Contextual and Behavioral Customer Journey Discovery Using a Genetic Approach

Gaël Bernard[1] and Periklis Andritsos[2]

[1] University of Lausanne, Faculty of Business and Economics (HEC), Switzerland
gael.bernard@unil.ch
[2] University of Toronto, Faculty of Information, Toronto, Canada,
periklis.andritsos@utoronto.ca

Abstract. With the advent of new technologies and the increase in customers' expectations, services are becoming more complex. This complexity calls for new methods to understand, analyze, and improve service delivery. Summarizing customers' experience using representative journeys that are displayed on a Customer Journey Map (CJM) is one of these techniques. We propose a genetic algorithm that automatically builds a CJM from raw customer experience recorded in a database. Mining representative journeys can be seen a clustering task where both the sequence of activities and some contextual data (e.g., demographics) are considered when measuring the similarity between journeys. We show that our genetic approach outperforms traditional ways of handling this clustering task. Moreover, we apply our algorithm on a real dataset to highlight the benefit of using a genetic approach.

Keywords: customer journey mapping, process mining, customer journey analytics, genetic algorithms

1 Introduction

A customer experience can be defined as a customer's journey with an organization over time across multiple interactions called touchpoints [1]. A journey can be as simple as a single activity but can also involve complex interactions. Recent studies show that customer interactions are increasing [2], services are becoming more complex, and customers are often unpredictable [3]. In this context, understanding the main trajectories that were followed by customers to consume a service is a complex task. According to Verhoef et al., a strategy based on customer experience may provide a superior competitive advantage [1]. It is, therefore, not surprising that "Characterizing the Customer Journey along the Purchase Funnel and Strategies to Influence the Journey" has been ranked as one of the most important research priorities for the coming years by the Marketing Science Institute [4]. Concretely, a challenge faced by many practitioners is to make sense of thepotentially infinitecombination of activities that exist in order to consume a service. As a response, new methods to understand, design, and analyze customer journeys are emerging from the industry and are

becoming increasingly popular among researchers. One of these methods, which will be the focus of this paper, is the Customer Journey Map (CJM). A CJM is a conceptual tool used to better understand customers' trajectories when they are consuming a service [5]. It depicts typical journeys that will be experienced by the customers across several touchpoints.

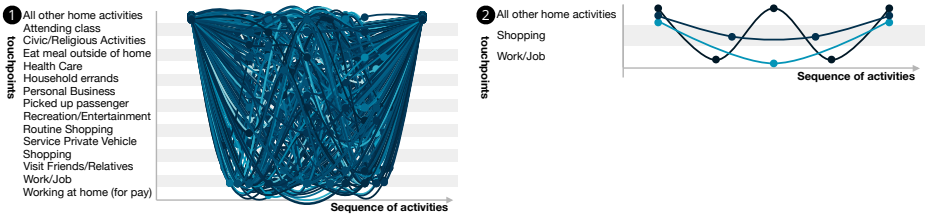

Fig. 1. ❶ A fraction of the dataset –612 sequences out of 123'706– displayed on a CJM, and ❷, a CJM that summarizes the entire dataset using three representatives.

Fig. 1 shows CJMs derived from a real dataset[3]. In this dataset, a journey is all the activities that are performed by a citizen throughout the day. For instance being at home, attending class and going back home is one of the potential journeys. As can be seen in ❶ of Fig. 1, displaying such actual journeys on the CJM without preprocessing the data results in an overwhelming chart. It becomes clear that when a company deals with very large numbers of actual journeys, it is necessary to reduce the complexity and to look at these journeys at a higher level of abstraction. Specifically, representative journeys address this issue, [6], by summarizing the dataset (using three journeys visible in ❷ of Fig. 1).

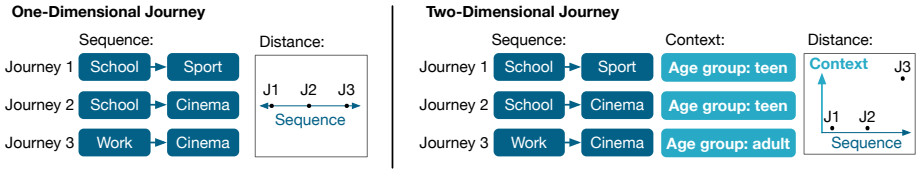

Fig. 2. Measuring the distance among three journeys with and without the context.

The existing solutions to summarize collections of journeys [7, 6] consider only the sequence of touchpoints when measuring the distance between journeys. Fig. 2 illustrates the process with 3 short journeys. Using a basic distance measure between sequences (e.g., edit-distance), we cannot say which one of 'Journey 1' or 'Journey 3' is closer to 'Journey 2' (left-part of Fig. 2). We suggest that

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

demographics and other contextual information might be equally important to measure the distance between journeys. Hence, in this paper, we propose to integrate such information when mining journeys. The right-part of Fig. 2 shows that when we also consider the age group, it becomes clearer that the closest journey to 'Journey 2' is 'Journey 1'.

We propose an algorithm summarizing a customer experience using both the sequence of activities as well as the contextual information. Our genetic approach uses only three intuitive parameters: 1) the approximate number of representative journey to use, 2) the weight of the sequence of activities, and 3) the weight of the contextual data. Using synthetic datasets and cluster analysis techniques, we demonstrate that our approach outperforms existing techniques. Finally, we highlight the impact of the three parameters using a real dataset and illustrate the results with CJMs.

The paper is organized as follows. Section 2 clarifies the customer journey discovery activity. In Section 3, we propose an overview of existing techniques. Section 4 depicts our genetic algorithm. In Section 5, we evaluate the results using internal and external evaluation metrics. Section 6 illustrates CJMs produced by our algorithm. Finally, Section 7 concludes the paper.

2 Customer Journey Discovery

The goal of a customer journey discovery algorithm is to find a reasonable amount of representative journeys that well actual journeys.

Definition 1 (Touchpoint): a touchpoint is an interaction between a customer and a company's products or services [5]. 'Sharing on social network' or 'ordering a product on the website' are two typical examples of touchpoints in a retail context. We define t as the touchpoint, and T the finite set of all touchpoints. The touchpoints are visible in the y-axis of the CJMs (Fig. 1).

Definition 2 (Actual Journey): An actual journey J_a is a sequence of touchpoints observed from customers. For instance, $J_a = \{\langle$'Visiting the shop', 'Testing the product', 'Sharing on social network'$\rangle\}$ is a journey with three touchpoints. For the sake of brevity, we replace the touchpoints with alphabetical characters so that J becomes $\langle$ABC$\rangle$. The order in which the activities are executed is represented by the x-axis of the CJMs visible in Fig. 1.

Definition 3 (Representative Journey): A representative journey, J_r, is a journey that summarizes a subset of actual journeys. The left-part of Fig. 1 shows how a CJM would look like when we display actual journeys, while the right-part uses representative journeys. Clearly, as can be seen in Fig. 1, the use of representative journeys increases the readability of the CJM.

Definition 4 (Event Logs): Let $J_{\mathcal{A}}$ be an event log, which we define as the collection of all actual journeys observed by customers.

Definition 5 (Customer Journey Map): A customer journey map summarizes customer journeys through the use of representative journeys. Let a customer journey map $J_{\mathcal{R}}$ be the set of all the J_rs summarizing $J_{\mathcal{A}}$. Let $k_{\mathcal{R}}$ denote

the number of journeys in a map (i.e., $|J_{\mathcal{R}}|$). Typically, the part ❷ of Fig. 1 is a CJM, $J_{\mathcal{R}}$, containing three representative journeys summarizing an event logs.

The customer journey discovery activity can be defined as a function that maps all members of $J_{\mathcal{A}}$ to a member of $J_{\mathcal{R}}$; i.e., that maps all the actual journeys to representative journeys ultimately displayed on a CJM. Discovering customer journeys from event logs is an unsupervised clustering task that entails interesting challenges. First, there is no evidence in the literature that deals with the ideal number of representative journeys to use. When the goal is to have a general overview about dataset, it seems reasonable to display only few journeys so the CJM is readable. However, discovering few dozens representative journeys might also be a relevant choice if the goal is to catch complex and less generic patterns. Finally, the sequence that best summarizes its assigned actual journeys needs to be found. It might be the case that an ideal representative journey was never observed but well summarizes the actual journeys. This phenomena was observed by Gabadinho et al., and illustrated as follows: "We could imagine synthetic – not observed – typical sequences, in the same way as the mean of a series of numbers that is generally not an observable individual value." [8]. Before presenting our solution, the next section describes related work.

3 Related Work

Our work is close to the summarization of categorical sequences used in social sciences. In [7, 8], Gabadinho et al. propose summarizing a list of observed sequences (i.e., actual journeys) using a representative (i.e., a representative journey). They define a representative set as *"a set of non-redundant 'typical' sequences that largely, though not necessarily exhaustively, cover the spectrum of observed sequences"* [7]. We argue that it matches our definition of a representative sequence summarizing a set of sequences. The authors propose multiple ways to select a representative. The *'frequency'* (1), where the most frequent event is used as the representative. The *'neighborhood density'* (2), which consider the representative as the sequence with the most neighborhood in a defined diameter. The *'centrality'* (3): the representative – or medoid – can be found using the centrality. Being the most central object, the representative is the sequence with the minimal sum of distances to all other sequences. Finally, the *'sequence likelihood'* (4): the sequence likelihood of a sequence derived from the first-order Markov model can also be used to find the representative.

Because it starts from data in order to produce conceptual models, process mining is another discipline closely related to the topic of customer journey discovery. The link between customer journey maps and process mining was highlighted in [5]. However, business process models and CJMs are not built for the same purpose. While a business process model captures how a process was or should be orchestrated, a CJM is built for the purpose of better understanding what customers have experienced. Gartner highlighted that CJMs are used to supplement but not to replace BPMs [9].

In [10], we propose CJM-ex, a web-based tool to navigate CJMs. It uses a hierarchical structure that allows drilling-down in customer journeys. In [11], it was shown that customer journey maps can be discovered using Markov model. In [6], we suggested a genetic approach to discover representative journeys that uses only the sequence of touchpoints to measure the distance between journeys. Hence, this current work can be seen as an extension of [6] to allow taking both the sequence of touchpoints and the contextual information into account when build CJMs.

4 Genetic Algorithm for CJM Discovery

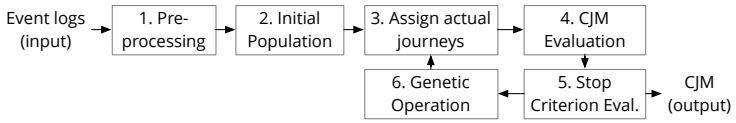

Fig. 3. Overview of the genetic algorithm to discover CJMs

As an introduction, Fig. 3 provides intuition on how the genetic algorithm builds a CJM. A set of actual journeys, J_a, is provided to the algorithm. Then, during *Generation 1*, we build p number of $J_{\mathcal{R}}$, where p *is the population size*; i.e., the number of CJMs that will be evaluated after each generation. In Fig. 4, p is 3. Each $J_{\mathcal{R}}$ is evaluated and we keep the e best $J_{\mathcal{R}}$s, called the *elite set* while we discard the other $(p-e)$ $J_{\mathcal{R}}$s. Then, we move to *Generation 2*, where we keep an untouched version of the e number of $J_{\mathcal{R}}$s in the elite set. In Fig. 4, e is set to 1. For instance, $J_{\mathcal{R}2}$ was kept in Generation 2 because it has the best average quality in Generation 1, i.e., $J_{\mathcal{R}2}$ is intact in Generation 2. We then apply some transformation (to be discussed next) to generate $(p-e)$ new $J_{\mathcal{R}}$s that will, in turn, be evaluated. In our case, we generate $J_{\mathcal{R}4}$ and $J_{\mathcal{R}5}$. We recursively transform and evaluate the p number of $J_{\mathcal{R}}$s until a stopping criterion is met. Once a stopping criterion is met, we return the best $J_{\mathcal{R}}$ ($J_{\mathcal{R}7}$ in Fig. 4). *The best $J_{\mathcal{R}}$ can be interpreted as the best set of representative journeys, J_r, representing a set of journeys from $J_{\mathcal{A}}$ that have been found given a certain evaluation criterion.* The next section describes how we generate the initial population, what various types of operations we apply on each $J_{\mathcal{R}}$ to transform them, and how we evaluate each one of them given a set of journeys in $J_{\mathcal{A}}$.

4.1 Preprocessing

To gain in efficiency, we make the assumption that $J_{\mathcal{R}}$ will be close to the frequent patterns observed in J_a. Let Top_{ℓ_n} be the n most occurring patterns of length ℓ and $Top_n \supseteq Top_{\ell_{[2,m]}}$ be the superset of all the most occurring

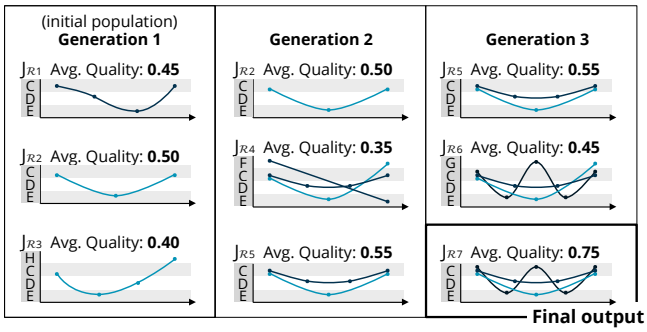

Fig. 4. Illustration of the genetic process to discover the best CJM

patterns of lengths 2 to m. Top_n is used later to form the initial population of $J_{\mathcal{R}}$ (described in Section 4.2), and to add a random journey to $J_{\mathcal{R}}$. Using Top_n we avoid generating journeys by picking a random number of touchpoints from T. According to our experiments, using Top_n reduces the execution time by two to get an output $J_{\mathcal{R}}$ of the same average quality.

4.2 Initial Population

The initial population is generated by adding a sequence randomly picked from Top_n (defined in Sect. 4.1). In our running example, depicted in Fig. 4, the initial population is visible in column 'generation 1'. In Fig. 4, the population size is 3. In our experiments, we set the population size to 100.

4.3 Assign Actual Journeys

The quality of a representative journey can only be measured when knowing which actual journeys it represents. Hence, a first step toward evaluating the quality of $J_{\mathcal{R}}$ is to assign each journey $J_a \in J_{\mathcal{A}}$ to its closest journey in $J_r \in J_{\mathcal{R}}$. To characterize the closeness between J_a and J_r, we use the Levenshtein distance borrowed from [12]. It is a metric particularly well-suited to measure the distance between sequences. The Levenshtein distance counts the number of edit operations that are necessary to transform one sequence into another one. There are three types of operations: deletions, insertions, and substitutions. For instance, the distance between $\langle$ABC$\rangle$ and $\langle$ACCE$\rangle$ is 2 since one substitution and one insertion are required to match them. We define the closest representative as the one having the smallest Levenshtein distance with the actual journeys. Note that if a tie occurs between multiple best representatives, we assign the J_a to the J_r having the smallest amount of actual journeys already assigned to it. Once each actual journey has been assigned to its closest representative, we can evaluate $J_{\mathcal{R}}$ using the criteria described in the next section.

4.4 CJM Evaluation Criteria

This section introduces the evaluation criteria used to determine the quality of each $J_{\mathcal{R}}$, namely, (1) the fitness, (2) the number of representatives, (3) the contextual distance, and (4) the average quality.

Fitness. The fitness measures the distance between each sequence of activities J_a and its closest representative $J_{\mathcal{R}}$ using the Levenshtein distance [12].

$$Fitness(J_a, J_{\mathcal{R}}) = 1 - \frac{\sum_{i=1}^{|J_a|} min_{j=1}^{|J_{\mathcal{R}}|}(Levenshtein(\sigma_{\mathcal{A}_\mathbf{i}}; \sigma_{\mathcal{R}_\mathbf{j}}))}{\sum_{i=1}^{|J_a|} Length(\sigma_{\mathcal{A}_i})} \tag{1}$$

where

$\sigma_{\mathcal{A}_\mathbf{i}}$: i$^{\text{th}}$ actual sequence observed in event logs
$\sigma_{\mathcal{R}_\mathbf{j}}$: j$^{\text{th}}$ representative contained in $J_{\mathcal{R}}$
$Length(x)$: Length of the sequence of activity x

A fitness of 1 means that the representative journey perfectly catches the behavior of the actual journeys assigned to it. In contrast, a fitness close to 0 implies that many edit operations are necessary to match the sequences.

Number of Representatives. An event log can contain several hundreds or thousands of unique actual journeys. Hence, we maximize the fitness without trying to keep a low $k_{\mathcal{R}}$, the CJM will become unreadable because too many representative journeys will be displayed in it. In other words, $J_{\mathcal{R}}$ overfits. Hence, the goal is to find a $k_{\mathcal{R}}$ that offers a good compromise between underfitting and overfitting. Finding the optimal number of clusters is a recurrent challenge when clustering data. We propose integrating traditional ways of determining the optimal number of clusters, such as the Bayesian information criterion [13], or the Calinski-Harabasz index [14]. The idea is to evaluate a range of solutions (e.g., from 2 to 10 journeys) and to keep the best solution. Let k_h be the optimal number of clusters returned by one of the techniques mentioned above. By integrating k_h into the evaluation, we can guide the solution toward a $k_{\mathcal{R}}$ that is statistically relevant. To evaluate the quality, we measure the distance between $k_{\mathcal{R}}$ and k_h. To do this, we propose the following distribution function:

$$NumberOfRepresentatives(k_{\mathcal{R}}, k_h, x_0) = \frac{1}{1 + (\frac{|k_{\mathcal{R}} - k_h|}{x_0})^2} \tag{2}$$

where

$k_{\mathcal{R}}$: Number of J_r journeys on $J_{\mathcal{R}}$ (i.e., $|J_{\mathcal{R}}|$)
k_h : Optimal number of journeys (e.g., using the Calinski-Harabasz index)
x_0 : x value of the midpoint

The parameter x_0 determines where the midpoint of the curve is. Concretely, if $x_0 = 5$, $k_{\mathcal{R}} = 11$ will result in a quality of 0.5 because the absolute distance from k_h is 5. We set $x_0 = 5$ for all our experiments. Because the number of representatives is not the only criteria to assess the quality of a CJM, the final CJM might contain more or less journeys if it increases the average quality.

Contextual Distance. The contextual distance allows us to consider the set of contextual data C when grouping similar journeys. The more distant the set of contextual data is between J_a that are represented by distinct J_r, the better the quality is. To measure the distance, we first build a value frequency table which count all the values per representatives (v_i is the value frequency counter for J_{r_i}). Then, for each pair of clusters, we calculate the cosine similarity, which is defined as:

$$ContextualDistance(v_1, v_2) = \frac{v_1 \cdot v_2}{||v_1|| \cdot ||v_2||} \tag{3}$$

Finally, the cosine distances are averaged to get the overall contextual distance. A short overall distance indicates that the contextual data of J_a that are assigned to distinct J_r are similar. In other words, the contextual data does not help in classifying J_a between several J_r.

Average Quality. We assign weights to the fitness, the number of representatives, and the contextual distances qualities to adjust their relative importance. Then, we get the overall quality by averaging the weighted qualities using the arithmetic mean.

4.5 Stopping Criterion

Once we assess the quality of generated CJMs, we check if a stopping criterion is met. There are three ways we can use to stop the algorithm taken from [15, 16]. (1) The algorithm could stop after a certain number of generations. (2) One could stop the algorithm when a certain number of generations have been created without improving the average quality. (3) We could stop the algorithm when a certain quality threshold is reached for one of the evaluation criteria. Because it is difficult to predict the quality level that can be reached, we believe that stopping the algorithm using a threshold is not advisable. For this reason, we used a combination of approaches 1 and 2 for our experiments. Once the stopping criteria have been evaluated, there are two outcomes. Either one of the criteria is met and the algorithm stops, returning the best $J_{\mathcal{R}}$, or, we generate new candidates by recursively calling a function that generates the next population, described in the next section.

4.6 Genetic Operations

Once all the CJMs have been evaluated, we rank them by their average quality and copy a fraction (i.e., e) of the best ones in *elite*. In Fig. 4, the elite size is 1. In our experiments, we set the elite size to 5. Because we keep an untouched version of the e number of $J_{\mathcal{R}}$s, we make sure that the overall quality will only increase or stay steady. Then, we generate $(p - e)$ new $J_{\mathcal{R}}$s as follows. *(1) Add a random journey (mutation):* A sequence is randomly picked from Top_n and added to $J_{\mathcal{R}}$. *(2) Add an existing journey (crossover):* An existing representative

journey is randomly picked from the elite population and added to $J_{\mathcal{R}}$. *(3) Delete a journey (mutation):* A random journey is removed from $J_{\mathcal{R}}$. Nothing happens if $J_{\mathcal{R}}$ contains only one journey. *(4) Add a touchpoint (mutation):* A touchpoint from T is added to one of the journeys from $J_{\mathcal{R}}$ at a random position. *(5) Delete a touchpoint (mutation):* A touchpoint is removed from $J_{\mathcal{R}}$ unless removing this touchpoint would result in an empty set of touchpoints.

In Fig. 4, $J_{\mathcal{R}5}$ have been produced by taking $J_{\mathcal{R}2}$ and adding a journey picked from Top_n (defined in Sect. 4.1). As described in Fig. 3, once new $J_{\mathcal{R}}$s have been created, we go back to the evaluation phase where the new $J_{\mathcal{R}}$s are evaluated until one stopping criterion is met. Once such a criterion is met, we return the best $J_{\mathcal{R}}$s of the last generation (e.g., $J_{\mathcal{R}7}$ in Fig. 4) and the algorithm stops.

We loop over each of these 5 types of transformations three times. Each time, the probability of applying the transformation is 10%, which means that more than one transformation is applied. It also means that the same transformation might be applied up to three times (with a probability of 0.1%). At the very least, one transformation has to be applied. If it is not the case, we loop over each transformation three times again until at least a transformation is applied.

5 Evaluation Using Synthetic Datasets

In order to evaluate the quality of our approach to return the best set of representative journeys in $J_{\mathcal{R}}$, we evaluate the results using a collection of synthetic customer journeys that includes some contextual data. We first describe how we generated the dataset. Then, using this synthetic dataset, we evaluate and compare our algorithm with state-of-the-art techniques to summarize sequences of categorical data.

5.1 Datasets

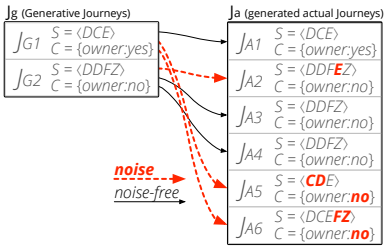

Fig. 5. Dataset with 50% of noise.

To evaluate the results of our algorithm, we produced several event logs that simulate journeys. Generating the event logs ourselves means that we know the ground truth represented by the generative journeys and therefore our task would be to recover these journeys from a set of actual ones we produce. A generative journey is a known sequence of activities with a known set of characteristics from which we generate the event logs. If we were to use only these generative journeys to generate 1,000 thousand journeys, we would obtain only $k_{\mathcal{G}}$ distinct journeys. From a business point of view, this would describe an ideal situation where each group of customers behaves in an homogeneous way. However, we know that this is not the case. A more realistic situation would depict a scenario where each group of

customers can be described by a representative sequence of activities, but the actual journeys within the group can deviate from the representative one. To produce more realistic data, we inject noise for a fraction of the journeys. For instance, if the noise level is set to 50%, $J_a = J_g$ is true for half of the data. Fig. 5 illustrates how six journeys are generated from two generative journeys. If we assume that the noise level is defined to be 50%, three actual journeys in the events logs deviate from the original generative journeys. The goal of our experiments is to retrieve the set of generative journeys, as representatives, from the produced actual journeys. The 40 generated datasets as well as details on how we produced them are made publicly available[4].

5.2 Metrics

To evaluate and compare the quality of representative journeys, we use both external and internal evaluation metrics. On the one hand, the external ones evaluate the results relative to the ground truth, i.e., from the generative journeys. On the other hand, the internal evaluation uses cluster analysis techniques to assess the results. The aim is to account for the fact that the ground truth might not be the optimal solution. Indeed, adding random noise might change the optimal solution. This section introduces these metrics. For the internal evaluation metrics, we borrowed metrics from [8].

External Evaluation - Distance in Number of Journeys. This metric measures the distance between the number of generative journeys and the number of representative journeys returned by the algorithm. We propose the following metric:

$$NbJourneysDistance(k_{\mathcal{G}}, k_{\mathcal{R}}) = abs(k_{\mathcal{G}} - k_{\mathcal{R}}) \tag{4}$$

External Evaluation - Jaccard Distance. To evaluate the distance between the sequences of activities from the generative journeys ($\sigma_{\mathcal{G}}$) and the discovered representative journeys ($\sigma_{\mathcal{R}}$), we propose to use the Jaccard Distance where a score of 1 indicates a perfect match between the set of sequences from the generative journeys and the representative ones.

$$JaccardDistance(\sigma_{\mathcal{R}}, \sigma_{\mathcal{G}}) = 1 - \frac{|\sigma_{\mathcal{R}} \cap \sigma_{\mathcal{G}}|}{|\sigma_{\mathcal{R}} \cup \sigma_{\mathcal{G}}|} \tag{5}$$

Internal Evaluation - Mean distance [8]. The mean distance i returns the average distance between the representative sequence i and the sequence of actual journeys that have been assigned to i. If the mean distance i is 0, then the representative journey i perfectly matches the underlying actual journeys.

$$MeanDistanceScore_i = \frac{\sum_{j=1}^{k_i} D(S_i, S_{ij})}{k_i} \tag{6}$$

[4] http://customer-journey.unil.ch/datasets/

where

$D(x_1, x_2)$: Levenshtein distance between two sequences
k_i : Number of actual journeys assigned to the representative journey i
S_i : Representative sequence i
S_{ij} : Sequence of actual journeys j assigned to i

Internal Evaluation - Coverage [8]. The coverage indicates the proportion of actual journeys that are within the neighborhood n of a representative.

$$Coverage_i = \frac{\sum_{j=1}^{k_i} (D(S_i, S_{ij}) < n)}{k_i} \tag{7}$$

where

$D(x_1, x_2)$: Levenshtein distance between two sequences
k_i : Number of actual journeys assigned to the representative journey i
S_i : Representative sequence i
S_{ij} : Sequence of actual journeys j assigned to i

Internal Evaluation - Distance gain [8]. The distance gain measures the gain in using a representative journey instead of the true center of the set (i.e., the medoid of the whole dataset). In other words, it measures the gain obtained in using multiple representative journeys instead of a single one.

$$DistGain_i = \frac{\sum_{j=1}^{k_i} D(C(\sigma_{\mathcal{A}}), S_{ij}) - \sum_{j=1}^{k_i} D(S_i, S_{ij})}{\sum_{j=1}^{k_i} D(C(\sigma_{\mathcal{A}}), S_{ij})} \tag{8}$$

where

$D(x_1, x_2)$: Levenshtein distance between two sequences
k_i : Number of actual journeys assigned to the representative journey i
S_i : Representative sequence i
S_{ij} : Sequence of actual journeys j assigned to i
$C(x)$: True center of the set

5.3 Settings

We evaluate two settings of our genetic algorithm against traditional approaches. The traditional approaches are state-of-the-art techniques that are used to cluster and summarize sets of sequential and categorical data. Fig. 6 depicts the approach at a high-level. As can be seen, with traditional approaches, we first build a distance metric. We use the edit distance with a constant cost operation set to 1. Once the distance matrix is built, we create k clusters. Because we do not know the number of representative journeys to be found, we test using from

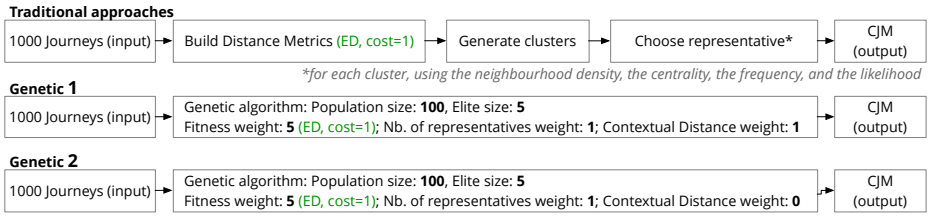

Fig. 6. Approach used to evaluate our clustering algorithm from traditional approaches.

2 to 12 clusters and use the squared Calinski-Harabasz index described in [14] to return the most statistically relevant. Then, we return the best representatives of each cluster using the neighborhood density, the centrality, the frequency, or the likelihood using the package Traminer available in R [17]. These techniques do not use the contextual data. Hence, to allow for a fair comparison, we compare these techniques with a version of our genetic algorithm that does not use contextual data and which was presented in [6]. We call this version $Genetic_1$. We also test our genetic algorithm with a version that considers the contextual data, called $Genetic_2$. Note that both the traditional and genetic approaches use the same techniques to find k_h and the distance is measured using the edit distance with a constant cost operation set to 1. Due to the non-deterministic nature of the genetic algorithm, we run it ten times for each setting.

5.4 Results

Fig. 7 shows the external evaluation metrics. It can be seen that the best solution is the $Genetic_2$, highlighting that considering the contextual information when grouping journeys improves the quality. Next, the best solution that does not use contextual data is $Genetic_1$ proposed in [6].

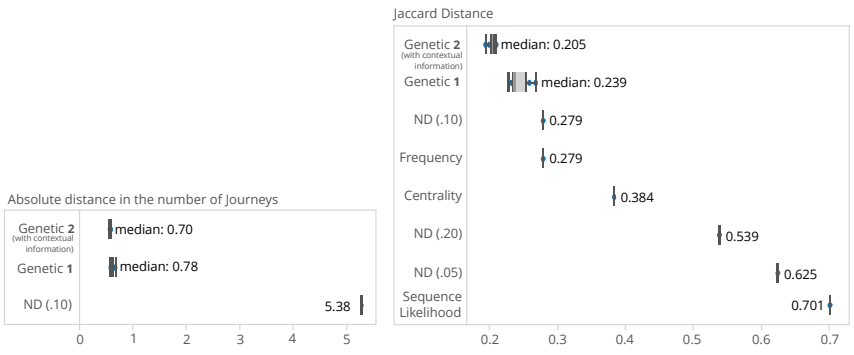

Fig. 7. External evaluation. The genetic algorithm that uses the contextual information (i.e., $Genetic_2$) performs best.

The internal evaluation of Fig. 8 shows that not only does the genetic algorithm outperform the traditional approaches, it also proposes a better solution than the ground truth. This can be explained by the fact that when we inject noise, we potentially change the optimal solution.

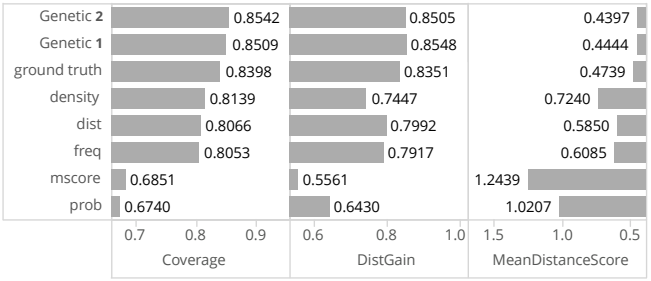

Fig. 8. Internal evaluation. The $Genetic_2$ has the best coverage and mean distance while $Genetic_1$ has the best distance gain.

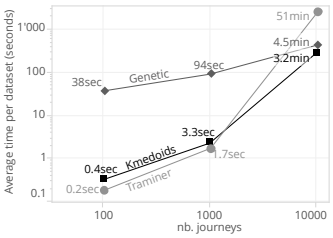

Fig. 9. Comparing the execution time per dataset for 100, 1'000, and 10'000 journeys.

Finally, the execution time to for 1,000 actual journeys is much faster using the approach with the techniques implemented in Traminer [17] compared to our genetic approach. We compare how the different algorithms scale when the number of journeys increases. Hence, we ran each configuration five times with the 40 different datasets. Fig. 9 summarizes the results. As can be seen, the algorithms implemented in Traminer are orders of magnitude faster than our approach when dealing with 100 or 1,000 journeys. However, note that our algorithm has a better scaling potential when the number of journeys grows. All the algorithms tested tend to be slow and will not scale when dealing with several thousand journeys.

6 Experiments Using Real Datasets

This section reports on the experiments with a real dataset, the goal being to illustrate how a change in the settings impacts the results. We used a publicly available dataset[5] describing the activities performed throughout the day by Chicago's citizens. There are 15 types of activities, such as, 'being at home', 'attending class', 'going shopping', or 'doing households errands'. In the context of this dataset, a journey is the sequence of activities starting from the morning

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

until the night. Typically, 'being at home' → 'attending class' → 'being at home' is a journey consisting of three activities. The total number of journeys is 29,541 and there are 123,706 activities (with an average of 4.817 activities per journey). This dataset is interesting not only for the relatively large number of data points describing life trajectories, but also because of the available detailed contextual data, such as information on the citizens' demographics.

The goal of the experience is to show the influence of taking the citizen's age in consideration when measuring the distance between journeys. Fig. 10 shows the results using three different configurations. In configuration 1, we did not leverage the contextual data (i.e., the contextual distance weight is set to 0). We interpret the resulting CJM as follows. The first journey represents people going to 'work', going back 'home' at noon, and returning to 'work' in the afternoon. The second journey is close to the first one, the main difference being that people do not seem to go back 'home' at noon. The third journey shows citizens being at 'home', going 'shopping' twice in the afternoon, and going back 'home'.

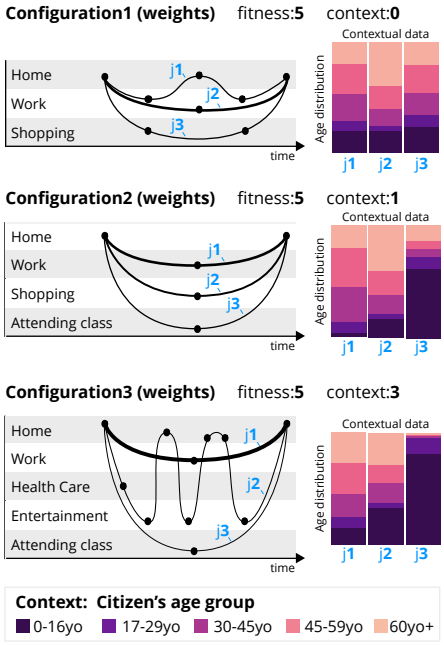

Fig. 10. Results with real dataset using three configurations

In configuration 2, we test the effect on the resulting CJM when considering the ages of the customers. Therefore, we changed the weight put on the contextual distance from 0 to 1. As can be seen in Fig. 10, three representative journeys were generated. Each of these journeys has three touchpoints. They start from 'home' and finish at 'home'. In between, the first journey has the activity 'work', the second one has the activity 'shopping', and the last one the activity 'attending class'. It is interesting to note the effect of the configuration on the contextual data (the distribution charts on the right side of Fig. 10). Indeed, while the age was equally distributed for each journey in configuration 1, we can observe that the age is discriminant in configuration 2. For instance, more than half of the citizens in the journey j_3 are under 16 years old, while this population represents only 8.7% of the entire dataset.

In configuration 3, we show the effect when we increase the weight put on the contextual distance parameter. Journeys j_1 and j_3 are identical to those in configuration 2. However, a new and rather complex journey j_2 emerges. We observe that the distribution is impacted when giving more weight to homogeneity.

We interpret the result as follows: Citizens younger than 29 years old tend to have two typical patterns of activities involving either 'school' or 'entertainment' while the most typical journeys for the other citizens involve 'work'.

Of course, this is an extremely simplified overview of the data. For the almost 30,000 actual journeys in the event logs, there are numerous unique actual journeys that differ from the representative journeys we get from these three configurations. By letting the user choose the weight for each parameter, we let them explore different perspectives of the data. We claim that the best parameters depend on the dataset, the business context, and the goal of the exploration.

7 Conclusion

Our genetic approach to summarizing a set of customer journeys with the purpose of displaying them on a CJM offers an interesting alternative to approaches used in social sciences for three reasons. First, the quality of the results is better, which is true using both internal and external evaluation metrics. Second, the weights of the three quality criteria are a flexible way to analyze a dataset under different perspectives. All the other parameters, such as the number of representative journeys to display or the length of the representative journeys are left entirely to the genetic algorithm. Third, in addition to the sequence of activities, our genetic algorithm can leverage contextual data to group similar journeys. By doing so, we provide a way to summarize insights from customers that are hidden in the data.

We tackle the task of building a CJM from event logs as a single-objective optimization problem so that a single 'best' CJM is returned. Due to the inherent conflicting objectives of the quality criteria, we acknowledge that a multi-objective approach might be a relevant choice that we did not investigate.

We show that techniques from social sciences are also useful for studying customer trajectories. As suggested by Gabadinho et al., "The methods are by no way limited to social science data and should prove useful in many other domains" [7]. This present study supports this claim and highlights how research from social science can benefit our understanding of customers. At a time when a customer-centric culture has become important [18], or even a matter of survival according to [19], we anticipate that research at the crossroads between data science, marketing, and social sciences will be key to a full understanding of customer experiences.

References

1. Lemon, K.N., Verhoef, P.C.: Understanding customer experience throughout the customer journey. Journal of Marketing (2016)
2. Gürvardar, İ., Rızvanoğlu, K., Öztürk, Ö., Yavuz, Ö.: How to improve the overall pre-purchase experience through a new category structure based on a compatible database: Gittigidiyor (ebay turkey) case. In: International Conference of Design, User Experience, and Usability. pp. 366–376. Springer (2016)

3. Peltola, S., Vainio, H., Nieminen, M.: Key factors in developing omnichannel customer experience with finnish retailers. In: International Conference on HCI in Business. pp. 335–346. Springer (2015)
4. Research priorities 2014-2016. Tech. rep., Marketing Science Institute (2014), http://www.msi.org/uploads/articles/MSI_RP16-18.pdf
5. 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)
6. Bernard, G., Andritsos, P.: Discovering customer journeys from evidence: a genetic approach inspired by process mining. In: Proceedings of the Forum and Doctoral Consortium Papers Presented at the 31st International Conference on Advanced Information Systems Engineering (CAiSE 2019) (2019)
7. Gabadinho, A., Ritschard, G., Studer, M., Mueller, N.S.: Summarizing sets of categorical sequences: selecting and visualizing representative sequences pp. 94–106 (October 2009)
8. Gabadinho, A., Ritschard, G., Studer, M., Müller, N.S.: Extracting and rendering representative sequences. In: International Joint Conference on Knowledge Discovery, Knowledge Engineering, and Knowledge Management. pp. 94–106. Springer (2009)
9. Olding, E., Cantara, M., Robertson, B., Dunie, R., Huang, O., Searle, S.: Predicts 2016: Business transformation and process management bridge the strategy-toexecution gap. Tech. rep., Gartner (November 2015), https://www.gartner.com/doc/3173020/predicts–business-transformation-process
10. 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)
11. Harbich, M., Bernard, G., Berkes, P., Garbinato, B., Andritsos, P.: Discovering customer journey maps using a mixture of markov models (2017/12)
12. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. In: Soviet physics doklady. vol. 10, pp. 707–710 (1966)
13. Schwarz, G., et al.: Estimating the dimension of a model. The annals of statistics **6**(2), 461–464 (1978)
14. Caliński, T., Harabasz, J.: A dendrite method for cluster analysis. Communications in Statistics-theory and Methods **3**(1), 1–27 (1974)
15. Buijs, J.C., van Dongen, B.F., van der Aalst, W.M.: A genetic algorithm for discovering process trees. In: Evolutionary Computation (CEC), 2012 IEEE Congress on. pp. 1–8. IEEE (2012)
16. De Medeiros, A.A., Weijters, A.: Genetic process mining. In: Applications and Theory of Petri Nets 2005, Volume 3536 of Lecture Notes in Computer Science. Citeseer (2005)
17. Gabadinho, A., Ritschard, G.: Searching for typical life trajectories applied to childbirth histories. Gendered life courses-Between individualization and standardization. A European approach applied to Switzerland (2013), 287–312 (2013)
18. Verhoef, P.C., Lemon, K.N., Parasuraman, A., Roggeveen, A., Tsiros, M., Schlesinger, L.A.: Customer experience creation: Determinants, dynamics and management strategies. Journal of retailing **85**(1), 31–41 (2009)
19. Goran, J., LaBerge, L., Srinivasan, R.: Culture for a digital age. Tech. rep., McKinsey (July 2017), https://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/culture-for-a-digital-age