Discovering Customer Journey Maps using a Mixture of Markov Models

Matthieu Harbich[1,2], Gaël Bernard[1], Pietro Berkes[2], Benoît Garbinato[1], and Periklis Andritsos[3]

[1] University of Lausanne, Faculty of Business and Economics (HEC), Switzerland,
[firstname].[lastname]@unil.ch
[2] NAGRA Kudelski, Switzerland, pietro.berkes@nagra.com
[3] University of Toronto, Faculty of Information, Canada,
periklis.andritsos@utoronto.ca

Abstract. Customer Journey Maps (CJMs) summarize the behavior of customers by displaying the most common sequences of steps they take when engaging with a company or product. In many practical applications, the challenge lies in automatically discovering these prototypical sequences from raw event logs for thousands of customers. We propose a novel, probabilistic approach based on a mixture of Markov models and show it can reliably extract CJMs with just one input parameter (and potentially none).

Keywords: markov models, customer journey mapping, event logs, customer journey analytics

1 Introduction

Managing customer experience is becoming increasingly complex. The plethora of ways in which a customer can consume a service, the acceleration of media and channel fragmentation, and the increase in customer-to-customer interactions through social media create opportunities, but also present major challenges to companies [7]. Customer Journey Maps (CJMs) respond by proposing an intelligible visualization that makes it possible to understand, discuss, or improve the main paths (i.e., journeys) in the usage of a service.

In practice, one is usually faced with a log of raw events generated from thousands of users interacting with a system. Each of these events symbolizes a step a customer takes when using a service, while a single customer's journey might be composed by several steps (i.e., events). This yields the following challenge: how to build a CJM that summarizes well the journeys observed in the event logs? Figure 1 illustrates this challenge by using an event log composed of five journeys ❶. As shown in Fig. 1, displaying all individual journeys on a single CJM leads to an unreadable model ❷. One option, proposed by Bernard and Andritsos [2], is to summarize individual journeys with a few representative journeys, and assign each journey to its closest representative ❸. To this end, they

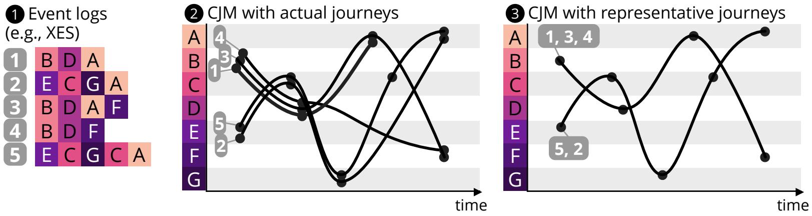

Fig. 1. An event log containing five journeys, where each letter represents an activity type ❶, a CJM with five actual journeys ❷, and its equivalent using two representative journeys ❸.

take advantage of hierarchical clustering to organize the journeys and, then, they apply a frequent sequence mining algorithm to define representative journeys.

We contribute by proposing the first probabilistic approach that converts event logs in a CJM. We first describe our approach and, then, we illustrate its most appealing features – missing in [2] – using the Chicago activity survey dataset.

2 Approach

The starting point for our approach is to assume that the customers belong to a small number of groups, each having a similar behavior in the sense of having the same probability of transitioning from one event to the next.

We describe our approach in more details below: 1) the input, 2) the Markov model, and 3) the use of an Expectation-Maximization (EM) algorithm on a mixture of Markov models.

Input. Event logs are the input data for our algorithm. For example, logs can be provided in the XES format, a standard representation used in Process Mining to model processes [1], which can be mapped to store customer journeys [3]. The only other input is the number of representative journeys one would like to discover, k.

Markov Model as an internal representation. Our model for event sequences is a Markov model: according to the model, each sequence is generated by choosing the first event in the sequence, X_1, with probability $P(X_1 = i) = f_i$. The probability of every successive event in the sequence depends only on the event preceding it, i.e. $P(X_t = i|X_{t-1}, X_{t-2}, ..., X_1) = P(X_t = i|X_{t-1} = j) = T_{ij}$. By choosing the parameters $\mathbf{f}$ and $\mathbf{T}$ to match the probability of the first event and the probability of event transitions for a group of observed sequences, the model captures our assumption that journeys belonging to the same group share the same probability of transitioning between events.

Expectation-Maximization on a mixture of Markov models. Computing one Markov model for the entire dataset would only describe one generic type of behavior. Our assumption is that we have a set of k groups of customers, each with its own model, i.e., we have a mixture of k Markov models. Given a set of journeys, and the set of k pairs of parameters ($\mathbf{T_k}$, $\mathbf{f_k}$) for each group, the probability of the data given the parameters (known as the *likelihood* of the parameters) can be written as

$$LH(\mathbf{T_k}, \mathbf{f_k}, \pi_k) = \prod_{n=1}^{N} \sum_{k=1}^{K} P(X^n | C_k; \mathbf{T_k}, \mathbf{f_k}) P(C_k; \pi_k) \qquad (1)$$

where

N : The set of sequences
K : The number of Markov models
P$(X^n|C_k)$: The probability of sequence X^n under model C_k
P(C_k) : The probability of model k (one group of behavior might be more likely than another

The parameters $\mathbf{T_k}$, $\mathbf{f_k}$, and π_k are learned by Maximum Likelihood, i.e., by maximizing the probability of the data under the mixture, $LH(\mathbf{T_k}, \mathbf{f_k}, \pi_k)$. An efficient way of achieving that is using the Expectation-Maximization algorithm [5,6]. The expectation part evaluates the models' responsibilities for the behavior using the current parameters. In the maximization part, the algorithm re-computes the parameters using the responsibilities of the different models computed in the expectation part. The algorithm loops until the likelihood converges [4].

The K discovered models, each composed by a transition matrix and a first probability vector, represent the K different types of behaviors the dataset contains. Our approach offers interesting features that will be illustrated in the next section.

3 Case study

The probabilistically-grounded nature of the algorithm offers many advantages compared to the approach described in [2], which is based on hierarchical clustering of edit distances of the underlying journeys. We illustrate the advantages of our approach by applying our algorithm to a real, publicly available dataset. We first describe the dataset and then list four interesting features of this probabilistic approach.

The dataset consists of an activity survey for the city of Chicago [1]. It comprises 29,542 journeys (2,381 are unique). Typically, one journey might be: "being at home" → "going to school" → "having a meal" → "being at home". Such a journey has a length of four events. The average length for the dataset is 4.817 events and there are 16 types of event. Fig. 2 shows the resulting CJM with k=3.

[1] http://www.cmap.illinois.gov/data/transportation/travel-survey

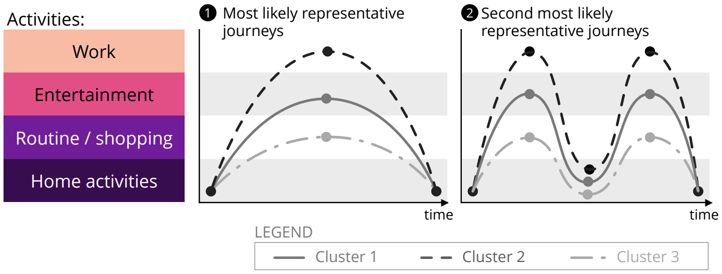

Fig. 2. Results when applied on the Chicago activity survey dataset with k=3. ❶ shows the most likely representative journeys seen in the event logs for each cluster and ❷ shows the second most likely representative journeys.

Feature 1. The first interesting feature resides in the ability to determine the likelihood of a given journey under each model (see Fig. 2). Each journey is assigned to a model k with a given probability, $P(X|C_k)$. The journey with the highest probability becomes the representative. It should be noted that this also means that we can return other alternatives (e.g., the second most likely journeys as seen on Fig. 2), which can be useful for investigating how a journey might deviate from the most likely representative journey.

Feature 2. The next interesting feature is the limited number of parameters. The work proposed in [2] includes numerous parameters that must be chosen from (e.g., the distance between journeys, and the method for finding the representative or computing the distance between clusters), which can have a substantial impact on the representative journeys that are returned. In contrast, our algorithm requires us to provide only the number of groups, k. Even this parameter can be removed by using standard model selection techniques (e.g., using a Bayesian Information Criterion (BIC) penalty), but such a technique is outside the scope of this short paper.

Feature 3. The next most probable event of an incomplete journey can be computed on the fly by checking the most probable transition parameter under its most likely model (see Fig. 3). Moreover, one can also use the transition matrix to generate new journeys that have never been seen but that are still likely to happen according to the mixture of models. Generating such data journeys can be useful for training or analysis purposes.

Feature 4. One of the interesting features of our probabilistic-approach is the way it handles shifting customers' behavior. For example, an incomplete journey is assigned to a Markov model with a given probability. The most probable Markov Model might change when new events are observed. Hence, it allows us to dynamically cope with shifting behaviors. A customer's behavior is complex [7] and, therefore, we argue that assigning the customer to a continuum between,

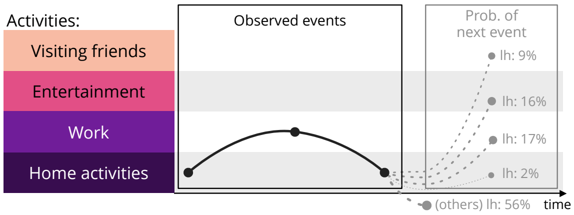

Fig. 3. Inferring the most likely next event based on a set of observed sequences.

potentially several different models is more appropriate than using a discrete classification.

Our algorithm scales well as it has been applied – as part of the collaboration with Kudelski – on a real dataset consisting of more than 7 million events for more than half a million customers. It results in the discovery of several customer behavioral patterns that enable marketers to design more personalized campaigns.

Overall, a probabilistic approach offers a robust alternative based on a small number of assumptions and with a small number of parameters. The four features we presented in this paper might serve as a starting point for studying alternative probabilistic approaches to discovering sets of journeys to be displayed on CJMs using standardized event logs.

4 Acknowledgement

This research was conducted at NAGRA Kudelski (Lausanne, Switzerland). We would like to thank the Kudelski Group for the fruitful collaboration and support.

References

1. van der Aalst, W.: Process Mining: Data Science in Action. Springer (2016)
2. Bernard, G., Andritsos, P.: Cjm-ex: Goal-oriented exploration of customer journey maps using event logs and data analytics. In: 15th International Conference on Business Process Management (BPM2017) (2017)
3. Bernard, G., Andritsos, P.: A process mining based model for customer journey mapping. In: Proceedings of the Forum and Doctoral Consortium Papers Presented at the 29th International Conference on Advanced Information Systems Engineering (CAiSE 2017) (2017)
4. Bishop, C.M.: Pattern recognition and machine learning. Information science and statistics, Springer, New York (2006)
5. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society. Series B (methodological) pp. 1–38 (1977)
6. Kiran, P.M., Rao, A.P., Ratnamala, B.: An efficient approach for filling incomplete data `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.735.3758&rep=rep1&type=pdf`
7. Lemon, K.N., Verhoef, P.C.: Understanding customer experience throughout the customer journey. Journal of Marketing (2016)