A Chain Composite Item Recommender for Lifelong Pathways

Patrick Marcel, Alexandre Chanson, Thomas Devogele, Nicolas Labroche, Nicolas Ringuet, Vincent t'Kindt

A Chain Composite Item Recommender for Lifelong Pathways

Alexandre Chanson[1] , Thomas Devogele[1], Nicolas Labroche[1] ,
Patrick Marcel[1(✉)] , Nicolas Ringuet[1,2], and Vincent T'Kindt[1]

[1] University of Tours, Tours, France
{alexandre.chanson,thomas.devogele,nicolas.labroche,patrick.marcel,
vincent.tkindt}@univ-tours.fr, nicolas.r@neolink.link
[2] Neolink, Blois, France

Abstract. This work addresses the problem of recommending lifelong pathways, i.e., sequences of actions pertaining to health, social or professional aspects, for fulfilling a personal lifelong project. This problem raises some specific challenges, since the recommendation process is constrained by the user profile, the time they can devote to the actions in the pathway, the obligation to smooth the learning curve of the user. We model lifelong pathways as particular chain composite items and formalize the recommendation problem as a form of orienteering problem. We adapt classical evaluation criteria for measuring the quality of the recommended pathways. We experiment with both artificial and real datasets, showing our approach is a promising building block of an interactive lifelong pathways recommender system.

Keywords: Chain composite item recommendation · Orienteering problem

1 Introduction

We consider in this work the problem of building a recommendation system to support social actors and beneficiary users in the interactive co-construction of a personal lifelong project, for example the assistance of job seekers, or elderly home support. Such a system would help different social actors to interact for building a personal project for beneficiaries from a very large set of possible actions pertaining to health, social or professional aspects. Such actions have an intrinsic cost, they should complement well one another and be relevant for the beneficiary, and doable in a period of time suitable for the beneficiary. We denote the long-term sequence of actions proposed to beneficiaries as their *lifelong pathways*. Recommending lifelong pathways raises many challenges: (i) formalizing the recommender output, i.e., lifelong pathways, (ii) formalizing the problem of computing pathways adapted to beneficiary users, (iii) determining the quality of the recommended pathways.
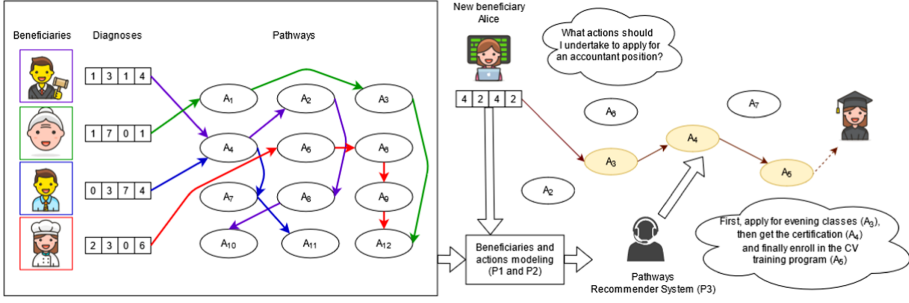
Fig. 1. Pathway Recommender System (PRS). P_1, P_2 and P_3 relate to the formal problems defined in Sect. 2.

Let's consider the example of Alice as pictured in Fig. 1 who wants to know how to achieve her personal long-term goal (apply to an accountant position). Alice has already achieved a diagnosis of her current situation, similar to previous beneficiaries for which diagnoses and pathways were observed beforehand. The objective of the pathways recommender system (PRS for short) is to determine from there what would be the most relevant next actions for Alice to undertake to achieve her long-term goal. This is a complex problem as it is heavily constrained: Alice has a limited time budget, and has to follow a certain learning curve so that each new action can easily build over the previously achieved actions. As a consequence, PRS has to estimate from previously observed beneficiaries, diagnoses, actions and their respective pathways, what would be the cost and the relevance for Alice to undertake an action. Similarly, the PRS needs to estimate a distance between actions to smooth the learning curve of Alice. Finally, the number and the diversity of actions make this problem difficult to solve.

In this paper, we contribute with a first approach that sets the bases of lifelong pathways recommendation. We model the problem, study its complexity, propose a resolution and optimizations, as well as evaluation criteria. Consistently with previous works, we consider that recommending such composite items calls for non traditional recommendation approaches [2]. More precisely, we consider lifelong pathways as particular chain composite items (CCI) [8,10] built from atomic actions, that have to satisfy the classical CCI properties of maximality (in terms of relevance), validity (in terms of beneficiary budget), and compatibility (in terms of action composition). The recommendation problem is formalized as a form of orienteering problem [6,13], for which we implemented two approaches for finding exact and approximate solutions. We adapt classical evaluation criteria for measuring the quality of the recommended pathways. We experiment with both artificial and real datasets, showing our approach is a promising building block of an interactive lifelong pathways recommender system.

This paper is organized as follows. Section 2 gives the formal background and defines the problem, while Sect. 3 presents our approach. Section 4 details the

experimental validation. Section 5 discusses related work, and Sect. 6 concludes and draws perspectives.

2 Formal Definitions

In this section, we briefly define the concepts on top of which we build our problem statement for the recommendation of lifelong pathways to beneficiaries.

2.1 Model

We consider a set A of atomic actions representing various personal, professional or health steps that can be undertaken by a beneficiary to further his life goal.

Each action $a \in A$ is associated with a cost noted $cost(a)$ that represents the action difficulty. Given two actions $a_1, a_2 \in A$, we consider a distance between them, noted $dist(a_1, a_2)$ that represents how smooth the progression of the beneficiary would be by pairing these two actions in a pathway.

In what follows, action costs and distances constitute the action profiles. Notably, we do not differentiate between types of actions, since this would be use case-dependent. However, they could be easily introduced by adapting cost and distance measure.

A beneficiary b is represented by a tuple of features $\overrightarrow{f}$, called a diagnosis (see Sect. 4.1 for examples of features).

A profile for b is a vector of relevance scores, one for each action. For a beneficiary b and an action a, we note $rel_b(a)$ the relevance of b in a. Given a beneficiary b and a set of actions A, a pathway p_b for b is a sequence of actions from A.

2.2 Problem Formulation and Complexity

Our recommendation problem has the following inputs: (i) a set of actions A for forming pathways, (ii) a set of former beneficiaries B represented by their diagnoses D and the set of pathways P they have undertook, formed by actions in A, (iii) a new beneficiary b with diagnosis $\overrightarrow{f}$. The problem consists of recommending for b a pathway p_b with actions of A, based on D, P and $\overrightarrow{f}$.

We decompose our recommendation problem in three sub-problems:

P_1 compute action profiles in terms of action cost and distance between actions,
P_2 compute a user profile in terms of relevance in actions,
P_3 compute recommendations of pathways.

Problem 1 (Action profiles). *Let B be a set of beneficiaries, A be a set of n actions and P be a set of pathways with actions in A, for the beneficiaries in B. The problem consists of computing a profile for each action a of A, i.e., $cost(a) \geq 0$ and, for all $a' \in A$, a metric $dist(a, a')$.*

Problem 2 (Beneficiary profile). *Let B be a set of beneficiaries, A be a set of n actions and P be a set of pathways with actions in A, for the beneficiaries in B. Let b be a new beneficiary not in B, with diagnosis $\vec{f}$. The problem consists of computing a profile for beneficiary b, i.e., a vector of relevance score for each action in A, noted $\vec{b} = \langle rel_b(a_1), \ldots, rel_b(a_n) \rangle$.*

Problem 3 (Pathway recommendation). *Let A be a set of n actions, each associated with a positive cost $cost(a_i)$ and a positive relevance score $rel_b(a_i)$. Each pair of action is associated with a distance $dist(a_i, a_j)$. Let b be a beneficiary with profile $\vec{b}$, and let t be a budget in terms of days left to the beneficiary for fulfilling the recommended pathway. The optimization problem consists in finding a sequence $p_b = \langle a_1, \ldots, a_m \rangle$ of actions, $a_i \in A$, without repetition, with $m \leq n$, such that:*

1. *maximize $\sum_{i=1}^{m} rel_b(a_i)$ (maximality)*
2. *minimize $\sum_{i=1}^{m-1} dist(a_i, a_{i+1})$ (compatibility)*

subject to

3. *$\sum_{i=1}^{m} cost(a_i) \leq t$ (validity).*

Complexity of P_3. Problem P_3 is strongly NP-hard [7] since the TSP can be reduced to it. Indeed, any instance of TSP can be turned into an instance of P_3 by assigning a cost of 0 to all cities along an relevance of 1 and leaving the distance unchanged, it is then trivial to show that an optimal solution to this P_3 instance is optimal for the original TSP instance. This result means that, unless $P = NP$, P_3 can only be solved to optimality by algorithms with a worst-case time complexity in $A^*(c^n)$, with c a positive root and n the size of A.

3   Problem Resolution

To remain close to what is empirically observed, problems P_1 and P_2 are mostly treated as learning problems, where distances between actions and the relevance of an action for the beneficiary are learned from the set of existing diagnoses and pathways, that were devised by professional social actors. The cost of an action can be more trivially extracted from past pathways available data.

3.1   Computing Action Profiles (P_1)

We model each action a in A as a vector of features, computed as the average of diagnoses of former beneficiaries who were recommended the action a. The cost of actions is simply fixed to the median time spent for each action, as reported in the set of past pathways P. The median is used due to its robustness to extreme values. Note that, should this information be unavailable, a machine learning based approach similar to the ones described next for distances and relevance computation, should be applied.

Learning distances between actions corresponds to a traditional metric learning problem. Classical approaches addressing this problem include (i) transformation to classification task [5,9,14], or (ii) weekly-supervised metric learning [15]. Regarding (i), distances are usually linear combinations of feature-wise distances between objects, obtained by fitting a linear classifier (e.g., SVM) over pairs of objects labeled positively if objects should be in the same group and negatively otherwise.

As to (ii), weekly-supervised metric learning approaches aim at defining a new space based on input pairwise Must-Link or Cannot-Link constraints that specify respectively if two points should be close or distant in the output space. A traditional method is to learn a generalized Mahalanobis distance [15], that is basically a generalization of the Euclidean distance defined as follows:

$$d_M(a_1, a_2) = \sqrt{(a_1 - a_2)^t M (a_1 - a_2)} \tag{1}$$

with $(a_1, a_2) \in A^2$ and where $M \succeq 0$ is a positive semi-definite matrix to ensure that d_M is a proper metric. In its simplest form, i.e. when $M = I$, d_M is a Euclidean distance and when M is diagonal, d_M changes the relative weights of each features in the computation of the distance. Finally, in the general case, it can express more complex scaling and relations between features. The method proposed in [15] relies on the following optimization problem to find the expression of M:

$$\min_M \sum_{(a_i, a_j) \in S} d_M^2(a_i, a_j) \tag{2}$$

$$\text{s.t.} \sum_{(a_i, a_j) \in D} d_M(a_i, a_j) \geq 1 \tag{3}$$

$$M \succeq 0 \tag{4}$$

where S (resp. D) is the set of constraints indicating that 2 actions should (resp. should not) appear in the same pathway in our case. Equation (3) is added to avoid the trivial solution $M = 0$. We followed this approach to learn the distances between actions.

3.2   Computing Beneficiary Profile (P_2)

Preliminaries. Recall from Sect. 2.2 that any beneficiary $b \in B$ is represented by a tuple of features $\overrightarrow{f} = \langle b_1, \ldots, b_f \rangle$. It is then possible to determine for each action $a \in A$ if beneficiary b has undertaken it. Thus, each action a can be represented as a tuple of f features $\overrightarrow{a}$, whose values are set as the average values observed for the beneficiaries $b \in B$ that undertook a in their respective pathways:

$$\overrightarrow{a} = \frac{1}{|B|} \sum_{b \in B} \langle b_1, \ldots, b_f \rangle \tag{5}$$

Collaborative Relevance Score. As can be seen in Fig. 1, the set of former pathways can be seen as a graph of the actions that were undertaken in the past. While this graph could have been used to extract action popularity scores for instance using Page Rank or centrality measures, which may be useful for recommending popular actions, our goal is to compute a different profile for each new beneficiary. As such, our problem is close to a cold start problem for a new user, for which we use a hybrid approach (in the sense of [11]), leveraging both the beneficiary diagnosis and the former diagnoses and pathways.

Precisely, we define the collaborative relevance of action $a \in A$ for beneficiary b by the probability $rel_b(a) = P(a|b)$ that beneficiary b undertakes the action a. We express this problem as a traditional binary classification task aiming to predict $True$ if action a is relevant for b or $False$ otherwise, based on the diagnosis $\vec{f}$ of b. We use a naive Bayes approach to obtain the expected probability values. By generalizing the prediction for a beneficiary based on the observations over past beneficiaries, this score accounts for a collaborative score of relevance.

3.3   Pathway Recommendation (P_3)

Problem P_3 can be seen as an extension of the orienteering problem [12] with service time. Such problem was already formulated in the Exploratory Data Analysis community, to find a series of relevant queries [6]. We choose to use the Mathematical Integer Programming (MIP) model proposed by [4] as we expect most instances to be small enough for an exact solution to be tractably found by a state-of-the-art solvers like CPLEX. Such model allows to trivially add linear constraints on a case by case basis, whenever needed. For efficiency purpose, the resolution of such problems classically reformulates the initial multi-objective problem into a single objective problem (the maximization of the relevance) and rewrites the last two remaining objectives (minimizing the time budget and the distance) as so-called epsilon constraints with upper bound on the time budget and the distance [4]. This reformulation benefits from single objective optimization mechanisms available in CPLEX.

For those larger instances of the problem that could not be handled by CPLEX, we also implemented a simple greedy algorithm to efficiently compute approximate solutions to P_3. Its principle is as follows. It starts by picking the most relevant action, and then adds subsequent actions by computing a score by dividing relevance by distance, and picking the actions achieving the best score. A windowing system is used to limit the number of comparisons (the larger the window, the greater the number of best relevant actions considered). Additionally, at each iteration, costly actions not respecting the remaining budget are pruned.

4   Tests

This section describes the experiments conducted to evaluate our approach. They answer 2 main questions: (i) how effective is our PRS approach to recommend

lifelong pathways?, (ii) to which extent PRS scales to larger instances of the problem, i.e., to a larger population of possible actions?

Problems 1 and 2 where solved using Python, sklearn library providing an implementation of a Naive Bayes classifier, and metric-learn library providing the implementation of the algorithm proposed by [15]. The mathematical model relies on CPLEX 20.10 and is implemented in C++[1]. The greedy algorithm, GreedyPRS, is implemented in Java.

4.1   Effectiveness of PRS

Dataset. In this first test, we use a real dataset named "French RSA[2]" composed of 2812 user diagnoses and 56 actions, with 14 descriptive features that represent a set of beneficiary contextual information such as "need for housing assistance", "need for childcare", "health diagnosis", "searching for a job", etc. A few null values were replaced with the most frequent ones.

Evaluation Criteria. Our evaluation scheme uses classical evaluation metrics for recommender systems, namely precision and recall of the discovered composite items when compared to the real pathways that were undertaken by the beneficiaries. However, as the recommendation task is complex, we consider precision and recall at a certain similarity threshold. In what follows, we denote by $\sim$ a similarity function between actions. We consider that there is a match between a recommended action and an expected action from the pathway if their similarity exceeds a threshold that is a parameter of the precision and recall measures, similar to what is done in [1,5].

More formally, considering: (i) the set of all actions A, (ii) a real pathway P as a set of s actions $\{p_j\}_{j\in[1,s]}$ and, (iii) the set R of m recommended actions $\{r_i\}_{i\in[1,m]}$ produced by PRS, we define the True Positive set as $TP = \{r \in R \mid r \sim p, p \in P\}$, the False Positive set as $FP = R \setminus TP$ and the False Negative set as $FN = \{a \in A \setminus R \mid a \sim p, p \in P\}$. From these sets, it is possible to compute, $Recall = \frac{|TP|}{|TP|+|FN|}$, $Precision = \frac{|TP|}{|TP|+|FP|}$ and $F1\text{-}measure = 2\frac{Precision \cdot Recall}{Precision + Recall}$.

Methodology. As explained in Sect. 3.3, P_3 is handled by reformulating objectives as epsilon constraints for which an upper bound has to be provided. We experiment with several values for these bounds. Noticeably, we consider 3 thresholds for the cost constraints expressed as the run time of actions, respectively 207, 364 and 601 days as these are the average observed period of time for respectively, below 25 years old (y.o.), between 25 and 55 y.o. and above 55 y.o. to go back to work after an unemployment period[3]. Similarly, for the distance constraint we
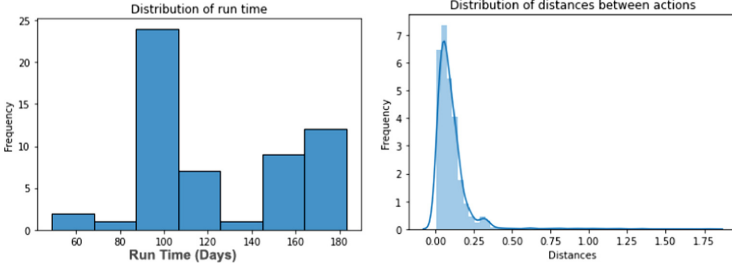
---

consider 3 distance thresholds: 3, 4, and 5 that allows to run several actions in one recommended pathway.

Figure 2 provides histograms of the distributions of run time costs and distances which assess the choices for the aforementioned thresholds.


Fig. 2. Histograms of observed run time cost for the actions in the French RSA dataset (left) and histogram of distances between actions (right).

In order to assess the accuracy of our metric learning, we run a comparison with a simple Euclidean distance on all the previous scenarios. Finally, a 3-fold cross validation is set up to ensure that train and test sets are well separated and to produce average values.

Results. Figures 3, 4 and 5 summarize the main results of our experiments in terms of F-Measure at a given similarity threshold. Note that Figs. 4 and 5 represent different groupings of the same data, averaging distance and time, respectively. Results presented are averaged by the 3-fold cross validation process and standard deviation around each mean is represented as light color areas around the plots. Importantly, the similarity is expressed as a function of the distance $d(a_1, a_2)$ between actions as follows: $sim(a_1, a_2) = 1/(1 + d(a_1, a_2))$.

Figure 3 and 4 show the importance of the distance between actions in our method. Indeed, Fig. 3 presents results with a traditional Euclidean distance between actions, that does not benefit from external knowledge of which actions should be grouped together to better fit the observed pathways. Also, the lower the similarity threshold on theses figures, the easier it is to find a match between a recommender and an observed action, and the more likely the F-measure score is to be high. Consistently with this observation, both Figs. 3 and 4 show a decreasing F-Measure score with the increase in similarity threshold. However, our learned distance between action allows for a very high F-Measure until the similarity score is close to 1 while the traditional Euclidean distance falls for smaller similarity thresholds. Interestingly, in both cases, the F-Measure score when looking for a perfect match (similarity threshold = 1) are close to 0.3 but with slightly better performances for the tuned metric. Finally, the F-Measure score is higher in case of short-term pathways which is normal as it is easier to predict for short period of time and shorter sequences for which it is easier to improve the recall.
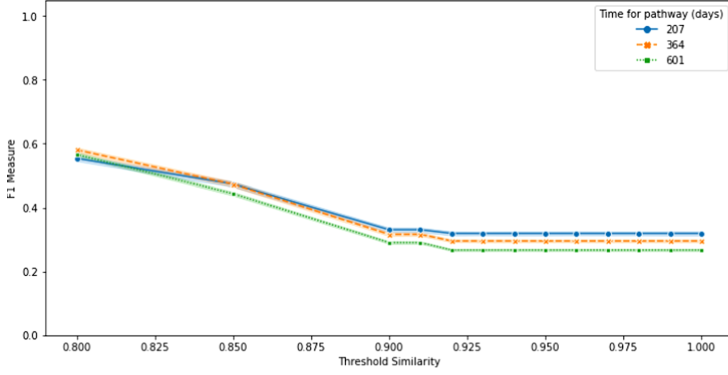
Fig. 3. F-measure scores computed for different similarity thresholds for 3 **run time costs** thresholds. Distance between action is set to an **Euclidean** distance.
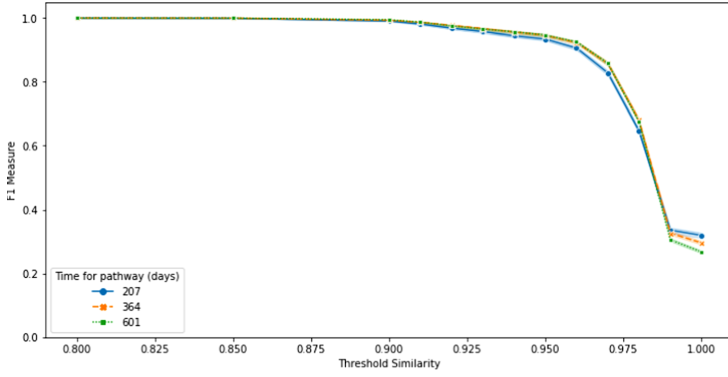
Fig. 4. F-measure scores computed for different similarity thresholds for 3 **run time costs** thresholds. Distance between action is learned as explained in Sect. 3.1.
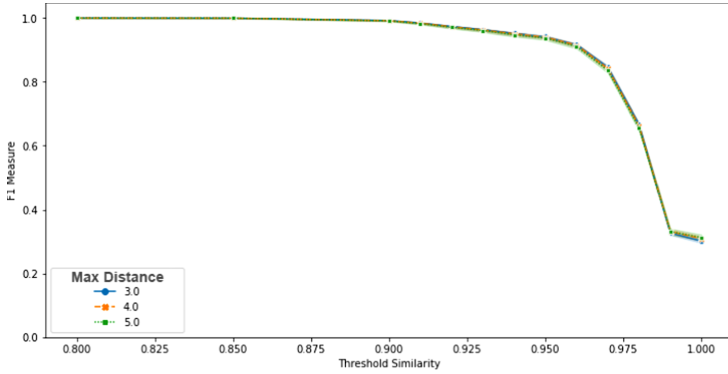

Fig. 5. F-measure scores computed for different similarity thresholds for 3 **distances** thresholds. Distance between action is learned as explained in Sect. 3.1.

Figures 4 and 5 show that Problem P_3 can be solved as efficiently if we consider constraints on the time budget as in Fig. 4 or the maximal distance as in Fig. 5. All in all, the formulation of the resolution of P_3 with epsilon constraints does not seem to be the limiting factor in the observed F-Measure results. In the next section, we study the influence of the number of actions on the resolution of Problem P_3.

4.2   Scaling to Larger Instances

For our scalability test, in order to understand what is the maximum number of actions PRS can handle, we generated 8 artificial datasets as follows. We first used a KDE-gaussian kernel to fit the distributions of relevance, cost and distance, respectively, with the samples obtained from the real dataset used for the effectiveness tests. We then generated 8 datasets of increasing sizes (100, 200, 300, 400, 500, 1000, 5000, 10,000), representing the inputs of P_3 by drawing relevance, cost and distance following the distributions obtained with the estimator.

We run CPLEX on each dataset, setting a time out of 1 h. This test was conducted on a Fedora Linux (kernel 5.11.13-200), workstation running CPLEX 20.10 on a Intel Xeon 5118 with 256 GB of main memory. The results, depicted in Fig. 6, are as expected and illustrate the hardness of the problem. For sizes above 500 actions, the time out of one hour was reached.
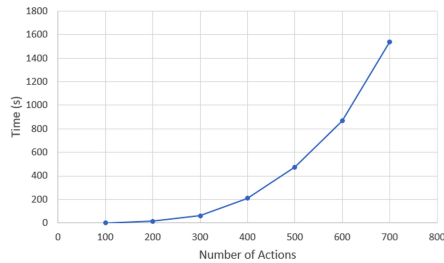
Fig. 6. Scalability of P_3 (exact)

We also run greedyPRS on the 8 datasets, varying budget (from 100 to 1000) and window size (from 10% to 100% of budget). For scalability testing, we report only the results for dataset 10,000. As shown in Fig. 7(a), contrary to the exact resolution by CPLEX, greedyPRS solves much larger instances of the problem in milliseconds. Of course, because of the simplicity of the greedy algorithm, solutions may not be feasible since the distance espilon-constraint is not checked. To understand the quality of the solutions found by greedyPRS, we also run it on the French RSA dataset, varying window size from 10 to 60. As illustrated in Fig. 7(b), the greedy approach maintains a reasonable F-measure in recommendation albeit not as good as PRS in our preliminary tests.
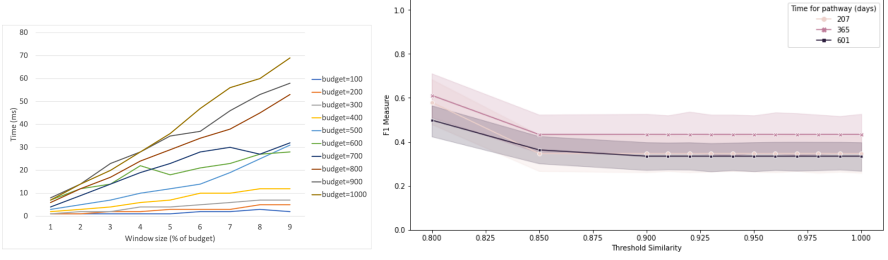
Fig. 7. Resolution of P_3 with GreedyPRS (approximate solution). Left: scalability. Right: F-measure.

5   Related Work

Our way to learn action profiles is similar to user intent discovery [5,9,14]. For instance, Guha et al. [9] discover user intent behind web searches, and obtain content relevant to users' long-term interests. They develop a classifier to determine whether two search queries address the same information need. This is formalized as an agglomerative clustering problem for which a similarity measure is learned over a set of descriptive features (the stemmed query words, top 10 web results for the queries, the stemmed words in the titles of clicked URL, etc.). A similar approach is used in [5] to discover BI user intents in BI queries.

Composite items (CIs) address complex information needs and are prevalent in problems where items should be bundled to be recommended together [2], like in task assignment in crowdsourcing or travel itinerary recommendation. CI formation is usually expressed as a constrained optimization problem, and different CI shapes require the specification of different constraints and optimizations. Our formulation of lifelong pathways is a particular case of chain shaped CIs [3,8,10], that are traditionally defined in terms of compatibility (e.g., geographic distance), validity (e.g., the total cost of an itinerary is within budget) and maximality (e.g., the itinerary should be of the highest value in terms of its POIs popularities), this last one often being used as the objective function. Retrieval of chain CIs is usually NP-hard, being reduced to TSP or orienteering problems, and has been addressed through greedy algorithm [3,10], dynamic programming or dedicated TSP strategies [8].

While being consistent with previous formulations, our formalization of lifelong pathway recommendations borrows from the Traveling Analyst Problem (TAP) [4,6]. This problem describes the computation of a sequence of interesting queries over a dataset, given a time budget on the query execution cost, and such that the distance between queries is minimized. TAP differs from the classical orienteering problem by adding a knapsack constraint to it. This formulation is close to that of [10], but the latter simplifies the problem by merging action cost and travel time budget.

In our case, similarly to [6], the distance we use has a semantics in itself and cannot be made analogous to a time or to a physical distance. Thus it must

be considered as a separate constraint. Furthermore we are not specifying any starting or finish point for the sequence as the classical orienteering problem.

6 Conclusion

This paper introduces a system for lifelong pathways recommendation. We model lifelong pathways as particular chain composite items that are built from atomic actions, and whose retrieval is formalized as a form of orienteering problem. We experiment with both artificial and real datasets, showing our approach is a promising building block of an interactive lifelong pathways recommender system. Our short term perspectives consist of including in our model a multi-stakeholders context, to conciliate the objectives or constraints of beneficiaries, social actors and services providers, as well as mechanisms for the exploration of pathways, so that users can interact with the recommender. On the longer term, we plan to add an explanation mechanism of the recommendations to increase trust of stakeholders in the proposed pathways and to improve their acceptance.

References

1. Aligon, J., Gallinucci, E., Golfarelli, M., Marcel, P., Rizzi, S.: A collaborative filtering approach for recommending OLAP sessions. DSS **69**, 20–30 (2015)
2. Amer-Yahia, S., Roy, S.B.: Interactive exploration of composite items. In: EDBT, pp. 513–516 (2018)
3. Cao, X., Chen, L., Cong, G., Xiao, X.: Keyword-aware optimal route search. Proc. VLDB Endow. **5**(11), 1136–1147 (2012)
4. Chanson, A., Labroche, N., Marcel, P., T'Kindt, V.: The traveling analyst problem, orienteering applied to exploratory data analysis. In: ROADEF (2021)
5. Drushku, K., Aligon, J., Labroche, N., Marcel, P., Peralta, V.: Interest-based recommendations for business intelligence users. Inf. Syst. **86**, 79–93 (2019)
6. Chanson, A., et al.: The traveling analyst problem: definition and preliminary study. In: DOLAP, pp. 94–98 (2020)
7. Garey, M.R., Johnson, D.S.: "Strong" NP-completeness results: motivation, examples, and implications. J. ACM **25**(3), 499–508 (1978)
8. Gionis, A., Lappas, T., Pelechrinis, K., Terzi, E.: Customized tour recommendations in urban areas. In: WSDM, pp. 313–322 (2014)
9. Guha, R.V., Gupta, V., Raghunathan, V., Srikant, R.: User modeling for a personal assistant. In: WSDM, pp. 275–284 (2015)
10. Roy, S.B., Das, G., Amer-Yahia, S., Yu, C.: Interactive itinerary planning. In: ICDE, pp. 15–26 (2011)
11. Son, L.H.: Dealing with the new user cold-start problem in recommender systems: a comparative review. Inf. Syst. **58**, 87–104 (2016)
12. Tsiligirides, T.: Heuristic methods applied to orienteering. J. Oper. Res. Soc. **35**(9), 797–809 (1984)
13. Vansteenwegen, P., Gunawan, A.: Orienteering Problems. EURO Advanced Tutorials on Operational Research (2019)
14. Wang, H., Song, Y., Chang, M., He, X., White, R.W., Chu, W.: Learning to extract cross-session search tasks. In: WWW, pp. 1353–1364 (2013)
15. Xing, E.P., Ng, A.Y., Jordan, M.I., Russell, S.J.: Distance metric learning with application to clustering with side-information. In: NIPS, pp. 505–512 (2002)