Predicting the Category of Customers' Next Product to Buy in Web Shops

Ramón Carrera Cuenca, Álvaro José Jiménez Palenzuela,
Laura Rekašiūtė, Nijolė Šalnaitė,
Flavius Frasincar

April 1, 2022

ERASMUS UNIVERSITEIT ROTTERDAM

Content

Problem statement

Problem

Task

Predict which category a customer will buy next by given purchase history in a Web shop.

Motivation

- Unlike other approaches, we are only interested in predicting the product category and not the specific product.
- Products that our considered Web shop offers are durable goods, instead of recurring purchases.
- Does not include the influence of marketing (e.g., pricing).

Issues and Challenges

- Incorporate sequential patterns.

Issues and Challenges

- Incorporate sequential patterns.
- Consider that more than one category can be purchased in the same basket.

Issues and Challenges

- Incorporate sequential patterns.
- Consider that more than one category can be purchased in the same basket.
- Account for the fact that customers vary in their buying patterns.

Existing Results

Related work

- Most of the related work that was overviewed for this problem do not take into account the sequential features of the data, i.e. [2], [3], [4];
- or used a black box model that has low explainability, i.e. [5]

SEP Model

Sequential Event Prediction model [1]

- Considers sequences of multiple events of items.
- Returns a ranked list of possible future items.
- Based on *association rules*.
- Returns very general results - predicts the most popular items all the time.

SEP Model

Sequential Event Prediction model [1]

- We suggest to apply the method per customer and obtain individual specific parameters. We present 3 possible approaches.
    - First, the individual model, which uses an individual confidence matrix.
    - Second, we introduce the general model, which instead uses a general confidence matrix.
    - We combine both models resulting in the mixed model.
- We hypothesize that a model personalized to each customers' behavior would lead to more accurate predictions.

Main Work

Data

- The Web shop data consist of transaction information on a selection of 246,932 customers from a Web shop in the Netherlands.
- This dataset contains a random and anonymized set of purchases of customers made in a three year period from 1 January 2015 to 31 December 2017.
- A total number of approximately 3.4 million orders.
- We focus only on 18 Categories.
- Due to privacy concerns the other summary statistics of the data are not made available.

Main Ideas

Notation:

- m, the number of customer histories;
- $\mathbb{Z}$, the set of categories, of size N;
- T_i, the number of purchases of customer i;
- $z_{i,t}$, t-th purchase of customer i (category or set of categories bought at time t by customer i);
- $x_{i,t}$, all purchases of customer i up to and including time t ($x_{i,t} = \{z_{i,j}\}_{j=1,\ldots,t}$);
- $X_i = x_{i,T_i}$, all purchases of customer i – his/her full history; and,
- X^m, all purchases of all m customers.

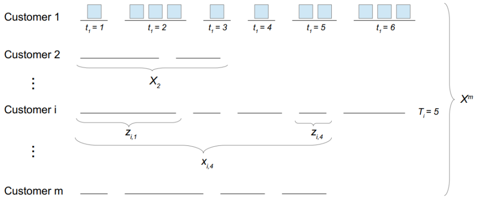


Figure 1: Notation scheme

Confidence Matrices

Association rule We define the confidence of an association rule as the proportion of the customers who bought category a and also category b in the remaining part of the sequence after category a:

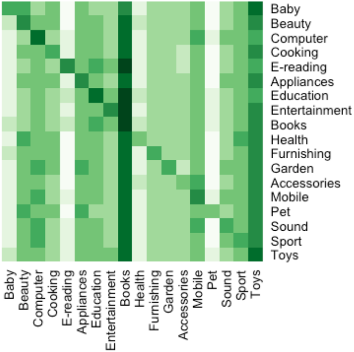
$$\underbrace{\left\{ \begin{array}{c} \text{mobile} \\ \text{phone} \end{array} \right\}}_{\text{a}} \rightarrow \underbrace{\left\{ \begin{array}{c} \text{screen} \\ \text{protector} \end{array} \right\}}_{\text{b}}$$

$$\{ \text{ printer, ink } \} \rightarrow \{ \text{ paper } \}$$

Confidence

$$Conf(a \rightarrow b) = \frac{\#(b \text{ bought after } a)}{\#a}$$

We use these confidences of association rules to construct a transition matrix.



Methodology of SEP Model

The method uses a scoring function f to score every category b. The higher the score the more likely it is to be bought.

$$f(x_{i,t}, b; \boldsymbol{\lambda}_{\varnothing}, \boldsymbol{\mu}) = \lambda_{\varnothing,b} + \sum_{j=1}^{t} \sum_{a \subseteq z_{i,j}} \mu_a \hat{P}(b|a), \tag{1}$$

where parameter $\lambda_{\varnothing,b}$ gives a score for category b when no purchase history is known and generally represents the “baseline” score for b, and the term $\mu_a \hat{P}(b|a)$ gives a score that b will be bought later in the sequence than a. μ_a is a “correction” term for category a.

Methodology of SEP Model

To get the parameter vector $\boldsymbol{\theta} = (\lambda_{\varnothing,b}, \mu_a)$ we need to optimize the loss function:

$$R_{exp}(f, X^m; \boldsymbol{\theta}) = \frac{1}{m} \sum_{i=1}^{m} \sum_{t=0}^{T_i-1} \frac{1}{T_i} \frac{1}{|K_{i,t}|} \frac{1}{|L_{i,t}|} \sum_{l \in L_{i,t}} \sum_{k \in K_{i,t}} e^{f(x_{i,t},k;\boldsymbol{\theta}) - f(x_{i,t},l;\boldsymbol{\theta})} + \beta ||\boldsymbol{\theta}||_2^2 \tag{2}$$

Here, $L_{i,t}$ is the set pf categories user i bought at time t and $K_{i,t}$ is a set of all the remaining categories, that is $K_{i,t} = \mathbb{Z}_{i,t}$

The optimisation is done using SGD algorithm.

Individually Optimised Model

To improve the personalisation of the model we propose to optimise the parameter vectors $\boldsymbol{\lambda}$ and $\boldsymbol{\zeta}$ individually for each customer. The scoring function then becomes:

$$f_I(x_{i,t}, b; \boldsymbol{\lambda}_{i,\varnothing}, \boldsymbol{\zeta}_i) = \lambda_{i,\varnothing,b} + \sum_{j=1}^{t} \sum_{a \subseteq z_{i,j}} \zeta_{i,a} \hat{P}_i(b|a). \tag{3}$$

Here $\hat{P}_i(b|a)$ is also calculated using only the sequences of the customer i. We call this the individual transition matrix.

Individually Optimised Model

Optimizing the model individually, we can also use the transition matrix that was calculated with all the available customers.

$$f_G(x_{i,t}, b; \boldsymbol{\lambda}_{i,\varnothing}, \boldsymbol{\mu}_i) = \lambda_{i,\varnothing,b} + \sum_{j=1}^{t} \sum_{a \subseteq z_{i,j}} \mu_{i,a} \hat{P}_G(b|a), \tag{4}$$

We call $\hat{P}_G(b|a)$ the general transition matrix.

Individually Optimised Model

A third way is to combine both the general and the individual transition matrices.

$$f(x_{i,t}, b; \boldsymbol{\theta}_i) = \lambda_{i,\varnothing,b} + \sum_{j=1}^{t} \sum_{a \subseteq z_{i,j}} \left(\mu_{i,a} \hat{P}_G(b|a) + \zeta_{i,a} \hat{P}_i(b|a) \right)$$

Here $\boldsymbol{\theta}_i$ is a parameter vector containing $\boldsymbol{\lambda}_i$, $\boldsymbol{\mu}_i$, and $\boldsymbol{\zeta}_i$.

Individually Optimised Model - Optimisation

For each of the approaches listed in previous slides we need to optimise a loss function for each customer.

$$R(f, X_i; \boldsymbol{\theta}_i) = \sum_{t=0}^{T_i - 1} \frac{1}{T_i} \frac{1}{|K_{i,t}|} \frac{1}{|L_{i,t}|} \sum_{l \in L_{i,t}} \sum_{k \in K_{i,t}} e^{f(x_{i,t}, k; \boldsymbol{\theta}_i) - f(x_{i,t}, l; \boldsymbol{\theta}_i)} + \beta ||\boldsymbol{\theta}_i||_2^2$$

Here the loss function is suitable for each approach. $\boldsymbol{\theta}_i$ represents the vector of all parameters needed in any of the approaches. We optimise this function using Gradient descent since its an "embarrassingly parallel" problem.

Baselines

First baseline is max confidence algorithm

It uses confidence rules $Conf(a \to b) = \frac{\#(b \text{ and } a)}{\#a}$, where a is an itemset and b is an item in the sequence. The right-hand sides of the confidence rules, i.e., the potential future items b in the sequence, are ranked and a list is constructed with these ranked items by descending confidence. This ranked list is used to make predictions and its output gives the recommendations of particular items to the user.

Baselines

Second baseline is item-based collaborative filtering

This method computes the similarities between items based on the ratings that people give to these items. The cosine similarity is intended for settings in which a user i applies a rating $R_{i,b}$ to item b. In our application, the rating reduces to $R_{i,b} = 1$ if sequence i contains item b and 0 otherwise. For each item b the binary vector of ratings $R_b = [R_{1,b}, ..., R_{m,b}]$ is constructed and then the cosine similarity between every pair of items a and b can be expressed as

$$sim(a, b) = \frac{R_a R_b}{||R_a||^2 ||R_b||^2} \tag{5}$$

Results

Evaluation

The accuracy measure is Top-3 accuracy. That is we check if any of the first 3 categories in the ranked list were actually bought in the following purchase. The proportion of times in which at least one of the predicted categories was bought next is the top-3 accuracy.

In order to test the models we split each customer's purchase history into two parts: all purchases except the last one (as training data), and the last purchase (as test data) to verify whether our models perform well.

Results

Setup

Model	Confidence matrices	Model Parameters	Optimisation algorithm	Convergence criterion
Individually-optimised	20 000 customers	$\lambda_0 = 0.1, \nu = 0.2, \beta = 0.1$	Gradient descent	$\epsilon = 10^5$
Base model	5000 customers	$\lambda_0 = 0.1, \nu = 0.2, \beta = 0.1$	Stochastic gradient descent	$\epsilon = 10^5$

- A maximum of 10 previous baskets are used for optimization.
- For **base model** we ran the model 10 times on 5000 random customers because of the lengthy optimisation.
- For **individually optimised models** we ran 1000 iterations of randomly selected 5 000 customers.

Results

Model	Mean Top-3 accuracy	Number of runs
Base model	40.0%	10
Individually optimised		
Individual transition matrix	47.7%	1000
General transition matrix	50.3%	1000
Mixed transition matrix	50.7%	1000
Max-Confidence baseline	46.9%	1000
Item-based collaborative filtering baseline	43.7%	1000

Future work

We propose to use clustering $\rightarrow$ find similar customers and recommend based on them.

For example, clustering based on MOST POPULAR CATEGORY, when we have as many clusters as categories and a customer is assigned to a cluster based on the number of times he/she bought an item from a certain category:
51% in mixed model $\rightarrow$ 58% in clustered model

References

B. Letham, C. Rudin, and D. Madigan.
Sequential event prediction.
Machine Learning, 93(2-3):357–380, 2013.

G. Linden, B. Smith, and J. York.
Amazon. com recommendations: Item-to-item collaborative filtering.
IEEE Internet Computing, 7(1):76–80, 2003.

S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme.
Factorizing personalized markov chains for next-basket recommendation.
In *Proceedings of the 19th International Conference on World Wide Web (WWW 2016)*, pages 811–820. ACM, 2010.

P. Wang, J. Guo, Y. Lan, J. Xu, S. Wan, and X. Cheng.
Learning hierarchical representation model for nextbasket recommendation.
In *Proceedings of the 38th international ACM SIGIR Conference on Research and Development in Information Retrieval, (SIGIR 2015)*, pages 403–412. ACM, 2015.

F. Yu, Q. Liu, S. Wu, L. Wang, and T. Tan.
A dynamic recurrent model for next basket recommendation.
In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, (SIGIR 2016)*, pages 729–732. ACM, 2016.