Hidden Markov Model for Tourist Visits Prediction

Theo Demessance
Léonard De Vinci, Research Center,
92 916 Paris La Défense, France
College of Intelligence and Computing
Tianjin University
theo.demessance@edu.devinci.fr

Chongke Bi
College of Intelligence and Computing
Tianjin University
bichongke@tju.edu.cn

Sonia Djebali
Léonard De Vinci, Research Center,
92 916 Paris La Défense, France
sonia.djebali@devinci.fr

Guillaume Guérard
Léonard De Vinci, Research Center,
92 916 Paris La Défense, France
guillaume.guerard@devinci.fr

***Abstract*—Nowadays, social networks is become a popular means of analyzing tourist behavior, thanks to the digital traces left by travelers during their stays on these networks. The massive amount of data generated by the propensity of tourists to share comments and photos during their trip makes it possible to model their journeys and analyze their behavior. Predicting the next movement of tourist plays a key role in tourism marketing to understand demand and improve decision support.**

In this paper, we propose a method to understand and to learn tourists' movements based on social network data analysis to predict the future movements. The method relies on a machine learning grammatical inference algorithm. A major contribution in this paper is to adapt the grammatical inference algorithm to the context of big data. Our method produces a hidden Markov model representing the movements of a group of tourists. The hidden Markov model is flexible and editable with new data. The capital city of France, Paris is selected to demonstrate the efficiency of the proposed methodology.

***Index Terms*—Tourist behavior, Hidden Markov Model, Grammatical Inference.**

I. Introduction

In 2019, *World Tourism Organisation* UNWTO[1] has recorded 1.5 billion international tourists, 4% more than the previous year. Tourism is one of the most important areas for the world economy. It is also considered to be one of the fastest-growing industries in the world [1]. In this economic context, the understanding and knowledge of travel motivations and the anticipation of tourist behavior play an essential role in tourism marketing. It can lead to the recognition of demand, to make targeted advertising [2], and help tourists to make decisions [3].

With the recent booming of digital tools and mobile internet technology, alternative sources of data to understand tourism behavior have emerged. Users of social networks, like $Tripadvisor$, $Booking$, $Facebook$, $Instagram$, tend to share openly and frequently photos, reviews, recommendations and videos on tourist places. Thus, when users share photos or reviews, geographical information is included. These geo-located data represent tourism and sociological views [4], [5].

Analyzing the behavior of tourists represents an important challenge to better monitor their movement and spreading in a given area. We can subsequently adapt supply to demand, recommend a stay to a tourist, provide relevant information for the tourism industry and management. In this paper, we focus our reviews on the prediction of tourist movements.

This paper addresses the problem of modeling and predicting the future movement of a tourist based on his present and past practices in a given area. Based on the geo-localized and temporal data information, we propose in this paper a model for predicting future tourist movement by analyzing the time sequences of places visited by a set of tourists. Our approach is to learn tourist's practices from a set of temporal sequences of places, through various methods to handle the difficulties due to big data. The proposed method uses a machine learning method building a Hidden Markov model representing the whole data set. The model can produce predictions as a recommendation of future places to visit. The model can also be updated to adapt to new data.

Our model is built from the whole data without reducing its size and without extracting a mathematical model. Thereby, we propose a new algorithm for automatic learning of grammatical inference to reduce its complexity in the context of big data. Moreover, this algorithm is designed to maintain all behavioral possibilities on the data set. The principal contributions of our works are:

1) A method to establish sequences representing a unique stay of a tourist from data set;
2) A new method of grammatical inference for processing very large data set;
3) A flexible and relevant decision-making tool to represent all tourists' movements in a data set. This decision-making tool is able to predict future visited place as recommend places to visit.

In this paper, we will first detail in Section II the related work on network analysis. In section III presents data analysis

[1] International tourism Growth continues outpacing the global economy: edition 2020.

methods, and in section IV provides the various process of our method. Our model is implemented and is the subject of a case study introduced in section V. In section VI, we conclude our paper.

II. Related work

Many data analysis methods and algorithms are used in social networks to study and understand interactions between the entities, to predict the presence of connections, or to predict the following movement of an entity [6]. Each method uses specific theories to produce an understanding of the network. To generate predictions of future movements in the networks, many studies typically use techniques and concepts from various fields of research like Machine Learning, Data Mining and Stochastic Models. These fields are not independent, studies combine methods and theories from these fields to analyze and to predict behavior. Related work is structured in these three axes by their principal contributions.

a) Machine learning methods: Machine learning is a broad area of research that focuses on identifying patterns in empirical data. Machine learning algorithms establish a mathematical model based on sample data, known as training data, to make predictions or decisions. The sample data representing temporal sequences and are seen by machine learning algorithms as time series [7]. In social networks analysis, time series analysis is widely used on a variety of purposes. Many studies are working on social networks to predict relations between elements to be recommended by exploring the evolution of topological metrics [8]. Tourism research has taken a leap forward with the development of deep learning like neural networks [9], [10]. They are robust to noise in input data, in the mapping function and can even support learning and prediction in the presence of missing values. *Wang* et al. [11] presents a model based on a neural network to estimate tourist arrivals in $Taiwan$ from $Hong\ Kong$, $United\ States$, and $Germany$. *Zhang* et al. [12] propose a deep learning analysis of photos to define tourists' behaviors in $Beijing$, $China$.

Machine learning and deep learning methods show their effectiveness over long time series containing thousands of points. With short time series, composed of less than five elements which is the case in our study, classic machine learning methods fail to provide a relevant model. The primary drawback of machine learning, especially in deep learning is the difficulty to train the model. The model can oscillate between a local generalization and an extreme generalization. It is equally complex to determine an efficient deep learning structure because of significant variations in tourists' behaviors depending on the studied places, the source of the data, and the objectives of the study.

b) Data mining methods: A second approach to generate a prediction from a data set is to use data mining methods. Data mining is a set of techniques that allow valuable information to be extracted from vast and loosely structured databases. To produce a prediction in a time-related data set, data mining methods need to decompose it into sequences. Data mining methods provide information about how an element is represented in the data set or information about relations between each element.

Related to tourists' behavior extraction, various studies combine machine learning methods with data mining. *Bao* et al. [13] predict the next location to be visited by a tourist in $Wuhan$, $China$. This approach is based on a clustering area graph of user preferences. *Majid* et al. [14] use weather and Points of Interest in Flickr to recommend the next location to visit in $Beijing$, $China$, thanks to pattern mining and similarity matrix. *Memon* et al. [15] presents a probabilistic approach for the recommendation based on various context rank and kernel estimation of Flickr geotagged data. The object is to determine the next position of the tourist in various cities of $China$. *Wan* et al. [16] employed a Bayes classifier to predict a user's latent interests in a location, based on weather, date, geotagged data of Flickr in the region of $Beijing$.

Data mining has two significant drawbacks for tourists' behaviors recommendation. First, data mining methods use thresholds due to their complexity. Thus, only the most common behaviors are kept as results of a data mining methods. Second, most methods deal with the temporal order in a sequence but not with the relationship between elements according to their position in the sequence.

c) Stochastic model: Another area of research to predict the next element of a sequence is to represent the data set of the sequences with a stochastic model. A stochastic model, or a Markov process, is a random process indexed by time, and with the property that the future is independent of the past, given the present. A finite discrete Markov chain is represented as a weighted oriented graph [17]. An arc between two nodes represents the probability of reaching the final node as the next element in a sequence.

When it comes to recommendation and prediction, Markov process models show promises [18]–[20]. Among relevant studies, *Ahmad* et Al. [21] propose a Markov model and a steady analysis to create a popularity index based on historical trajectories of mobile visitors inside $Jeju\ Island$, $South\ Korea$. *Xia* et Al. [22] proposes a semi-Markovian process to model the probability that a tourist will visit and spend a certain amount of time in each location on $Philip\ Island$, $Australia$. In [23], [24], the Markov process is used for arterial route travel time distribution and road congestion prediction. A variant of the Markov model was also used for the popularity of the routes [25].

Those methods resolve both of the drawbacks of data mining and machine learning mentioned above but are very sensitive to the distribution of elements in the sequences. Since machine learning, data mining, and stochastic models have pros and cons, we propose a hybrid method to learn tourist behaviors. This method is composed of a data analysis defining the concept of tourist and tourist stays. From these stays, we use a new machine learning method providing a prediction model. This model can be adapted to new stays and can predict behaviors that reflect the data set.

III. Data Analysis

Our database is composed of users, reviews, and geo-located locations. A location is composed of coordinates $(lat, long)$ and a rating. To characterize the location, each of them has been aligned with administrative areas (GADM)[2]. A user is identified by nationality, age and is described by a timeline. A user timeline represents a chronological set of reviews from its first reviews to its last reviews. This timeline allows computing intermediate properties: the time between two consecutive reviews, consecutive visited places. A review represents a note given by a user on a location at a given discrete-time domain.

Now we must clarify the key points of our study, the definition of a tourist and a tourist's stay.

A. Tourist

According to the UNWTO, a tourist is defined as: *"The activities that people undertake during their travels and their stays in places outside their usual environment for a consecutive period not exceeding one year from recreational, business and other purposes"*.

The term *activities* should be understood here in a general sense of individual occupations.

B. Tourist stay

The main objective of our study is to model and predict tourist next visited place over a given area by analyzing past and present individual tourist behavior. The model is established on a set of *sequence* created from the tourists' stays. A tourist stay is a succession of days where a tourist publishes at least one comment per day. As soon as there is a day when the tourist does not post any comment, the stay is considered interrupted. However, a tourist may not post anything for a limited period of time during the same stay. Therefore, we determine the maximum number of possible days between two comments before considering it as a new stay. According to idea of *Gossling* [26], we consider that reviews should be written within a maximum of 7 days.

The method consists of merging two stays if they satisfy the following conditions [27]:

$$\Delta B \leq \Delta S_i \; and \; \Delta B \leq \Delta S_j \; and \; L_{S_i} = F_{S_j}$$

where ΔB is break duration, ΔS_i is the i^{th} stay duration ΔS_j is the j^{th} stay duration, L_{s_i} is the last area visited during i^{th} stay and F_{S_j} is the first area visited during j^{th} stay. The Figure 1 presents an example of merge two stays **Stay-1** and **Stay-2**.

Based on a user timeline, several stays are defined. Each one is represented by a list of visited places ordered temporally in a *sequence*. Our data set is a set of temporal sequences. Each sequence is composed of places, placed in the chronological order of appearance on the tourist's timeline, thus in the order of the stay.

IV. Our approach

The purpose of this paper is to use temporal sequences to predict the next visited place of tourists.

To make the prediction, we decompose our approach into three steps:

- Learning process. This first step consists to understand tourist movements over the set of sequences and then learn past and present movements to predict a future movement. The learning process consists of inducing rules, called *Grammatical Rules*, from the sequence set. From these, a *Hidden Markov Model (HMM)* is built. An HMM is a graph with nodes and arcs weighted by probabilities.
- Prediction. The second step is based on the Hidden Markov Model. Given a node of the HMM, all the future movements are generated browsing the HMM.
- Update. The third step is to update and adapt the weighted on nodes and arcs of the Hidden Markov Model to a given set of sequences.

A. Learning process

The learning process is a set of algorithms to build a mathematical model from raw data in aims to induct knowledge, called *Grammatical Rules*. To learn from the sequences set and to induce grammatical rules, our learning process starts by modeling the sequences set to the *Frequency Prefix Tree*. Based on tree model, grammatical rules are inducted. To make predictions easier, we have structured the rules in an HMM. In what follows, we will represent each element of the sequence by a unique integer called item. In other words, an item refers to a place visited by a tourist.

1) Frequency Prefix Tree (FPT): FPT also known as $Trie$, represents a directed and rooted hierarchical tree where the nodes and the arcs are weighted.

The arcs between two nodes of FPT are multi-weighted by two attributes; 1) an $item$, represented in parenthesis and 2) the $occurrence\ frequency$ of the item in the sequences set.

The nodes of FPT have one attribute that corresponds to the number of times a sequence ends in this node. By default, it equals zero.

Construction of FPT: The process of constructing FPT is described in Algorithm 1. Initially the FPT is only composed of a root node with attribute equal zero. The construction of FPT consists of inserting each item of the sequences in the tree. The insertion process is an iterative process on each sequence as follows:

- Starting at the root of tree, the tree traversal are performed to insert each item of sequences
- The traversal compares the first item of the sequence to the arcs attribute item in the root.
  - If no arc exists or no arc has the same item attribute, a new arc to a new node is created. The attribute item of the new arc will be the item of the sequence and a frequency equals to one.

[2] GADM:https://gadm.org/index.html. 386, 735 administrative areas (country, region, department, district, city, and town).

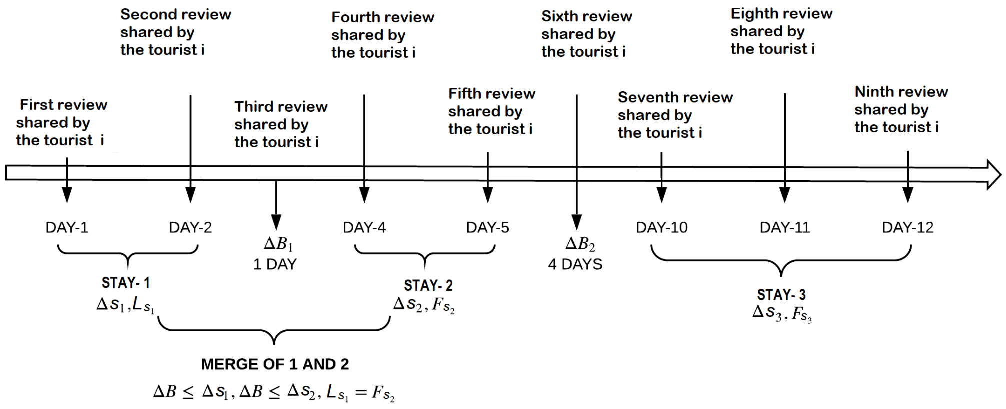

First review shared by the tourist i
Second review shared by the tourist i
Third review shared by the tourist i
Fourth review shared by the tourist i
Fifth review shared by the tourist i
Sixth review shared by the tourist i
Seventh review shared by the tourist i
Eighth review shared by the tourist i
Ninth review shared by the tourist i
DAY-1
DAY-2
ΔB_1 1 DAY
DAY-4
DAY-5
ΔB_2 4 DAYS
DAY-10
DAY-11
DAY-12
STAY- 1 $\Delta s_1, L_{s_1}$
STAY- 2 $\Delta s_2, F_{s_2}$
STAY- 3 $\Delta s_3, F_{s_3}$
MERGE OF 1 AND 2
$\Delta B \leq \Delta s_1, \Delta B \leq \Delta s_2, L_{s_1} = F_{s_2}$

Fig. 1: Example of merging two stays.

- If a arc with the same attribute item is found, its frequency is incremented by one. Then, the traversal searches the next item of the sequence in the corresponding sub-tree.
- When an item is the last of a sequence, then the attribute in the end node of the traversal is incremented by one.

Algorithm 1: Construction of a FPT.

```
Input: sequences set $A$
Output: Return FPT from sequences set $A$
Create root node of the FPT;
foreach Sequence $a$ in $A$ do
    node ← root ;
    foreach Item $l$ in sequence $a$ do
        if node has arc with item $l$ then
            arc($l$).frequency ← +1;
        else
            node.addArc($l$) // Create a arc and its end node
             arc($l$).frequency ← 1;
        end
        node ← getArc($l$) // Go the the end node of an arc
        if $a$ is end of sequence then
            node.sequence_ends ← +1
        end
    end
end
return root
```

The generated FPT cannot be used directly due to its large number of nodes. The number of nodes is at most the number of unique items power the length of the largest sequence in the sequences set. To overcome this problem, one must reduce its size using *grammatical inference (GI)*.

2) Grammatical Inference (GI): The grammatical inference is a machine learning method consists of reducing a FPT into a *Stochastic Automaton* by merging nodes. Two nodes can be merged if they are compatible *i.e.,* they valid the compatibility test.

Definition 1 (Compatibility test): Two nodes are compatible if all their arcs to their children until reaching the leaves have the same item attribute with the same *Relative Frequency* following the *Hoeffding bounds*. Parameters of Hoeffding bounds follow Mao et al. recommendation [28] and described by *Thollard* et al. [29].

Definition 2 (Relative Frequency):

> **Relative frequency of an arc** is equal to its frequency attribute divided by the sum of frequencies attributes of the outgoing arcs of its start node; *i.e.,* the outgoing arcs of its parent.
>
> **The relative frequency of a node** is equal to its attribute divided by the sum of all ingoing arc frequencies to that node. For the root, it is divided by the sum of the outgoing arc frequencies.

In the literature, there are three GI algorithms, *MDI* [30], *Alergia* [29] and *DEES* [31]. In our approach, we use the *Alergia algorithm* because of its error function. The error function computes the rate of information loss during the GI merging process.

Alergia algorithm: is a GI method, which takes as a parameter the FPT and iteratively attempts to merge all node pairs if they are compatible. With Alergia algorithm, the compatibility between two nodes depends on the compatibility between all their children until the leaves. To check if two nodes are compatible, the number of tests to compute with a given FPT having n different items and a height of h is at most n^h. Performing the compatibility test between each pair of nodes in the FPT generates an explosion of combinations.

To counter this problem, we purpose a *Relaxed Alergia algorithm*. With the Relaxed Alergia algorithm, two nodes are compatible if and only if the arcs to their children have the same relative frequency following the *Hoeffding bounds*. Despite the difference of the compatibility test between Alergia algorithm and Relaxed Alergia algorithm, they return the same result if in FTP all children of two compatible nodes have a similar relative frequency on the same item.

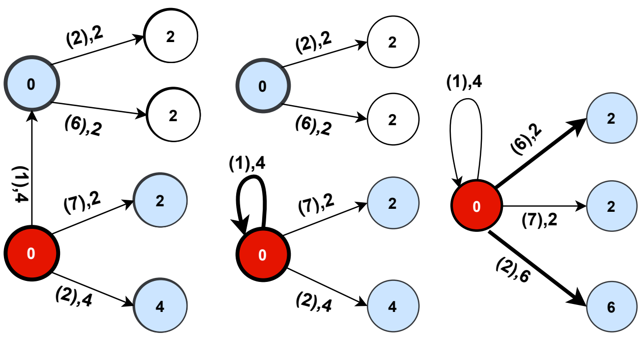
Relaxed Alergia algorithm: tries to recursively merge all the pairs of nodes, from the root to the leaves. Unlike the Alergia algorithm, it does not test the compatibility of children of these pairs of nodes. To differentiate nodes that cannot merge from those that will be tested, we define two types of nodes, RED nodes and BLUE nodes. In the beginning, only the root node of the FPT is RED, and BLUE contains the direct children of the root. The BLUE nodes are visited successively. Once visited, a BLUE node tries to merge with the RED node. If the visited node does not merge, it becomes a RED node and all its children become BLUE. This process is repeated until all FPT nodes are RED.

When a BLUE node is compatible with a RED node, the *merging process* is launched. The merging process is composed of two operations *Merge* and *Fold*.

Definition 3 (Merge operation): To merge a BLUE node into a RED node, all the ingoing arcs of BLUE node are redirected to the RED node. If the RED node already has an ingoing arc with the same item attribute as a redirected arc, then only the frequencies are added. Next, the BLUE node attribute is added to the RED node attribute.

Definition 4 (Fold operation): When a BLUE node merges into a RED node, all children of BLUE node are recursively merged into children of the RED node with the same item attribute arc. If the arc with an item attribute doesn't exist in the RED node, a new child is inserted.

The Figure 2 present an example of the merge and fold process. We consider that the RED node and the top BLUE node are compatible, see Figure 2a. The RED node merges with the BLUE node in bold. The ingoing arcs of BLUE with attributes $(1), 4$ is redirected to the RED node then make a loop, see Figure 2b . The fold operation is on the two children of BLUE, see Figure 2c. The item (2) in the arc with attributes $(2), 2$ already exists in RED as $(2), 4$, thus the arc becomes $(2), 6 = 2 + 4$. The node with attribute at 4 adds the attribute 2 thus becomes 6. The item (6) in the arc $(6), 2$ doesn't exist in RED, thus a new child with the same attributes is created.



(a) Initial graph. (b) Merge operation. (c) Fold operation.

Fig. 2: Merge and fold operations of the Relaxed Alergia algorithm.

The output of the Relaxed Alergia algorithm is a graph with frequency and item called *frequency automaton*. To make a prediction, in the literature, there are not existing algorithms or methods that allow us to analyze a frequency automaton. Thus, we will transform the frequency automaton into an *Hidden Markov Models*.

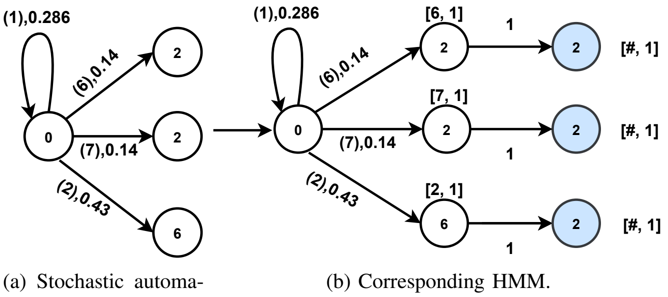
3) Hidden Markov Models: To transform a frequency automaton into a Hidden Markov Models (HMM), one has to start to change the frequency automaton into a *stochastic automaton*. This first step consists to convert the frequencies to relative frequency.

The Figure 3a provides the stochastic automaton of the frequency automaton of Figure 2c. The root node has four outgoing arcs: $(1), 4$; $(6), 2$; $(7), 2$; $(2), 6$. The sum of frequencies is equal to $4 + 2 + 2 + 6 = 14$, thus the arcs in relative frequencies are: $(1), 0.286$; $(6), 0.14$; $(7), 0.14$; $(2), 0.43$.

The process to transform a stochastic automaton into an HMM is described and proved by *Dupont* et al. [32] and *Harbrard* et al. [31]. The steps of the process are not discussed in our paper. Although we provide an explanation of the structure of an HMM.

An HMM is a graph where a node has a set of couples $[item, probability]$, refer to the $probability$ to generate the $item$ on this node. The item # refers to the end of a sequence. The root node of an HMM is characterised by an ingoing arc without a start node. Arcs possess a $probability$ to go from a node to another node.

To be able to predict, we introduce two operations on an HMM: *jump* and *observation*. The jump operation consists of going from a node to another one through an arc. After a jump, one can generate an $item$ given its $probability$. The jump and item generation form an observation. The probability to observe this item is equal to the product of the two probabilities, *i.e.,* the jump and the item's generation. The Figure 3b illustrates the transformation of the stochastic automaton in Figure 3a to an HMM.



(a) Stochastic automaton. (b) Corresponding HMM.

Fig. 3: Changing a stochastic automaton into an HMM.

The probabilities of outgoing arcs in the root stochastic automaton are : $(1), 0.286$; $(6), 0.14$; $(7), 0.14$; $(2), 0.43$. Let's observe item (1) in the HMM. This is only possible by jumping through the loop and generating an item on the root. Thus, the probability to observe (1) from the root is equal to $0.286 * 1 = 0.286$.

B. Prediction

In a HMM, the prediction process is done from a given sequence of items. Therefore, it is necessary to find where to start the prediction thanks to the *Viterbi algorithm* [33]. Given an input sequence of items, this algorithm computes the most probable sequence of observations. Hence, the prediction process occurs at the node of the last observation of this sequence. A set of all predicted observations are generated from this node. This means all jumps from the start node to another node and all generated items at these nodes are computed.

An observation is composed of couples *(suffix, probability)* where $suffix$ refers to the generated item with its $probability$ to be observed. If we want to continue to predict new places to visit from a given $suffix$, all new observations are computed. In the set of new observations, the new $suffix$es are concatenated to the old $suffix$. Since many observations can build the same new $suffix$es, their probabilities are added. The prediction process is shown in Algorithm 2.

In our context, we want to predict the next visited places of a tourist given a chronological sequence of the past and present visited places. Following the prediction process, we can recommend visiting one or many $suffix$es following the decreasing value of their probabilities.

Algorithm 2: Prediction Process.

```
Input  : An HMM with $O_{i,l}$ the probability to observe
         element $l$ on node $i$
         $P_{i,j}$ the probability to jump from node $i$ to
         node $j$ at a discrete time
         Length of suffix $L$
         Sequence $s$.
Output: A set $S_O$ of observations
$S_O \leftarrow$ empty;
root $\leftarrow$ End point of Viterbi(s);
Suffixes(root, 1, $S_O$, empty, $L$);
Function Suffixes(root, probability, set, suffix, $L$) is
if $L == 0$ then
    return;
else
    foreach Child c in root.children do
        foreach Observation l in c do
            suffix $\leftarrow$ concat(suffix,'l');
            probability $\leftarrow$ probability*$P_{root,c}$*$O_{c,l}$;
            $S_O$.add(suffix,probability);
            return Suffixes(c, probability, $S_O$, suffix,
              $L - 1$);
Sum of probabilities of same suffixes in $S_O$;
```

C. Update

The update process adapts the probabilities to jump and the probabilities to generate items for a sequence set. To update the HMM, we use the *Baum-Welch algorithm* [33], [34]. In our method, we implement this algorithm without any modification, so it isn't discussed in this paper.

V. Case study

The proposed method is implemented with data extracted from $Tripadvisor$, concerning posted reviews in $Paris$, the capital of France, from 2013 to 2019. For this case study, we consider only the 6 most visited places of Paris. In what follow, the places are renamed as: (0) represents $Arc\ de\ Triomphe$, (1) $Notre\ Dame\ Cathedral$, (2) $Eiffel\ Tower$, (3) $Luxembourg's\ Garden$, (4) $Quai\ de\ Seine$ and (5) $Louvre's\ Museum.$

After data processing, our data set is composed of $11'471$ sequences of at least 2 items. Indeed, sequences of only 1 item are removed because they do not show the behavior of a tourist. Based on the sequences set, the generated FPT has 599 nodes and 556 of them have a non-null attribute. The complete source code and detail of results and figures are provided in a GitHub folder[3].

1) Validation of the Relaxed Alergia algorithm: First of all, we check if in FTP all arcs with the same item attribute have a similar relative frequency. The results show all the children with the same item from a given sequence have relative frequencies with an absolute variation of at most 8%. The Relaxed Alergia algorithm can be used on the sequences set. Therefore, it exists three anomalies: item (5) at the first position and second position, and item (3) at the first position. Due to the anomalies, the HMM model will provide some probabilities on the sequences containing those items with a more significant error than the other sequences comparing to the sequences set. Applying the Relaxed Alergia algorithm, the generated HMM has 37 nodes with 18 end nodes. To compensate the three anomalies, the HMM will be updated with the sequences set many times before beginning the predictions. To prove the efficiency of the update process, we will compare the HMM before and after updating in what follows.

2) HMM's validation: To validate our HMM, we verify if it's a reflection of the sequences set. Let's compare the probabilities of a sequence in the sequences set to the probability of the same sequence observed in the HMM. For a given sequence in the sequences set, we use the Baum-*Welch algorithm* [33], [34] to compute its probability to be observed in the HMM. To compare the probability R_s of a sequence s in the sequences set to its probability P_s to be observed in the HMM, we use the *Absolute Percent Error* (APE). APE is defined as follows: APE $= \left|\frac{R_s - P_s}{R_s}\right|$.

The Figure 4 shows the APE between the probabilities from the sequences set and the observed probabilities in the HMM. In the x-axis, each digit represents an item. For example 05 means the sequence $\{0, 5\}$, *i.e., Arc de Triomphe, Louvre's Museum*. The APE values are between 3% to 66% with its mean (MAPE) equals to 20.8%. In consequence, various observed sequences are not well represented in the HMM. The

[3] https://github.com/ TheoDemessance/Hidden-Markov-Model-for-Tourism

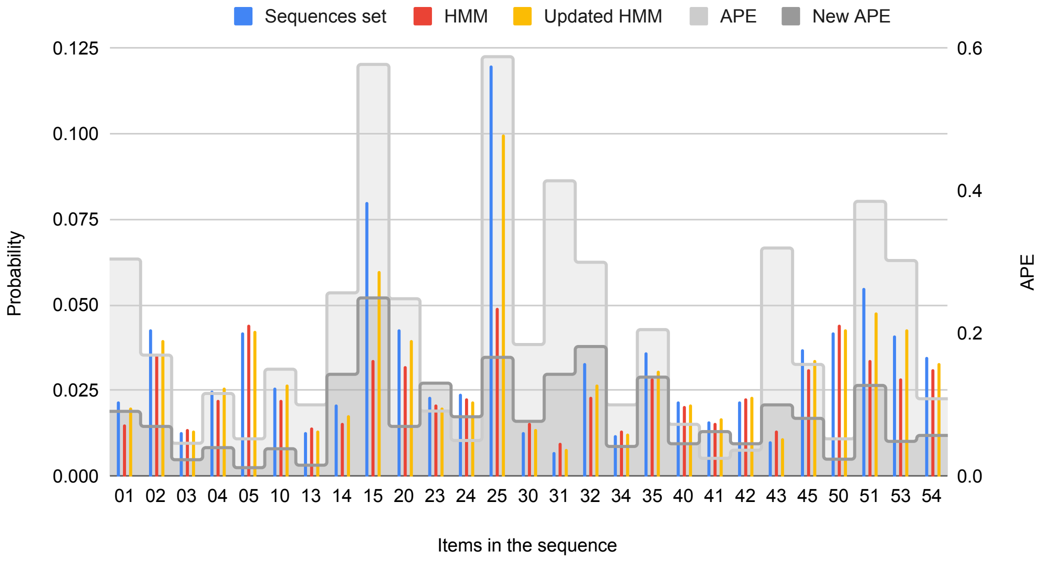


Fig. 4: HMM's validation.

item (5) in the first and second position and the item (3) in the first position produce the largest APE values as expected.

To provide a more accurate observation of the sequence containing those items, we update the HMM with the sequences set until the MAPE is strictly inferior to 10%. The figure 4 also presents the probabilities of observations of the update HMM with the new APE on each sequence. The MAPE is equal to 8.9%, APE is between 1.2% to 25%. The APE significantly decreases for all sequences and they are more restricted around the MAPE. These results demonstrate how the HMM adapts itself to better reflect the sequences set.

3) HMM's prediction: Presently, we can predict in the HMM the future behavior of a tourist-based on a sequence of places visited. The figure 5 presents the probability to observe suffix in the HMM, in the updated HMM and their respective APE. The sequence is composed of two sub-sequences A-B where A is the sequence of places visited and B is the predicted suffix. The MAPE between the sequences set and the HMM is equal to 36%, with APE values from 6% to 98%. In the updated HMM, the MAPE is equal to 9.1%, with APE value from 2.5% to 28%. The Relaxed Alergia algorithm has built an HMM that adapts to all sequences. The update process improves modeling and forecasting results.

The proposed model provides relevant and close to reality predictions of tourist behavior. Models can be constructed from a set of restricted sequences based on nationality, age, gender, etc. to constitute a model that fits a set of tourists. To make recommendations for a tourist, a set of suffixes is built from its visited places. One can recommend the suffix with the largest probability or a suffix containing a wanted place with the largest probability.

VI. Conclusion

The main idea of this study is to determine a decision tool representing the behaviors of a set of tourists. Our proposed method keeps all the possible behaviors. When a behavior is over-represented or under-represented, our method adapts the decision tool to better fit the reality.

Behaviors may vary according to the country of origin, age, gender, social class of a set of tourists. Since the methods presented can predict the behaviors of a specific group of tourists, we will compare the decision tool of various sets of tourists in future work. We aim to provide different decision tools according to the profile of tourists. Moreover, when a tourist needs a recommendation, his best profile is determined to best meet his expectations.

References

[1] C. Cooper and C. M. Hall, *Contemporary tourism*. Routledge, 2007.

[2] L. T. S. W. . L. L. Jingjing Li, Lizhi Xu, "Big data in tourism research: A literature review," *Tourism Management*, vol. 68, pp. 301–323, 2018.

[3] R. March and A. G. Woodside, *Tourism behaviour: Travellers' decisions and actions*. Cabi, 2005.

[4] R. M. Chalfen, "Photograph's role in tourism: Some unexplored relationships," *Annals of tourism research*, vol. 6, no. 4, pp. 435–447, 1979.

[5] G. Chareyron, S. Cousin, J. Da-Rugna, and D. Gabay, "Touriscope: map the world using geolocated photographies," in *IGU meeting, Geography of Tourism, Leisure and Global Change*, 2009.

[6] D. Knoke and S. Yang, *Social network analysis*. Sage Publications, 2019, vol. 154.

[7] G. Kitagawa, *Introduction to time series modeling*. CRC press, 2010.

[8] P. R. da Silva Soares and R. B. C. Prudêncio, "Time series based link prediction," in *The 2012 international joint conference on neural networks (IJCNN)*. IEEE, 2012, pp. 1–7.

[9] B. Askari, T. Le Quy, and E. Ntoutsi, "Taxi demand prediction using an lstm-based deep sequence model and points of interest," in *2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*, 2020, pp. 1719–1724.

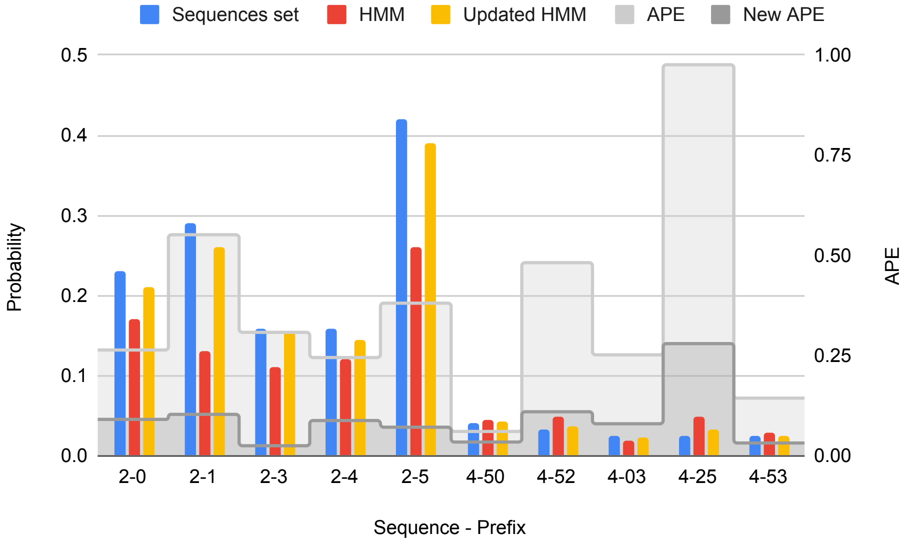


Fig. 5: HMM's predictions.

[10] D. Zhan, S. Yi, and D. Jiang, "Small-scale demographic sequences projection based on time series clustering and lstm-rnn," in *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, 2018, pp. 803–809.

[11] C.-H. Wang, "Predicting tourism demand using fuzzy time series and hybrid grey theory," *Tourism management*, vol. 25, pp. 367–374, 2004.

[12] K. Zhang, Y. Chen, and C. Li, "Discovering the tourists' behaviors and perceptions in a tourism destination by analyzing photos' visual content with a computer deep learning model: The case of beijing," *Tourism Management*, vol. 75, pp. 595–608, 2019.

[13] Y. Bao, Z. Huang, L. Li, Y. Wang, and Y. Liu, "A bilstm-cnn model for predicting users' next locations based on geotagged social media," *International Journal of Geographical Information Science*, pp. 1–22, 2020.

[14] A. Majid, L. Chen, H. T. Mirza, I. Hussain, and G. Chen, "A system for mining interesting tourist locations and travel sequences from public geo-tagged photos," *Data & Knowledge Engineering*, vol. 95, pp. 66–86, 2015.

[15] I. Memon, L. Chen, A. Majid, M. Lv, I. Hussain, and G. Chen, "Travel recommendation using geo-tagged photos in social media for tourist," *Wireless Personal Communications*, vol. 80, no. 4, pp. 1347–1362, 2015.

[16] L. Wan, Y. Hong, Z. Huang, X. Peng, and R. Li, "A hybrid ensemble learning method for tourist route recommendations based on geo-tagged social networks," *International Journal of Geographical Information Science*, vol. 32, pp. 2225–2246, 2018.

[17] R. A. Howard, "Dynamic programming and markov processes." 1960.

[18] R. L. . B. H. Y. Huy Quan Vu, Gang Li, "Exploring the travel behaviors of inbound tourists to hong kong using geotagged photos," *Tourism Management*, vol. 46, pp. 222–232, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.tourman.2014.07.003

[19] I. Saadi, A. Mustafa, J. Teller, and M. Cools, "Forecasting travel behavior using markov chains-based approaches," *Transportation Research Part C: Emerging Technologies*, vol. 69, pp. 402–417, 08 2016.

[20] B. Wang, Y. Hu, G. Shou, and Z. Guo, "Trajectory prediction in campus based on markov chains," in *BigCom*, 2016.

[21] S. Ahmad, I. Ullah, F. Mehmood, M. Fayaz, and D. Kim, "A stochastic approach towards travel route optimization and recommendation based on users constraints using markov chain," *IEEE Access*, vol. 7, pp. 90 760–90 776, 2019.

[22] J. C. Xia, P. Zeephongsekul, and D. Packer, "Spatial and temporal modelling of tourist movements using semi-markov processes," *Tourism Management*, vol. 32, pp. 844–851, 2011.

[23] M. Ramezani and N. Geroliminis, "On the estimation of arterial route travel time distribution with markov chains," *Transportation Research Part B: Methodological*, vol. 46, no. 10, pp. 1576 – 1590, 2012.

[24] J. Yeon, L. Elefteriadou, and S. Lawphongpanich, "Travel time estimation on a freeway using discrete time markov chains," *Transportation Research Part B: Methodological*, vol. 42, pp. 325–338, 2008.

[25] M. L. Hazelton, S. Lee, and J. W. Polak, "Stationary states in stochastic process models of traffic assignment: a markov chain monte carlo approach," 1996.

[26] S. Gössling, D. Scott, and C. M. Hall, "Global trends in length of stay: implications for destination management and climate change," *Journal of Sustainable Tourism*, vol. 26, pp. 2087–2101, 2018.

[27] L. B. Baccar, S. Djebali, and G. Guérard, "Tourist's tour prediction by sequential data mining approach," in *International Conference on Advanced Data Mining and Applications*. Springer, 2019, pp. 681–695.

[28] H. Mao, Y. Chen, M. Jaeger, T. D. Nielsen, K. G. Larsen, and B. Nielsen, "Learning deterministic probabilistic automata from a model checking perspective," *Machine Learning*, vol. 105, no. 2, pp. 255–299, 2016.

[29] F. Thollard, P. Dupont, C. De La Higuera *et al.*, "Probabilistic dfa inference using kullback-leibler divergence and minimality," in *ICML*, 2000, pp. 975–982.

[30] R. C. Carrasco and J. Oncina, "Learning stochastic regular grammars by means of a state merging method," in *International Colloquium on Grammatical Inference*. Springer, 1994, pp. 139–152.

[31] A. Habrard, F. Denis, and Y. Esposito, "Using pseudo-stochastic rational languages in probabilistic grammatical inference," in *International Colloquium on Grammatical Inference*. Springer, 2006, pp. 112–124.

[32] P. Dupont, F. Denis, and Y. Esposito, "Links between probabilistic automata and hidden markov models: probability distributions, learning models and induction algorithms," *Pattern recognition*, vol. 38, pp. 1349–1371, 2005.

[33] A. Kriouile, J.-F. Mari, and J.-P. Haon, "Some improvements in speech recognition algorithms based on hmm," in *International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 1990, pp. 545–548.

[34] P. M. Baggenstoss, "A modified baum-welch algorithm for hidden markov models with multiple observation spaces," *IEEE Transactions on speech and audio processing*, vol. 9, pp. 411–416, 2001.