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]. 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]. A challenge faced by many practitioners is that of understanding the large number of combinations of activities that may exist when consuming a service. As a result, new methods employed to design, analyze, and understand customer journeys are emerging from the industry and are becoming popular among researchers. One of these conceptual methods that will be the focus of

this work, is called the Customer Journey Map (CJM). By showing typical journeys experienced by customers across several touchpoints, a CJM helps to better understand customers' trajectories [5].
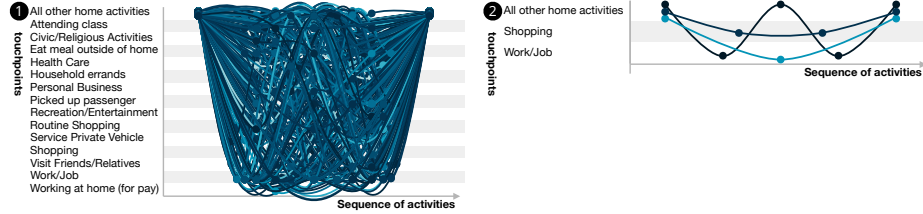

Fig. 1. ❶ Less than 0.01% of the entire dataset on a CJM, and ❷, a summary of the dataset using 3 representatives.

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).
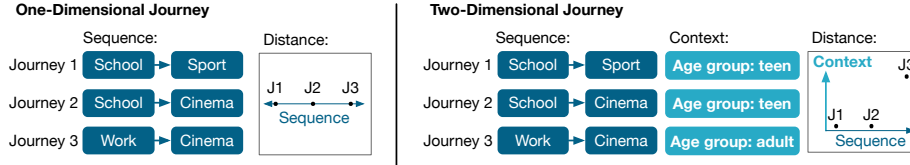

Fig. 2. Measuring the distance among three journeys with and without the context.

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 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

---

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

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 journeys to use, 2) the weight of the sequence of activities, and 3) the weight of the contextual data. In the evaluation section, we demonstrate that we outperform 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 discusses the discovery of customer journeys. In Section 3, we outline the existing techniques. Section 4 depicts our genetic algorithm. In Section 5, we evaluate our approach using internal and external evaluation metrics. Section 6 illustrates CJMs produced by our algorithm. Finally, in Section 7 we conclude the paper.

2   Customer Journey Discovery

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

Definition 1 (Touchpoint): We define a touchpoint as the interaction between a company's products or services and a customer [5]. 'Buying a product' or 'complaining about a product' are two examples of touchpoints in an online retail context. We define t as the touchpoint while T is the collection 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. To improve readability, we refer to touchpoints using alphabetical characters (e.g., 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. In Fig. 1, ❶, 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): An event log is denoted by $J_{\mathcal{A}}$, which is the list of all journeys observed by customers.

Definition 5 (Customer Journey Map): By using representative journeys, a CJM summarizes customer trajectories. Let a customer journey map $J_{\mathcal{R}}$ be the set of all the J_rs summarizing $J_{\mathcal{A}}$. $k_{\mathcal{R}}$ denotes the total number of journeys. Typically, the part ❷ of Fig. 1 is a CJM, $J_{\mathcal{R}}$, containing three representative journeys summarizing an event log.

We define the discovery of customer journeys 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 can be seen as an unsupervised clustering task. This task has interesting challenges. First, choosing the number of representatives is

difficult. When the goal is to have a general overview about a particular dataset, it seems reasonable to display only few journeys so the CJM is readable. However, discovering a few dozens of 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 still summarizes the actual journeys well. These phenomena were 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

There is a body of work in social sciences that is relevant to the summarization of customer journeys. Typically, in [7, 8], Gabadinho et al. are summarizing observed sequences with representatives. They define a representative as *"a set of non-redundant 'typical' sequences that largely, though not necessarily exhaustively, cover the spectrum of observed sequences"* [7]. The authors propose four ways to choose a representative. *'Frequency'*, (1), considers the most frequent sequence as the representative. *'Neighborhood density'*, (2), selects the sequence that has the most neighbors in a defined diameter. *'Centrality'*, (3), picks the most central object, i.e., the one having the minimal sum of distances from all other objects. Finally, *'sequence likelihood'* considers a sequence derived from the first-order Markov model.

Since Process Mining operates in a bottom-up fashion, from data all the way to the discovery of conceptual models, it 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.

In [9], we propose CJM-ex, an online tool to explore CJMs. Because it uses a hierarchical structure, it allows to efficiently navigate the space of journeys in CJMs. In [10], it was shown that customer journey maps can be discovered using Markov models. 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.

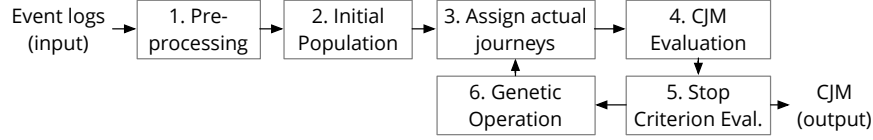# 4   A Genetic Algorithm for Customer Journey Discovery

Fig. 3. Our genetic approach

Our work is inspired by the genetic approaches proposed in [11–13] to discover business process models from event logs. However, we tailored it towards CJMs by introducing specific evaluation metrics suited for them. Fig. 3 depicts the main phases: (1) a preprocessing phase, (2) generating the initial population, (3) assigning each actual journey to its closest representative, (4) evaluating the quality of the CJMs, (5) the stopping criterion evaluation, and (6) creating new CJMs by applying some genetic operations. We introduce these phases in details while the Fig. 4 illustrates how it works.
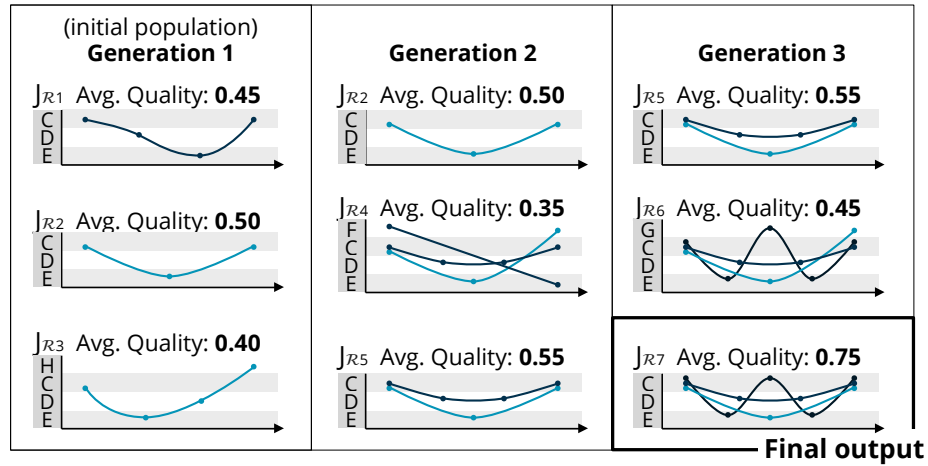

Fig. 4. Illustration of the genetic process to discover the best CJM

4.1   Preprocessing

We assume that the representative journeys will be similar to the most frequent patterns journeys. Hence, to reduce computation time, we extract the

most frequent patterns that we use to create new journeys and to generate the initial population. Let Top_{ℓ_n} be the n most occurring patterns of length ℓ and $Top_n \supseteq Top_{\ell_{[2,m]}}$ be the list of all the most occurring patterns of lengths 2 to m. By using Top_n, we reduce the execution time by two without impairing the quality of the final output.

4.2   Initial Population

We start by generating a set of random CJMs. There are created by picking journeys from from Top_n. 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

In order to evaluate the quality of the generated CJMs, it is required to assign each actual journey to its closest representative. The closeness between J_a and J_r is measured using the Levenshtein distance [14]. This metric counts the number of edit operations (i.e., deletions, insertions, and substitutions) required to match two sequences. Typically, the distance between $\langle \textsc{xyz} \rangle$ and $\langle \textsc{xyyw} \rangle$ is 2. The closest representative is the one having the smallest Levenshtein distance with the actual journeys. If a tie occurs, we assign the actual journey to the representative having the less journeys already assigned to it. When the actual journeys have been assigned to their respective closest representative, we can start evaluating the quality of the CJMs.

4.4   CJM Evaluation Criteria

We define four criteria to evaluate the quality of CJMs: (1) the fitness, (2) the number of representatives, (3) the contextual distance, and (4) the average quality.

Fitness. Using the Levenshtein distance [14], the fitness quality measures the distance between the representative sequence and the actual journeys assigned to it.

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

where

$\sigma_{\mathcal{A}_i}$: i$^{\text{th}}$ actual (observed) sequence in $J_\mathcal{A}$
$\sigma_{\mathcal{R}_j}$: j$^{\text{th}}$ representative contained in $J_\mathcal{R}$
$Length(x)$: Number of touchpoints in the sequence x

When an actual journey is strictly identical to its representative journey, the fitness is 1.

Number of Representatives. The more representative journeys we use, the more likely the fitness will be high. Hence, without a metric that helps keeping a low number of representatives, we would obtain a final CJM with several thousands representative journeys. Therefore, the goal of the metric is to keep a low number of representatives. To guide the algorithm towards an 'ideal' number of representatives, we borrowed a clustering technique that helps in choosing the number of clusters. More specifically, we used the Calinski-Harabasz index [15]. Let k_h be the optimal number of clusters returned by the Calinski-Harabasz index. To evaluate the quality, we measure the distance between $k_{\mathcal{R}}$ and k_h using the following distribution function:

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

where

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

We set the value of the midpoint, x_0, to 5 for all our experiments. The intuition behind this parameter is the following. If we have 11 representative journeys on a CJM and the ideal number of journeys is 6, we would have a quality of 0.5 (midpoint) because the distance between 11 and 6 is 5. Often, the final output will have a number of representative journeys that differs from k_h. This is due to the fact that there are others evaluation criteria.

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:

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

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 get the average weighted quality by getting the arithmetic mean of: (1) the fitness, (2) the number of representatives, and (3) the contextual-distance.

4.5   Stopping Criterion

Once we assess the quality of generated CJMs, we assess the stopping criterion. Inspired by the process mining genetic algorithms, [11, 13], we found three stopping criterion: (1) a certain amount of generations has been reached, (2) the quality does not improve anymore, or, (3), a quality threshold has been reached. Predicting the quality that will be reached by a CJM is difficult. Hence, we believe that the third way stopping criteria is not advisable. If a stopping criteria is met, the algorithm stops, returning the best $J_\mathcal{R}$. If none of the stopping criterion is met, we generate new candidates by recursively calling a function that generates the next population, described in the next section.

4.6   Genetic Operations

Before transforming the CJMs, we evaluate and rank them by average quality. We copy a fraction (i.e., e) of the best CJMs in a set named *elite*. In Fig. 4, the elite size is 1. In our experiments, we set the elite size to 5.

By keeping the best CJMs as-is, we ensure that the quality will increase or stay steady. We also generate $p - e$ new CJMs using the following operators. *(1) Add a random journey (mutation):* A sequence from Top_n is added to $J_\mathcal{R}$. *(2) Add an existing journey (crossover):* A journey from the elite population is added to $J_\mathcal{R}$. *(3) Delete a journey (mutation):* A journey is removed from $J_\mathcal{R}$. Nothing happens if $J_\mathcal{R}$ contains only one journey. *(4) Add a touchpoint (mutation):* A touchpoint is inserted in one of the existing journeys. *(5) Delete a touchpoint (mutation):* A touchpoint is removed from $J_\mathcal{R}$.

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.

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). Once new $J_\mathcal{R}$s have been created, we return to the evaluation phase as shown in Fig. 3.

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 existing techniques.
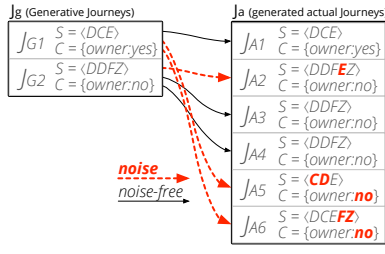
5.1   Datasets

Fig. 5. Dataset with 50% of noise.

To evaluate the results of our algorithm, we generated synthetic event logs that simulate journeys using generative journeys. A generative journey is a known sequence of activities with a known set of characteristics from which we generate the event logs. These generative journeys represent the ground truth. If we used only those known generative journeys to produce the dataset, we would get 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. Having group of similar journeys that slightly differ from a representative is a more realistic setting. To achieve this, we add some noise to the generated journeys. Typically, when 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 rely both on external and internal evaluations. The primer evaluates the results by using the generative journeys. Because we add some random noise, it might be the case that the ground truth is not the best solution. For this reason, we also use internal evaluations that relies on cluster analysis techniques. These metrics are described in [8].

External Evaluation - Distance in Number of Journeys. Measure the distance between the number of generative journeys and the number of representative journeys. We evaluate this metric using the following equation:

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

External Evaluation - Jaccard Distance. We use the Jaccard distance to evaluate how well we can retrieve the generative journeys.

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

---

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

Internal Evaluation - Mean distance [8]. This metric measures the distance between the actual journeys and their respective representative.

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

where

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

Internal Evaluation - Coverage [8]. This metric represents the density of journeys in 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 journeys attached to the representative i
S_i : Representative sequence i
S_{ij} : Sequence of journeys j attached to the representative i

Internal Evaluation - Distance gain [8]. This metric measures the gain in using representative journeys rather than the medoid of the dataset.

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

where

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

5.3   Settings

We test two settings of the algorithm against traditional approaches. The traditional approaches are state-of-the-art techniques that are used to cluster and
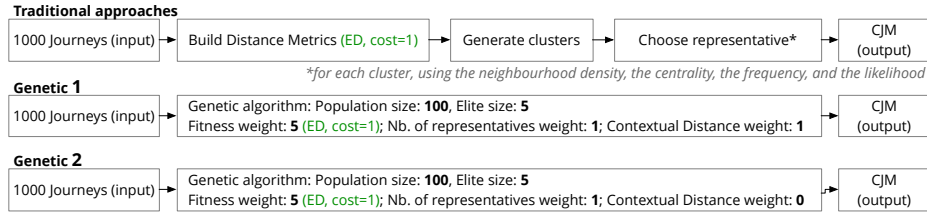
Fig. 6. Approach used to evaluate our clustering algorithm from traditional approaches.

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 2 to 12 clusters and use the squared Calinski-Harabasz index described in [15] to return the most statistically relevant. Next, we get the best representatives using the neighborhood density, the centrality, the frequency, or the likelihood using Traminer [16]. 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. To account for the fact that the genetic algorithm is non-deterministic, we run the algorithm 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].
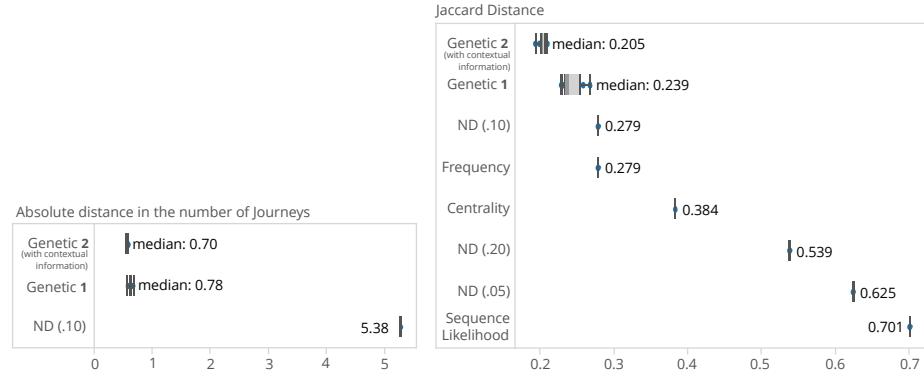
Fig. 7. External evaluation. The genetic algorithm that uses the contextual information (i.e., $Genetic_2$) performs best.

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.
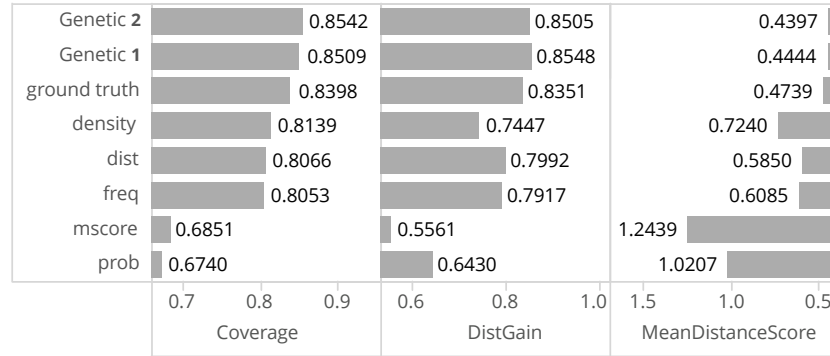

Fig. 8. Internal evaluation. The $Genetic_2$ has the best coverage and mean distance while $Genetic_1$ has the best distance gain.
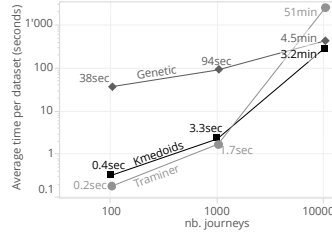
Fig. 9. Execution time for 100, 1'000, and 10'000 journeys.

The execution time for one thousand journeys is improved using Traminer [16] 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 until the night. Typically, 'being at home' $\rightarrow$ 'attending class' $\rightarrow$ '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.
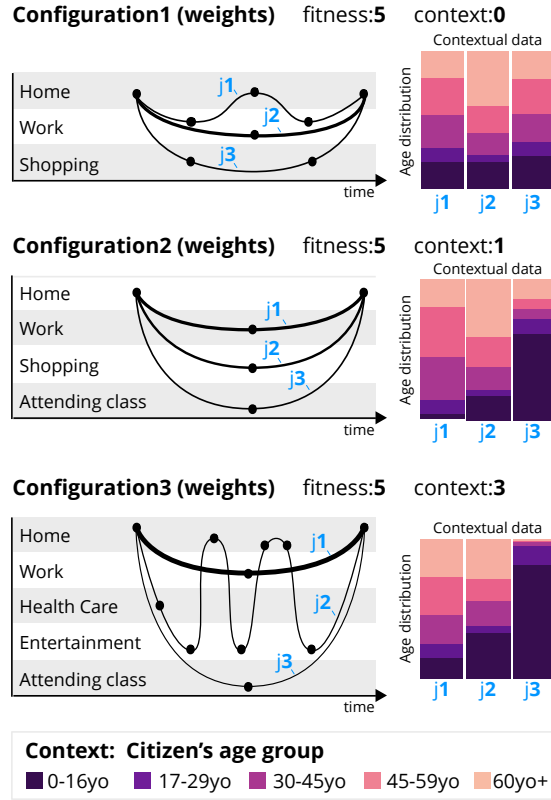
---

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

Fig. 10. Results with real dataset using three configurations

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'.

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 challenging 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.

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. 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)
10. Harbich, M., Bernard, G., Berkes, P., Garbinato, B., Andritsos, P.: Discovering customer journey maps using a mixture of markov models (2017/12)
11. 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)
12. Vázquez-Barreiros, B., Mucientes, M., Lama, M.: Prodigen: Mining complete, precise and minimal structure process models with a genetic algorithm. Information Sciences **294**, 315–333 (2015)
13. 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)
14. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. In: Soviet physics doklady. vol. 10, pp. 707–710 (1966)
15. Caliński, T., Harabasz, J.: A dendrite method for cluster analysis. Communications in Statistics-theory and Methods **3**(1), 1–27 (1974)
16. 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)