network. This approach has not been previously examine by Unilever.

We reformulate the problem and, instead of conditional probability formulations, will formulate the problem in terms of bipartite networks.

We used one of the Ta Feng data sets and grouped items by shopper rather than by individual baskets. In the simplest case, we also ignored demographics data. The data matrix may look something like this:

Items

$$\text{Shoppers} \quad \begin{pmatrix} 1 & 2 & 1 & 0 \\ 0 & 3 & 0 & 2 \\ 0 & 1 & 4 & 1 \end{pmatrix} =: M$$

Where raws correspond to shoppers and columns correspond to items and, for example, item 2 was bought 3 times by shoppet B. This matrix contains

all available information but for a large data set it is difficult to work directly with. Bipartite networks provide a simple and intuitive way to reformulate this matrix into a network and to calculate the similarity matrix between items.

Let us draw a simple network, where shoppers A, B, C bought various combinations of items 1, 2, 3, and 4. A connection between a shopper and an item is only made if this shopper bought this item. For example, in Figure 3, shopper A bought items 1, 2 and 3, shopper B bought items 2 and 4, and shopper C bought items 2, 3 and 4.

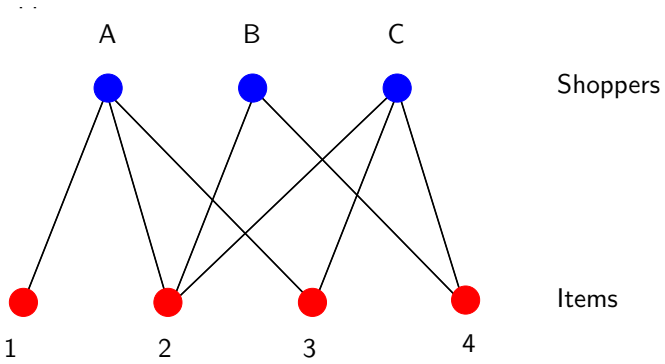

Figure 3: Example of a bipartite network where connection between a shopper and an item is made if the item is bought by the shopper.

Now let us assing a relevance x_i to each item i. For example, we may want to say that $x_i = 1$ if item i is in the basket and 0 otherwise. This initial assignment is shown in Figure 4.

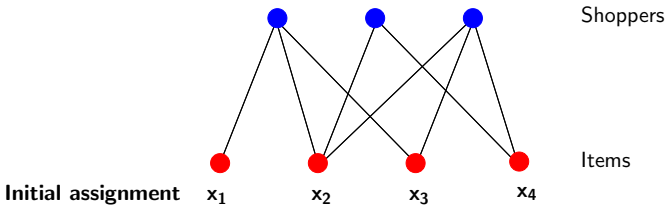

Figure 4: Bipartite network: assigning an initial relevance to each item. $x_i = 1$ if item i is in the basket and 0 otherwise.

We can now traslate this relevance back to the shoppers according to how many shoppers bought the item. Let us calculate relevances y_i (Figure 5.) Shopper 1 bought items 1, 2 and 3, therefore y_1 should be a function of x_1, x_2 and x_3. Item 1 was bougth just once, so the whole x_1 goes toward y_1. Item 2 was bought by three shoppers, so if we assume that relevance is distributed evenly, only $x_2/3$ goes towards y_1. Item 3 was

bought by two shoppers, so $x_3/2$ goes towards y_1. In the simplest case, the translation will be linear, so $y_1 = x_1 + x_2/3 + x_3/2$. Similar calculations give $y_2 = x_2/2 + x_4/2$ and $y_3 = x_2/3 + x_3/2 + x_4/2$.

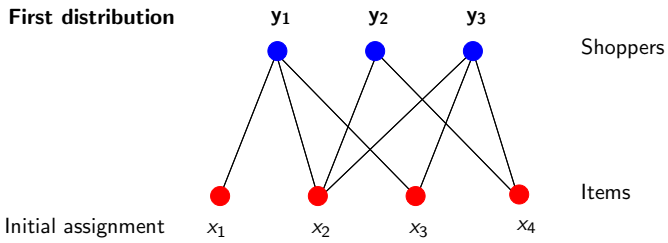

Figure 5: Bipartite network: distribute relevance of each item among its buyers according to how many shoppers bought the item. *E.g* if relevance is distributed evenly, then $y_1 = x_1 + x_2/3 + x_3/2$.

Now we can go one step further and distribute y_i between items again using exactly the same logic for calculating z_i. (Figure 6.) Simple calculations give

$$z_1 = y_1/3 = x_1/3 + x_2/6 + x_3/4,$$
$$z_2 = y_1/3 + y_2/2 + y_3/3 = 7x_2/18 + 5x_4/12 + x_3/3 + 1x_1/3,$$
$$z_3 = y_1/3 + y_3/3 = x_1/3 + 2x_2/9 + x_3/3 + x_4/6,$$
$$z_4 = y_2/2 + y_3/3 = 5x2/18 + 5x_4/12 + x_3/6.$$

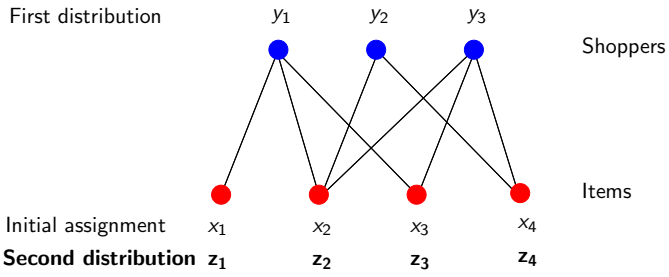

Figure 6: Bipartite network: distribute relevance of each buyer among their purchases again. See text for an example of calculations.

In matrix-vector notation, this can be written as

$$\mathbf{z} = W\mathbf{x},$$

where W is called the relevance (or similarity) matrix. Main properties of this matrix are

- W is asymmetric by construction.

- W_{ij} is the relevance of item j in recommending item i, in particular, if $W_{ij} > 0$, then item i is among most similar items to item j and the value of W_{ij} indicates the degree of similarity between items i and j.
- W is $m \times m$ matrix, where m is the total number of items. In contrast, shopper-item matrix M is $n \times m$.

The recommendation system based on calculating a vector of recommendation scores $\mathbf{z}$ using W may look something like the following:

- W is calculated offline (e.g. once per month) using the historical data.
- In real time, take $\mathbf{x}$ as current content of basket.
- Calculate recommendation scores $\mathbf{z} = W\mathbf{x}$ using precomputed matrix W.
- i.e. Recommend the item with maximal z_i not in basket.
- Easily and quickly calculated 'on the fly'

In the next section the descibed method is applied to the Ta Feng data set.

4.4 Numerical results of application of bipartite networks to Ta Feng data

Following the bipartite networks approach, Study Group participants created a number of programmes in Matlab to build and investigate properties of the relevance matrix W for one of the Ta Feng data sets. Figure 7 shows values of W for 180 items and uses a logarithmic scale to show the range of values of W. The figure confirms that matrix is asymmetric and red values on the top of the figure indicate higher recommending power of items from the long tail.

It is also clear that many values of W are very small. The problem with very small values is that they will translate into very small values of recommendation scores z_i (many values at or near zero) therefore making the job of comparing recommendation scores difficult, as one cannot rank items with zero values. One of the ways to overcome this difficulty is to compute and use higher orders of the matrix W. Using a higher order matrix means that we consider not just similarity between two items i and j (one hop on the grid) but also similarities between groups of items. For example, a second order matrix will capture the similarity information of the type: if items i and j are bought together and items i and k are bought together, what is the similarity between j and k? These will correspond to two hops on the grid.

One can think of a matrix $\tilde{W}$, which will capture all orders of similarity, and then write a Taylor expansion

$$\widetilde{W} = \alpha_1 W + \alpha_2 W^2 + \alpha_3 W^3 + ...$$

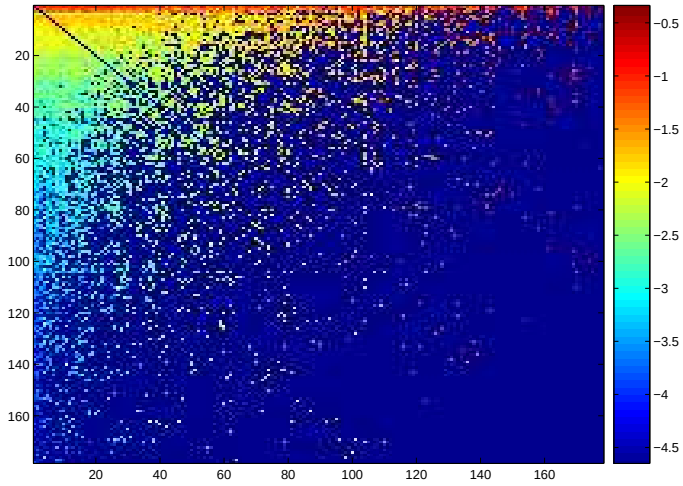

Figure 7: The relevance matrix W for Ta Feng data. Log scale.

The question of computing α_i is a non-trivial one and requires futher research (*e.g.* see [10]) but one can easily compute W^2. In Figure 8, W^2 is plotted on a logariphmic scale. The diagonal elements were substracted before taking a square to avoid assigning a recommendation score of an item to itself. It is clear from this figure that W^2 contains more non-zero elements than W.

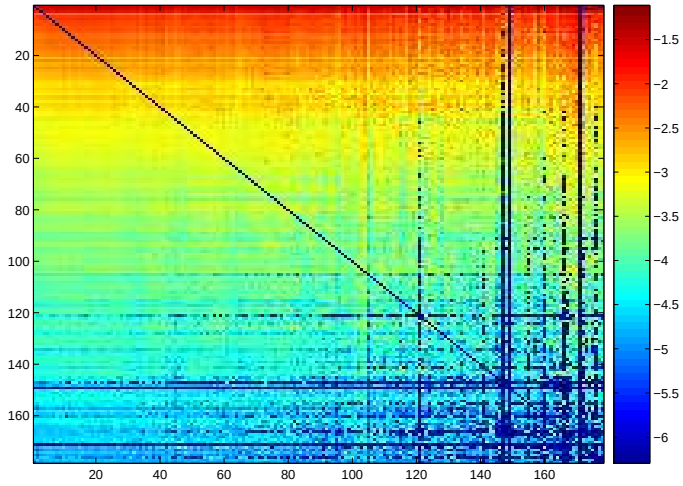

Figure 8: The second order relevance matrix W^2 for Ta Feng data. It is clear that it contains more non-zero values than the first order matrix W.

To illustrate this point more clearly, Figure 9 compares scores of item 500 computed from W and W^2. By scores of item 500 we mean that this would be the scores predicted for all other items by our recommendation

system assuming that our basket contains only item 500. To make things clearer, we sorted the scores from smallest to largest, and therefore 500 will not be an item number in the Ta Fend dataset but rather the 500th most popular item. Using W clearly prevents accurate ranking of most items as the scores for them are very small. In contrast, using W^2, gives a clear ranking curve. Therefore, using more hops, one can rank more items.

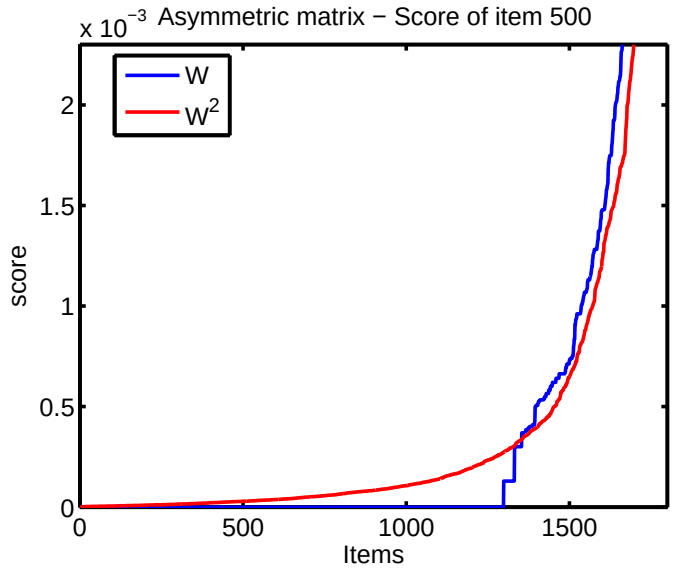

Figure 9: The score of 500*th* item. Blue line: from the first order matrix, Red line: from the second order matrix. It is clear that we can rank more items using the second order matrix.

In practice, the order of W that needs to be computed will depend of a data set and can be determined by empirical testing. As we suggest for W to be computed offline, this computational expense should not be an obstacle for the method.

It is also interesting to compare symmetric W with asymmetric W. This comparison is shown in Figure 10. On top we shown W matrices computed using cosine similarity for the symmetric W and relevance distribution for the asymmetric W. On the bottom of the figure, there are scores of item 500. The scores here are sorted by an item number, *i.e.* the items are listed from the most popular to the least popular. Surpisingly, the asymmetric score does not give a bulk of recommendations from the long tail while symmetric scores seem to perform better in recommending items from the long tail. This is because a symmetric similary matrix tends to eliminate recommendations of very frequent items, as these items will tend to be recommended only if other frequently purchased items are in the current basket. Therefore, in practice, it would be useful to devise a method, that produces an asymetric similarity matrix and at the same time tends to

eliminate recommendations of frequently bought items. This will combine advantages of both symmetric and asymmetric approaches.

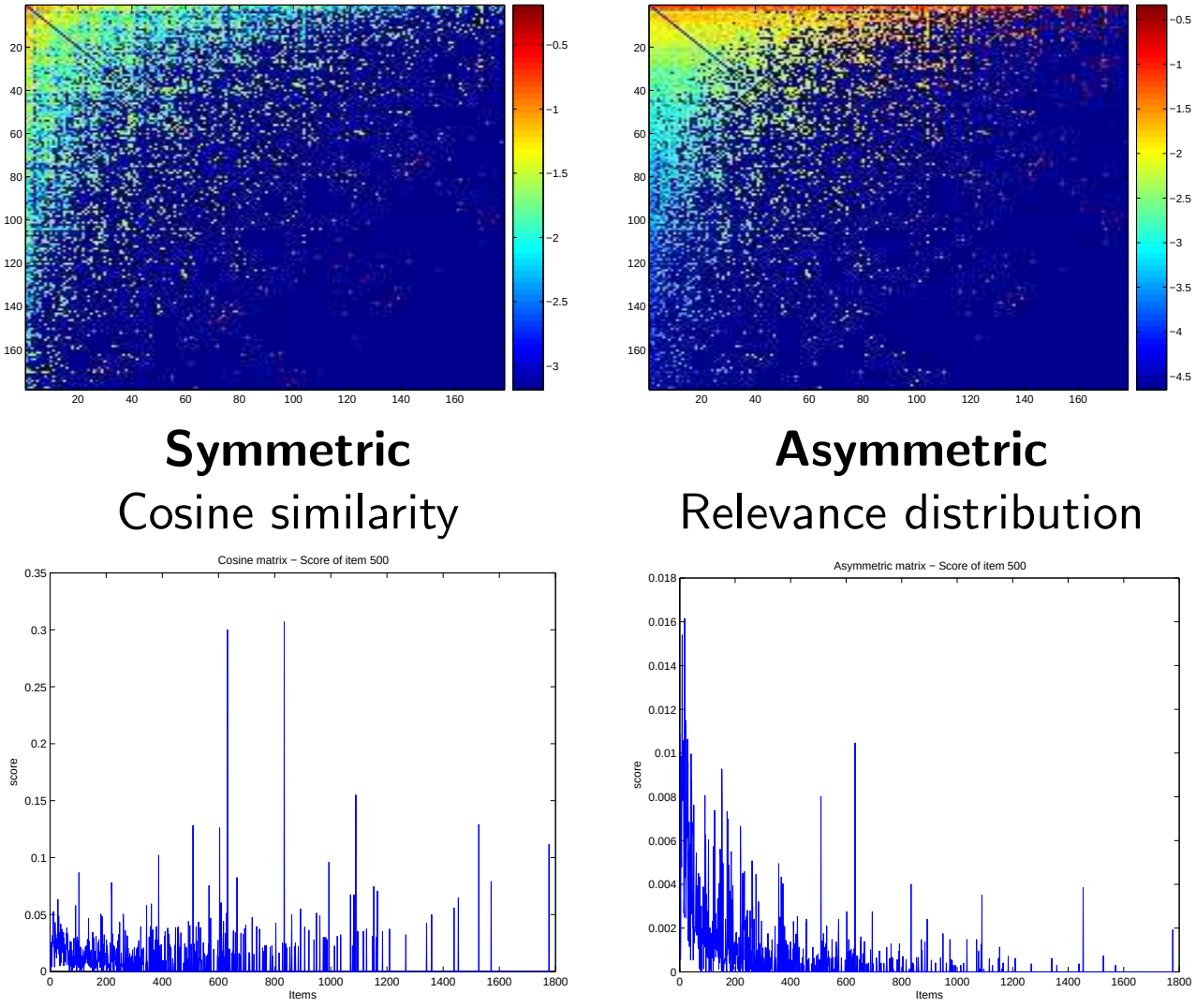

Figure 10: Graph comparing cosine similary matrix (symmetic) with relevance matrix W. Graphs on top show the full matrix. Graphs on the bottom are the scores for $500th$ item. See discussion in the text.

5 Final remarks

5.1 Suggestions for data collecing

During the Study Group discussions, various suggestions have been made as to what data would be useful to have to improve a recommendation system. They are summarised below.

- It would be useful to record refused recommendations and incorporate this information into designing a recommendation algorithm and building a customer profile.
- Some systems (*e.g.* Amazon) record items examined (viewed) as well as items purchased and it may be useful to include this browsing history into a recommendation algorithm.
- Additional demographic data may be extremely useful as it will allow the advantage of using of a social network appoach in designing a recommendation algorithm.
- It may be useful to try explicitly excluding popular items from recommendations to make a better use of data from a long tail.

5.2 Conclusions

Overall conclusions of a week-work of the Study Group are

- Reformulating the problem as a bipartite network provides useful insights into the data structure and approaches for recommendation algorithms. There is a huge body of literature on application of bipartite networks to recommendation systems, which Unilever should take advantage of.
- The key point in designing a recommendation algorithm is a careful determination a similarity matrix. Its key properties are that it should be asymmetric and higher than the first order.
- "Relevance distribution" gives Unilever a method to test on their data.

5.3 Future work

There are a number of the next steps that Study Group contributors believe would be useful for Unilever to take

- Test and optimise first-order asymmetric methods on Unilever data (*i.e.* how could one better construct W?).
- Formalise incorporation of higher-order corrections.
- Make recommendations for a returning shopper, *i.e.* build better customer profile by collection additional data (*e.g.* browing history).
- Inlcule time-dependence by taking into accoung changing customer preferences.

- Incorporate demographic data into models.
- It may be appropriate for Unilever to consider setting up a CASE studentship to investigate and test the application of bipartite networks to their recommendation system.
- New papare are actively appearing in the literature, for example something that came out after the study group [14]

References

[1] C. M. Sordo-Garcia, M. B. Dias, M. Li, W. El-Deredy and P. J. G. Lisboa, *Evaluating retail recommender systems via retrospective data: Lessons learnt from a live intervention study*, in proceedings of the International Conference on Data Mining, DMIN'07, pp 197 - 203, 2007.

[2] Chris Anderson, *The Long Tail: Why the Future of Business is Selling Less of More*, published by Hyperion, 2006.

[3] Brijs T., Swinnen G., Vanhoof K., and Wets G., *The use of association rules for product assortment decisions*: a case study, in: Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining, San Diego (USA), pp. 254-260, 1999.

[4] Chun-Nan Hsu, Hao-Hsiang Chung and Han-Shen Huang, *Mining sparse and skewed transaction data for personalized shopping recommendation*. Machine Learning, 57(1-2):35-59, Special Issue on Data Mining Lessons Learned, October-November, 2004.

[5] Li et al., in proceedings of the ACM Conference on Recommender Systems, 2007.

[6] Mukund Deshpande and George Karypis, *Item-Based Top-N Recommendation Algorithms*, ACM Transactions on Informative Systems, Vol. 22, No. 1, pp. 143-177, 2004.

[7] M. E. J. Newman, *The structure and function of complex networks*, SIAM Review 45, 167-256 (2003).

[8] Handbook of Graph Theory, edited by Jonathan L. Gross and Jay Yellen, hardcover, CRC Press, (2003).

[9] Tao Zhou, Jie Ren, Mat Medo and Yi-Cheng Zhang, *Bipartite network projection and personal recommendation*, Phys. Rev. E 76, 046115, 2007.

[10] E. A. Leicht, Petter Holme, and M. E. J. Newman, *Vertex similarity in networks*, Physical Review E 73, 026120, 2006.

[11] M. E. J. Newman, *Finding community structure in networks using the eigenvectors of matrices*, Phys. Rev. E 74, 036104, 2006.

[12] V. Blondel, A. Gajardo, M. Heymans, P. Senellart, and P. Van Dooren, *A Measure of Similarity between Graph Vertices: Applications to Synonym Extraction and Web Searching*, SIAM Review, Vol. 46, No. 4, pp. 647666, 2004.

[13] Zan Huang, Xin Li, Hsinchun Chen, *Link prediction approach to collaborative filtering*,Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries, USA, pp. 141-142, 2005.

[14] Tao Zhou et al., *Ultra accurate personal recommendation via eliminating redundant correlations*, http://arxiv.org/PS_cache/arxiv/pdf/0805/0805.4127v1.pdf