A probabilistic model for item-based recommender systems

A Probabilistic Model for Item-Based Recommender Systems

Ming Li and Benjamin Dias
Unilever Corporate Research
Sharnbrook, Bedfordshire, UK
{Ming.x.Li, Ben-jamin.Dias}@Unilever.com

Wael El-Deredy
School of Psychological Sciences
University of Manchester
Wael.El-deredy@manchester.ac.uk

Paulo J. G. Lisboa
School of Computing and Mathematical Sciences
Liverpool John Moores University
P.J.Lisboa@livjm.ac.uk

ABSTRACT

Recommender systems estimate the conditional probability $P(x_j|x_i)$ of item x_j being bought, given that a customer has already purchased item x_i. While there are different ways of approximating this conditional probability, the expression is generally taken to refer to the frequency of co-occurrence of items in the same basket, or other user-specific item lists, rather than being seen as the co-occurrence of x_j with x_i as a proportion of all other items bought alongside x_i. This paper proposes a probabilistic calculus for the calculation of conditionals based on item rather than basket counts. The proposed method has the consequence that items bough to-gether as part of small baskets are more predictive of each other than if they co-occur in large baskets. Empirical re-sults suggests that this may result in better take-up of per-sonalised recommendations.

Categories and Subject Descriptors: H.3.3 [Information Systems]: Information Search and Retrieval

General Terms: Algorithms, Experimentation, Performance.

Keywords: shopping basket recommendation, naive bayes classifier, performance evaluation, popularity-based ordering.

1. INTRODUCTION

The estimation of class-conditional probabilities lies at the heart of personalised recommender systems. It is normally the case that co-occurrence of item pairs is empirically estimated by the mean proportion of baskets in which the two items appear together. We shall refer to this as **basket-based** estimation. However, this loses track of the context of remaining items purchased, in particular whether the observed pairing is one of a few or very many item pairs in the relevant baskets in the user-item data matrix. This pa-per proposes an alternative modeling framework where each item pairing is counted as a proportion of the total basket size, that is to say the frequency of paired occurrences for item x_j with item x_i is normalised by the total number of items purchased alongside x_i. This we term **item-based** estimation. It requires a re-formulation

of the probabilistic calculus for the calculation of $P(x_j|x_i)$, which is discussed in the next section. Section 3 illustrates the proposed methodology by reference to the simplest estimator of the conditional, the Naïve Bayes (NB) recommender system, and is followed by a brief evaluation of the expected uptake of recommendations estimated from historical data in a real-world application. For con-venience, the NB classifiers using basket-based and item-based es-timates will be referred to NB(baskets) and NB(items), respectively, throughout the rest of the paper.

2. AN ITEM-BASED PROBABILISTIC CALCULUS

2.1 Basket-based and Item-based Probability Estimation

A probability estimate represents the expected frequency of oc-currence of an *event* of interest taken over a defined space of possibilities. In collaborative filtering there are two different in-terpretations of the base population from which inferences about item purchases are made, which may be the number of purchasing episodes (i.e. baskets) or the totality of individual items purchased [5, 7, 3, 6].

1. *Basket-based probability estimator*. In this estimate, which is the most commonly used in practice [5, 7, 6], the count-ing unit is a basket. Consequently, the prior probability $P(C_i = x_i)$ is empirically estimated to be the ratio be-tween the number of baskets containing x_i (i.e. $N_D(x_i)$) and the total number of baskets; the conditional probabil-ity $P(F_j = x_j|C_i = x_i)$ is similarly given by the number of baskets in which (x_i, x_j) co-occur divided by the total number of baskets that contain x_i.

$$P_D(C_i = x_i) = \frac{N_D(x_i)}{|baskets|} \quad (1)$$

$$P_D(F_j = x_j|C_i = x_i) = \frac{n(x_i, x_j)}{N_D(x_i)} \quad (2)$$

2. *Item-based probability estimator*. The proposal is to count each basket item as an event. In this model, the indepen-dent random variable becomes the decision to purchase each individual item, rather than the basket as a whole. Thus, $P(C_i = x_i)$ is the ratio between the number of items bought alongside x_i and the total number of possible item-pairs in all the baskets put together, which is termed |*item pairs*|. The conditional probability $P(F_j = x_j|C_i = x_i)$ is now calculated as the number of times that item x_i pairs-up with

item x_j divided by the total number items pairs purchased in the same baskets as x_i.

$$P_F (C_i = x_i) = \frac{N_F (x_i)}{|item\ pairs|} \quad (3)$$

$$P_F (F_j = x_j | C_i = x_i) = \frac{n(x_i, x_j)}{N_F (x_i)} \quad (4)$$

where $N_F (x_i) = \sum_{j:j=i} n(x_i, x_j)$ and $|item\ pairs| = \sum_i N_F (x_i)$.

Note that the basket-based estimate does not discriminate between items that have co-occurred with different numbers of other items. In contrast, the item-based approach captures the item co-occurrence information in the feature, or item, space.

Furthermore, for a probabilistic calculus to be complete and con-sistent, the following three conditions must be met:

- $\sum_i P_F (x_i) = 1$: normalisation of individual probabilities

- $\sum_{j:j=i} P_F (x_j | x_i) = 1$: normalisation of conditional prob-abilities

- $\sum_{j:j=i} P_F (x_i, x_j) = P (X_i)$: calculation of marginal prob-abilities

The first two identities follow trivially from the stated definitions. The third identity holds as

$$\sum_{j:j=i} P_F (x_i, x_j) = \sum_{j:j=i} \frac{n(x_i, x_j)}{|item\ pairs|} = \frac{\sum_{j:j=i} n(x_i, x_j)}{|item\ pairs|}$$

$$= \frac{N_F (x_i)}{|item\ pairs|} = P_F (C_i = x_i) = P (X_i)$$

2.2 An Illustrative Example

To illustrate the difference between the two probability estimations, consider the example given in Table 1 in which there are eight baskets (i.e. $|baskets| = 8$) and 23 basket items in total across all the baskets (i.e. $|item\ pairs| = \sum_i N_F (x_i) = 48$). With the basket-based probability estimation:

$$P_D (a) = \frac{n(a)}{|baskets|} = \frac{4}{8}, P_D (d) = \frac{6}{8}$$

$$P_D (d|a) = \frac{n(d, a)}{n(a)} = \frac{2}{4}, P_D (a|d) = \frac{2}{6}$$

Similarly, with the item-based estimation,

$$P_F (a) = \frac{\sum_{x=(b,c,d)} n(x, a)}{|item\ pairs|} = \frac{8}{48}, P_F (d) = \frac{13}{48}$$

$$P_F (d|a) = \frac{n(d, a)}{\sum_{x=(b,c,d)} n(x, a)} = \frac{2}{8}, P_F (a|d) = \frac{2}{13}$$

It is easy to show that both probability estimations satisfy Bayes theorem in this example, i.e. $P (d|a) = P (a|d)P (d)/P (a)$.

3. APPLICATION TO NAÏVE BAYES RECOMMENDER SYSTEMS

The recommendation list is usually generated by ranking all the non-purchased items by the corresponding posterior probabilities $P (C_i = x_i | F)$, and then recommending the *topN* most probable

Table 1: Example of Shopping Baskets.

Basket-I D	F_1	F_2	F_3	F_4	F_5	F_6
1				d	e	f
2				d	e	f
3			c	d		
4		b	c	d		
5	a	b	c	d		
6	a	b	c	d		
7	a	b				
8	a	b				

ones[1]. In a NB recommender system, the calculation of $P (C_i = x_i | F)$ can be simplified as follow:

$$P (C_i = x_i | F) \approx P (C_i) \prod_{j:j=i}^{m} P (F_j | C_i)$$

$$= \frac{\prod_{j:j=i}^{m} n(x_i, x_j)}{N (x_i)^{m-1}}$$

The above equation implies that NB(baskets) and NB(items) will generate the same recommendation lists if they rank $N (x_i)$, i.e. $N_D (x_i)$ for NB(baskets) and $N_F (x_i)$ for NB(items), in the same order. Otherwise, the two lists will be different.

In a preliminary experiment, we evaluate the impact of $N (x_i)$ on the ranking position of item x_i in the recommendation list. The result is shown in Figure1. As can be seen, among 210 active categories, there are only around 23 categories with the same ranking position in both cases.
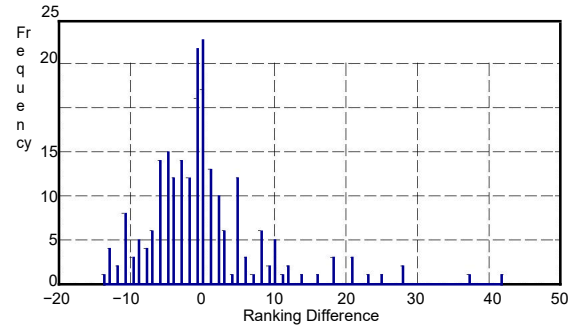

Figure 1: Histogram of the difference between the ranks of the prior probability generated by NB(baskets) and NB(items) i.e. $Rank(P_D (x_i)) - Rank(P_F (x_i))$. Among 210 active categories in total, there are only around 23 categories with t same ranking position in both cases.

4. EMPIRICAL PERFORMANCE ESTIMATION

In order to compare NB(baskets) and NB(items), LeShop provided us with access to link-anonymized shopping basket data from actual historical records comprising individual transactions, as previously described in [6]. We use data covering six consecutive months for training, and ran tests on the following seventh month.

---

[1]We notice that the authors in [7] calculate the purchase likelihood in a way similar to $\frac{P (C_i | F)}{P (C_i | F) + P (C_i | \bar{F})}$. In our pre-liminary experiment, however, we did not observe any ad-vantage of such an approach in terms of recommendation performance.
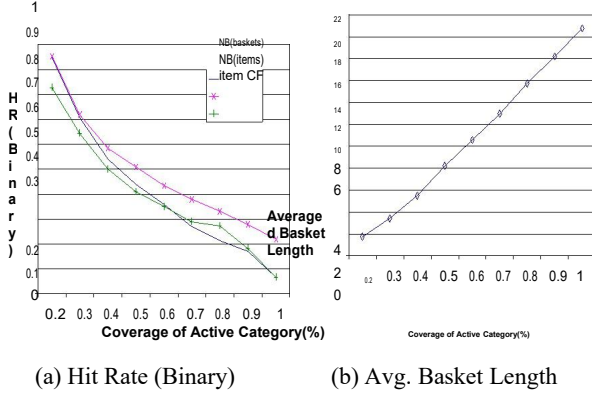
(a) Hit Rate (Binary)　　　　(b) Avg. Basket Length

Figure 2: Performance comparison with increasing coverage of the active categories. *testset* = LeShop.

After aggregating the training baskets[2], there are around 210 active categories and 40,000 baskets. The test data are not aggregated. We also used data from an anonymous Belgium retailer [2] in order to verify our results. However, as the results of the Belgium basket dataset show similar patterns to that of the LeShop, we report its results only in Section 4.2.

In keeping with our previous live intervention study of recommender systems [6], *binary hit rate* (HR Binary) is used for measuring the system performance and it is calculated as the proportion of the baskets having at least one correctly predicted item. Furthermore, we evaluate the performance following the same strategy proposed in [6], which removes duplicated items from the test baskets, re-orders those items in descending global-popularity order and then defines the last three (i.e. the least popular items) as the *target* to be predicted, given the remainder as *evidence*. This evaluation setting reflects a real scenario of marketing promotion: increasing visibility of less popular (or new) products. As shown in our previous study, among other alternatives (e.g. temporal and leave-one-out ordering), the popularity-based approach is the only evaluation strategy that generates a consistent ranking of several recommendation algorithms in terms of their live performance in a check-out recommender system using the marginal distribution of item purchase for the prior of a NB recommender algorithm. The consistency sought is to correctly rank the expected take-up of recommendations between four groups of consumers in a 2 × 2 contingency table comprising active vs. non-active and case vs. control baskets. The former characterizes consumers according to their frequency of shopping episodes during the preceding 6 month block period, and the latter separates the consumers receiving personalised recommendations from those receiving a generic recommendation using the global prior only. More details about the data and the popularity-based evaluation strategy can be found on [6].

4.1 Performance Comparison Using Popularity Based Protocol

4.1.1 Performance with Increasing Number of Active Categories

Due to the feasibility concern, it is common practise for many

---

[2]A well-known characteristic of shopping habits is seasonal behavior. Its effect is partly avoided in this study by aggre-gating shopping baskets for each customer over the training period

real-life recommender systems to model the correlations among a subset of all the product categories, also called *active categories*. In this section, we compare the performance difference among sev-eral algorithms. A performance curve is plotted, for each methods, with increasing *coverage* = (0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0) of all the $n\%$ categories (i.e. the categories that account for most of the total spend over the training data). Note that, a low (or high) coverage of the active categories is indicative of a small (or large) value of the average basket size, which is shown in Figure 2(b).

Here, we also implement the item-based collaborative filtering algorithm (item CF) [3], for comparison. In item CF, the pairwise item similarity is calculated as:

$$sim(x_i, x_j) = \frac{\sum_{\forall q: R_{q,j} > 0} R_{\cdot q} \cdot_j}{N_D(x_i) \times N_D(x_j)^{\alpha}} \quad (5)$$

where $\alpha \in [0, 1]$, $N_D(x_i)$ is the number of shoppers that have purchased the items in the training data and $R(x_i, x_j)$ is an element in the normalized $n \times m$ user-item matrix. With item CF, we report only its best performance.

Figure 2 shows results of each method evaluated on the LeShop data. As can be seen, all the methods do best at the minimum value of the average basket size, while the performance drops with increasing category coverage (i.e. the data become sparser). It is interesting to note that, both NB-based methods perform similar when the average basket size is small. This is because the value of $P_F(F_j|C_i)$ is closer to $P_D(F_j|C_i)$ (cf. Eq(2) and Eq(4)) when the basket size is small. In an extreme example where all the baskets contain no more than two items, $P_F(F_j|C_i)$ are identical.

With increasing basket sizes (Figure 2(b)), NB(items) consis-tently outperforms NB(baskets). We attribute this to the benefit of item-based probability estimate (Eq(4)) as it assigns lower weight to the co-occurring items in larger baskets. This is similar to the con-cept adopted in Eq(5) that gives lower weight to the shoppers that have purchased more items [3]. Consequently, we see that, with increasing basket sizes, both NB(items) and item CF outperform NB(baskets), which does not discriminate between items take have co-occurred with different number of other items. Our future work will explore the reasons why the item CF approach shows inferior performance in comparison with NB(items) for larger baskets.

4.1.2 Performance with Increasing last N Items

In ongoing work we find that the performance of NB(baskets) is adversely affected by the increasing number of evidence items. It is well known that feature selection using information gain can benefit the NB classifier in text classification [4]. In practice, however, we choose to use the *lastN* items to deal with the "noisy" features (or items) due to the concern of the difference between the recom-mender system and the text classifier [7]. Given a list of basket items ordered by their global prior (i.e. popularity), we use only the last few to predict the purchase probability of the new items. Intuitively, these less popular (or frequent) items are more useful in capturing item similarity than those universally liked ones [1].

Figure 3 shows the performance curves of all the methods with increasing number of the last n items, $n = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, all)$. Here, NB(baskets) shows its best performance at a value of five (*lastN* =5). Both the NB(items) and item CF models are relatively noise-resistant and do best when all the items are used for prediction. This is a practical advantage over NB(baskets) as it saves the time required to regulate the model for the optimal value of *lastN* over different time periods.
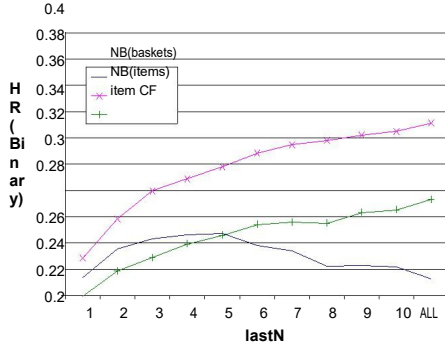
Figure 3: Performance comparison with increasing *lastN* items. *coverage* = 80% and *testset* = LeShop.

4.2 Performance Comparison Using Leave-One-Out Protocol

In keeping with our previous work regarding the performance evaluation of a recommender system, we also report the results obtained by using all-but-one (or leave-one-out) protocol [1] and compare them with another baseline method that simply recommends the most popular items not present in the basket (referred as *popular* in Figure4). Note that, with the Belgium data, we only test it with the NB-based methods as the user information required by the item CF is not available.

As shown in Figure4(b), the popular recommendation has the best performance with the leave-one-out evaluation ($p < 0.05$, McNemar's test). This is because the leave-one-out approach favors the performance on common classes as it weights all the items equally.

The popularity-based evaluation, on the contrary, favors the performance over rare classes (i.e. less popular items) as it assigns zero weight to the "hits" over the popular items in the test bas-kets. It is worth noting that, our previous live intervention study in LeShop has revealed that the performance comparison over the rare classes are more indicative of the models' live performance as the customers are unlikely to accept the recommendations of those common classes (popular product items). As shown in Figure4(a), NB(items) with the popularity ordering has the best performance in all the basket datasets ($p < 0.05$, McNemar's test).

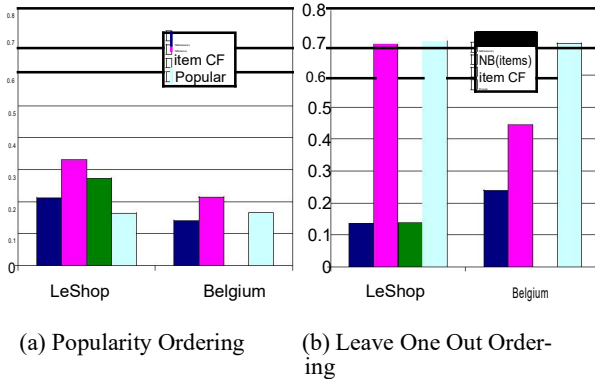(a) Popularity Ordering   (b) Leave One Out Ordering

Figure 4: Performance on the Belgium data and three months of the LeShop data. *coverage* = 80%.

5. CONCLUSIONS

A probabilistic framework using individual item purchases was proposed and its performance for personalised recommender systems was benchmarked against the standard basket-based estimates of conditional probabilities. The two alternative models of con-ditional probabilities were evaluated over two real-world basket datasets. The NB(items) model is found to be uniformly better than the NB(baskets) in terms of both binary hit rates and noise resis-tance. We attribute this to the item-based approach discriminating between items co-occurring with different numbers of other items. As the novelty (or focus) of this paper is not in the use of condi-tional probability but how it is estimated, one follow-up study is to evaluate the impact of the two estimates for other personalized recommender systems such as the association rule based system. Another future work is to generalize the current idea to estimate the high-order conditional probability, e.g. the conditional probability based on the co-occurrence of three or four items at the same time. It is intended also to deploy the NB(items) model in a live recom-mender system and so track its actual performance against other candidate methods.

Acknowledgements

6. REFERENCES

[1] J. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, 1998.

[2] T. Brijs, G. Swinnen, K. Vanhoof, and G. Wets. Using association rules for product assortment decisions: A case study. In *Knowledge Discovery and Data Mining*, pages 254–260, 1999.

[3] M. Deshpande and G. Karypis. Item-based top-n recommendation algorithms. *ACM Transactions of Information System*, 22(1):143–177, 2004.

[4] A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. In *In AAAI-98 Workshop on Learning for Text Categorization,*, 1998.

[5] V. Robles, n. P. Larra E. Menasalvas, M. S. Perez,´ and V. Herves. Improvement of naive bayes collaborative filtering using interval estimation. In *WI '03: Proceedings of the 2003 IEEE/WIC International Conference on Web Intelligence*, 2003.

[6] 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 *The 2007 International Conference on Data Mining, DMIN'07*, 2007.

[7] T. Zhang and V. S. Iyengar. Recommender systems using linear classifiers. *Journal of Machine Learning Research*, 2:313–334, 2002.