DOMAIN ABSTRACTION OF HIGHLY CORRELATED PAIRS TO RECOMMEND IN THE LONG TAIL

MINH THU TRAN NGUYEN, JEAN DANIEL ZUCKER

ABSTRACT

Among difficulties encountered by modern shopping recommenders is the long tail shape of sold items also related to cold-start issues. Various approaches including content-based recommendations attempt to overcome this problem that has serious impact on the accuracy of recommendations especially when new products are continuously added to the catalogue. This paper investigates the use of an algorithm to search for highly correlated pairs between abstractions of items. The advantage of this approach is evaluated on the basis of real data showing better results compared to an approach only based on the concrete pairs of items. Using rigorous protocols such as Given-n, experimental results show significant improvement in both the recommendation accuracy and the recommendation of products in the long tail.

Keywords. Knowledge Discovery, Mining Correlated Pairs, Recommender Systems.

1. INTRODUCTION

In recent years, recommendation systems have been extensively explored in scientific literature, in merchant and community sites [11,13]. On e-commerce sites [12, 11], the problem can be reformulated as a problem of predicting the k target items with the greatest chance of being selected by the user U based on p items $X_1 \dots X_p$ already selected by her/him. In practice, p is small and k is often three [2]. In other words, for any item $X_i$we seek the probability $P(X_j|X_1, \dots, X_p, U)$ [2]. There are many statistical models used to discover this probability from historical data: association rules [5], highly correlated pairs [15,10], collaborative filtering [1], etc. One of the simplest models is based on the support of the association $X_i \rightarrow X_j$by the formula:

$$P(X_j|X_i) = \frac{N(X_i, X_j)}{N(X_i)} \; . \tag{1}$$

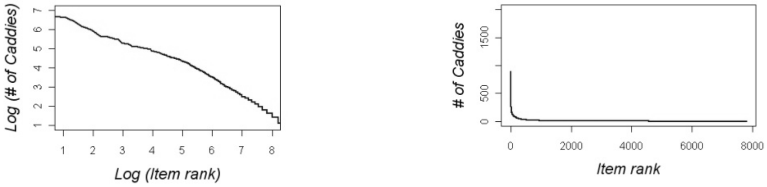

Figure 1. Empirical distribution of the popularity of items in shopping carts showing a “long tail” (left) and its characteristic curve in a log-log scale (right). In these curves, the ordinate represents the number of transactions (or caddies) containing a given item and the abscissa represents the rank of items per decreasing number of transactions

Although widely used, strong association rules (or correlated pairs) based on these measures may prove unsatisfactory in practice because supporting data is often missing and are biased (only partially representative of the real distribution). Indeed, the distributions of items in transactions have a "long tail" following a law also known as the Pareto law [9]. This is true in the case of the e-commerce data we have used, shown in figure 1. Recommender systems often recommend products in the head, and on the other hand when products from the tail are selected (including never sold or new products) no personalized recommendation can be made.

In this paper, both the problem of increasing recommendation in the long tail and the problem of accuracy are addressed. The approaches proposed rely on an existing item taxonomy [6], which is very detailed or very basic. Frequently, in ecommerce there are categorical attributes that segment items into groups of items [2]. For instance, a merchant site offering some "Clothes" items, will have categories such as Outwear, Pants, Shirts (See figure 2). At the level of categories ("abstract items") the long tail effect is weaker. This article proposes an approach based on the research of highly correlated pairs between abstract items and a framework to use in estimating the probability $P(X_j|X_1,\ldots,X_p,U)$.

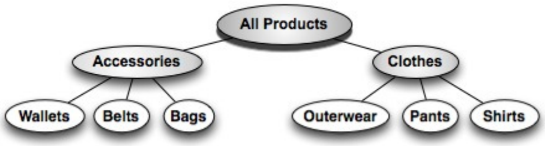

Figure 2. An example of a product taxonomy (From[6])

2. STATE OF THE ART

2.1. Item-based Recommendation Systems

The problem of estimating the co-occurrence of two items is the heart of recommendation systems. Among the choices for calculating the strength of association, [7] proposes to modify the calculation of the formula 1 by another one taking into account the number of items per transaction. The conditional probability $P(X_j|X_i)$ is the ratio of the number of transactions containing the co-occurrences of X_i, $X_j|$ $(N(X_i,X_j))$ and the total number of transaction containing $X_i(N(X_i))$. This estimate means to increase the strength of associations for items that appear in transactions having few items. The results show that when applying real data, we can improve the quality of the estimator. This work highlights the importance of the estimator of the co-occurrence without addressing rare items issues.

2.2. Recommendation system supplying long tail problem

The long tail problem is related to the "rare item issue", i.e. the lack of ratings or purchases is able to make valuable recommendation. To provide a good method to leverage the long tail, Y. Park [9] suggests splitting the whole itemset into two parts: the head and the tail parts. Then recommendations are built on each part separately. Another method to address the long tail problem is implemented in HyPAM (Hybrid Poisson Aspect Model) [4]. This model is able to solve the issues of data skew and sparsity.

2.3. Association rules for recommendation systems

In the field of recommendation systems, several approaches have focused on the problems of mining rare data [14]. The original approach consists of searching association rules without using a single minimum support, but rather using multiple ones depending of the data considered [5, 3, 14]. This approach has been proved useful for discovering rare associations that are not embedded in rules which are more important, but have less support. The use of abstraction items has been already investigated by Han and Fu [3]. They have extended the scope of the study of mining association rules from single level to multiple concept levels (different granularity) and studied methods for mining multiple-level association rules from large transaction databases. However, association rules are inefficient for collaborative recommendation when they mine many rules that are not relevant to a given user. Rules mined for specific target user reduce the time required for the mining process with the minimum support for each user corresponding [8].

2.4. The Search for Highly Correlated Pairs

The search for correlated pairs permits the estimation of the probability $P(X_j|X_i)$ based on the correlation of X_i with X_j. The difficulty often comes from the large number of pairs: due to the lack of memory space, all of them cannot be stored in memory [16]. It is therefore necessary to devise heuristics that both ensure that most correlated pairs are identified, and that a limited number of pairs only is tested. Several approaches support the use of an upper bound of Pearson correlation to prune the pairs that cannot be highly correlated [16]. In fact in the presented AbREC algorithm we rely on another algorithm that "TOP-COP" [15] to search efficiently for highly correlated pairs of concrete items.

3. DEFINITION AND ALGORITHM

A general view of the proposed approach is shown in Figure 3 below. Based on this abstracted transactions, a mining algorithm may be used to find abstract correlated pairs or associations rules. These pairs or rules are then in turn transformed in concrete association rules or concrete correlated pairs by instantiating the abstract items by one of its elements.

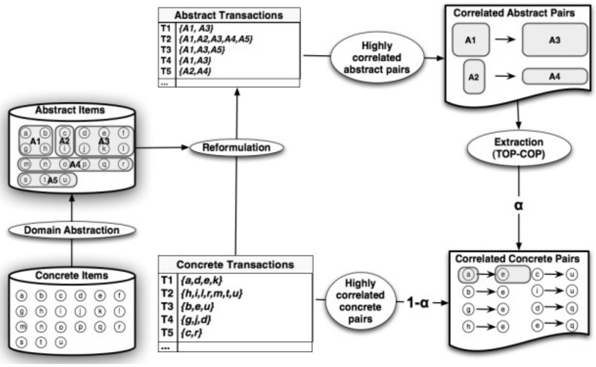

Figure 3. A general view of the abstraction process used to discover highly correlated concrete pairs of items. The Transactions table is abstracted in an abstract Transaction table where each item is replaced by its abstraction. Transaction T1={a,d,ek} is abstacted in {A1,A3} as A1 abstracts a and d,e,k are all abstracted by A3

3.1. Highly correlated abstract pairs

In recommendation systems, items and transactions are represented by:

1. A set of N items X_1 ... X_n data set N which contains their identifier and characteristics.
2. A set T of transactions in the form $(T_i, \{X_{i1}, \dots, X_{ip}\})$ $X_{ij} \in X$. Each transaction T_i is defined as a subset of T. $T\ (X_i)$ denotes the subset of T of all transactions that contains the item X_i .

We consider the notion of abstract items to describe particular categories or group of items. An “abstract item” A_i is dually represented by its description and its extension $A_i = \{X_{i1}, \dots, X_{ip}\}$. This notion of abstract item corresponds to a domain abstraction that is widely used in the field of Abstraction [17]. Using this definition, we may define the support of an “abstract item” A_i as the ratio of the number of transactions which contain the abstract item A_i $N(A_i)$, and the total number of transactions in the database (N). The value of $N(A_i)$ is the number of transactions in which at least one of the concrete items that belongs to A_i appears.

$$Support\,(A_i) = \frac{N(A_i)}{N} = \frac{N\big(OR(X_{i1}, \dots, X_{ik})\big)}{N} \quad . \tag{2}$$

3.2. Highly Correlated Abstract Pairs to improve recommendation.

Finding some highly correlated abstract item pairs is not sufficient (cf. Figure 3). So, we propose to define an algorithm that lists the TOP-K items (in practice K = 3 [13]) combining the information from correlated pairs of concrete items and those correlated pairs of abstract items at the same time. The approach we propose is, therefore, to compute a new estimate of the confidence of the association rule $X_i \to X_j$ called Conf$\alpha(X_i \to X_j)$ that combines the confidence of the concrete confidence Conf$(X_i \to X_j)$ and an abstract confidence of $A_i \to A_j$; which A_i (resp. A_j) is the domain abstraction of X_i (resp X_j). It reads:

$$P\alpha(X_j|X_i) = (1-\alpha)P(X_j|X_i) + \alpha P(A_j|A_i) \tag{3}$$

$$P\alpha(X_j|X_i) = (1-\alpha)\frac{N(X_i, X_j)}{N(X_i)} + \alpha\frac{N(A_i, A_j)}{N(A_i)} \tag{4}$$

for α = 0, we have a system purely concrete that does not use abstraction, and for α = 1, the system is purely based on the abstract highly correlated pairs.

3.3. The AbsTopKα algorithm

The algorithm AbsTopKα that we propose is a form of discretisation of the heuristics presented above. We consider the Top-K(1 − α) items outcome from the recommendation concrete (by pairs X_iX_j) and the Top-Kα pairs outcome from the abstract pairs of items. In practice, K is often chosen to be 3[13]) and for $\alpha = 2/3$ this amounts to 2 items from the concrete pairs et 1 item from the abstract pairs. The principle of the algorithm is as follows:

Algorithm 1 The AbsTopKα algorithm

```
1.  Input : a set $T$ of transactions on items $X_i$
2.  Output: the top $K$ items to be recommended.
    (a) For each $X_i$ , build its associated abstract item $A_i$
    (b) For each $A_i$, build the list of the top $K\alpha$ most correlated abstract
```

```
pairs items
                 $(A_i, A_{ij})$; $L(A_i) = \{A_{i1}, \dots, A_{iK\alpha}\}$
    (c) For each item $X_i$
    i.   Find the top $K(1-\alpha)$ most correlated concrete pairs of items $(X_i, X_{ij})$;
                 $L = \{X_{i1}, \dots, X_{iK(1-\alpha)}\}$
    ii.  $L = L \cup AbsExtract(X_i, L(A_i))$
    iii. Return the list $L$ of $K$ items associated with $X_i$
```

4. RESULTS AND EVALUATION

4.1. Dataset

To test the AbsTopKα and AbsExtract algorithms, like most authors, we used a database of real world data that cannot be made public for confidentiality reasons. Indeed, unlike the artificially generated data, this data corresponds to a volume of transactions that is found only very rare in data available in the public domain. There are several publicly available database to test recommender systems but most of them correspond to products (such as movies) associated to user ratings.

```
Algorithm 2 The AbsExtract algorithm
1. Input : a concrete item $X_i$, a set of $K'$ abstract items associated to the
   abstract item of $X_i$, $L(A_i) = \{A_{i1}, \dots, A_{iK'}\}$
2. Output: the top $K'$ items $X_{i1}, \dots, X_{iK'}$ to recommend based on abstract
   rules.
3. Algorithm
    (a) Find all concrete items $X_{ij}$ whose abstract items belong to $L(A_i)$
    (b) For each $X_{ij}$ compute the correlation (resp. support) of pair
        $(X_i, X_{ij})$
(c) Return the top $K'$ items $X_{i1}, \dots, X_{iK'}$ such that associated pair $(X_i, X_{ij})$ has
    maximum correlation (resp. support)
```

The data we are interested in are 0-1 Data (products that have been bought or not by users). We used real data from a merchant site that consists of transactions of books, cds, clothes and gifts sales. The dataset includes *shopping date*, *customer id*, *product id* and *the number of purchase* in each shopping record. And table 1 is the summary of the main characteristics of the database we have used.

Table 1. Characteristics of the database

(training data set: 2007; test data set: January and February of 2008)

Properties	2007	Jan and Feb of 2008
# of items (concrete)	9332	3943
# of items Abstracts by expert	451	365
# of Transactions	30009	5097
# average of items per transaction	2,53	2,65

4.2. Experimental analysis

The issue of evaluating of recommendation systems is one of the most difficult issues. However, some studies suggest several evaluation protocols that can gauge the performance of estimators [13, 2, 4] and are of the statistic learning cross-validation test type. We use *Given-P* method [4] for evaluating the system, the principle of the test method is to consider all of the transactions that have at least $N+1$ items. The first N items of the transaction are used to predict the presence of an item X_j presented or not in the remaining items of the transaction.

In table 2, we summarize the results of the evaluation of the AbsTopKα algorithm for $K=3$ and the value of α is varied from 0 to 1. We have used the data of the year 2007 to find the correlated pairs and that of the first two months of 2008 to check their effectiveness. The first remark is that the number returned by Given-1 is very small. This is also the case in the results of Hsu et al.[4]. This small number does not reflect the quality of recommendations but the result obtained by a particular stringent test which is implemented by Given-1. The second remark from table 4.2 is that the recommendations derived from the AbsTopK algorithm are better for alpha greater than 0 (the pure abstract system or combination between concrete and abstract). And the best result is obtained by alpha = 2/3 where we have used two abstracts and one concrete. There are 1051 (26,65%) new items in two months and we have about 10% good recommendation items for the new by our method.

Table 2. With $K=3$ and differents value alpha. For $\alpha = 2/3$, the result of "maximum of theory" is 1030 good recommended items (14,56% new items)

α	#successful by Given-1 (Maximum of theory)		# successful by Given-1 (AbsExtrait algorithm)	
	All items	New items	All items	New items
0	541	0	541	0
1/2	846	10,53%	656	7,62%
2/3	1030	14,56%	669	11,66%
1	984	18,19%	477	18,45%

The third remark will be illustrated in table 3 providing some insight on the type of items that are recommended when α is increasing. The number of items recommended from the long tail is increasing in total and percentage.

Table 3. The accuracy of the algorithm in the Head and Long Tail. The parameter alpha allows to control the proportion of abstract rules that are used. Using abstract pairs does increase the number of good recommendation in the long tail up to 44%

α	# items from concrete pairs	# items from abstract pairs	# good reco. in the head	# good reco. in the long tail	Gain in the long tail with abstract pairs
0	3	0	279	262	1
1/2	2	1	468	378	1.44
2/3	1	2	715	315	1.20
1	0	3	657	327	1.25

5. CONCLUSION

This paper has attempted to address the issue of improving recommendation in the long tail. We have proposed an algorithm that relies on the search of abstract pairs of items. To evaluate the interest of our approach we evaluated the gain compared to a direct approach only based on the pairs of concrete items. We used a "Given-N" methodology to evaluate and quantify the gain. This reached 44% in our particular dataset of real transactions. The gain is made by recommending products that were in the long tail part, supporting the fact that using abstraction improves recommending in the long tail. However, this result also needs to be contrasted with studies of other databases, we are currently analysing if this framework could be implemented on the Netflix and could be used in the public data provided by Hsu and al. [4]. The present approach to search for abstract association rules means to improve the accuracy of recommendations on concrete items. However, if the domain abstraction is not relevant, the performance can be worse. Our claim is not that combining abstract and concrete pairs will always improve the results but rather that based on appropriate product hierarchy, performance may be improved significantly.

Acknowledgments. The authors wish to thank Edith Nuss for her help in collecting the data used to test the AbsTopK algorithm. The authors also thank the authors of the TOP-COP algorithm [15] (Xiong et al.) who have sent us their code.

REFERENCES

1. Chen A., McLeod D. - Collaborative filtering for information recommendation systems, Encyclopedia of Data Warehousing and Mining, Idea Group, 2005.
2. Dias M. B., et al. - The value of personalised recommender systems to e-business: a case study, In: RecSys '08. pp. 291-294. ACM, NY, USA, 2008.
3. Han J., Fu, Y. - Mining multiple-level association rules in large databases, IEEE Trans. on Knowl. and Data Eng. **11** (5) (1999) 798-805.

4. Hsu C. N., Chung H. H., Huang H. S. - Mining skewed and sparse transaction data for personalized shopping recommendation, Mach. Learn. **57** (1-2) (2004) 35-59.

5. Hu Y. H., Chen Y. L. - Mining association rules with multiple minimum supports: a new mining algorithm and a support tuning mechanism, Decis. Support Syst. **42** (1) (2006) 1-24.

6. Hung L. P. - A personalized recommendation system based on product taxonomy for one-to-one marketing online, Expert Syst. Appl. **29** (2) (2005) 383-392.

7. Li M., Dias B., et al. - A probabilistic model for item-based recommender systems. In: RecSys ’07, ACM, USA, 2007, pp. 129-132..

8. Lin W., Alvarez S. A. - Efficient adaptive-support association rule mining for recommender systems, Data Mining and Knowledge Discovery **6** (2002) 83-105.

9. Park Y. J., Tuzhilin A. - The long tail of recommender systems and how to leverage it, In: RecSys., ACM, 2008, pp. 11-18.

10. Roy S., Bhattacharyya D. K. - Scope: An efficient one pass approach to find strongly correlated item pairs, ICIT ’08 0, 2008, pp. 123-126.

11. Sarwar B., et al. - Analysis of recommendation algorithms for e-commerce, In: EC’00, ACM, USA, 2000, pp. 158-167.

12. Schafer J. B., Konstan J. A., Riedl J. - E-commerce recommendation applications, Data Min. Knowl. Discov. **5** (1-2) (2001) 115-153.

13. Sordo-Garcia C. M., et al. - Evaluating retail recommender systems via retrospective data: Lessons learnt from a live-intervention study, In: DMIN. CSREA, 2007.

14. Szathmary L., Napoli A., Valtchev P. - Towards rare itemset mining, In: IC-TAI’07, Greece, IEEE Computer Society, 2007, pp. 305-312.

15. Xiong H., Brodie M., Ma S. - Top-cop: Mining top-k strongly correlated pairs in large databases. In: ICDM ’06, IEEE, DC, USA, 2006, pp. 1162-1166.

16. Zhang J., Feigenbaum J. - Finding highly correlated pairs efficiently with powerful pruning, In: CIKM ’06, ACM, USA, 2006, pp. 152-161.

17. Zucker J. D., Saitta L. (Eds.) - Abstraction, Reformulation and Approximation, 6th International Symposium, SARA 2005, Airth Castle, Scotland, UK, Proceedings, LNCS, Vol. 3607, Springer, 2005.

Received June 16, 2010

Address:

Minh Thu Tran Nguyen[1, 2, 4] , Jean Daniel Zucker[2, 3],

[1]LimBio, University Paris 13, F93017 - Bobingy, France.

[2]UMI 209 - UMMISCO, Centre IRD France Nord, F93143 - Bondy, France.

[3]MI 209 - UPMC University Paris 6, Paris – France.

[4]SI, Institut de la Francophonie pour l’Informatique (IFI), Hanoi, Vietnam.