Traveling Salesman in Reverse: Conditional Markov Entropy for Trajectory Segmentation

Mohamed Kafsi
EPFL, Switzerland
mohamed.kafsi@epfl.ch

Matthias Grossglauser
EPFL, Switzerland
matthias.grossglauser@epfl.ch

Patrick Thiran
EPFL, Switzerland
patrick.thiran@epfl.ch

***Abstract*—We are interested in inferring the set of waypoints (or intermediate destinations) of a mobility trajectory in the absence of timing information. We find that, by mining a dataset of real mobility traces, computing the entropy of conditional Markov trajectory enables us to uncover waypoints, even though no timing information nor absolute geographic location is provided. We build on this observation and design an efficient algorithm for trajectory segmentation. Our empirical evaluation demonstrates that the entropy-based heuristic used by our segmentation algorithm outperforms alternative approaches as it is 43% more accurate than a geometric approach and 20% more accurate than path-stretch based approach. We further explore the link between trajectory entropy, mobility predictability and the nature of intermediate locations using a route choice model on real city maps.**

I. Introduction

Mobility is one of the most informative and valuable types of human behavioral data. It is central to many new classes of online services, from navigation tools to new models of social interaction. Mobility patterns also correlate with many behavioral and demographic traits and personal preferences [4], [14]. This implies, on the one hand, that mobility is potentially very valuable (e.g., for targeted advertisement), but on the other hand, highly privacy-sensitive.

Human mobility usually serves the purpose of reaching a small, discrete set of locations, which we refer to as *waypoints* in this paper (e.g., workplace, shop, restaurant, movie theater)[1]. All the other locations in a trajectory over, say, a day, are just intermediate points en route to the next waypoint (e.g., train station, airport, car, walking). We usually want to arrive at the next waypoint as efficiently and quickly as possible, to then spend time at the waypoint. A waypoint reveals therefore much about individuals, more than intermediate locations do, as people spend time at locations that play a central role in their lives. In this paper, we investigate to which extent we can uncover a user's waypoints from his full trajectory, from minimal information.

In the absence of an explicit signal from the user, the time spent at a location best indicates whether this intermediate location is actually a waypoint [17]. However, time information might be missing or very sparse, which makes waypoint inference particularly challenging. Such a situation occurs, for example, when privacy-preserved trajectories with missing time information are released to the public. We can also think of the scenario for which we have sparse trajectories because a user reveals only a few locations along her trajectory. These trajectories typically arise from a sequence of check-ins (e.g., Foursquare or Twitter), geo-tagged photos (e.g., Flickr), credit card transactions, and snapshots of vehicles captured by surveillance cameras. Hence, even if some approaches [1] enable us to infer the unobserved locations, the successive observed check-ins are so distant that we are unable to accurately infer the duration the user stays at each location. The results of our work show that our segmentation approach enables data miners to uncover important intermediate locations along a trajectory, *only from spatial information*.

We address this question within a graph abstraction of the world. Instead of a trajectory through $\mathcal{R}^2$ (or even $\mathcal{R}^3$), we discretize the user's world to obtain a map that we represent as a mobility graph G. A vertex of this graph represents a branch point where the user takes the decision about where to move next (e.g., an intersection), and an edge represents a direct physical path between two vertices (e.g, a road segment between two intersections). The advantage of this model over full (geographic) trajectories is that it essentially encodes the space of possible user decisions, but abstracts away any finer but irrelevant details of the mobility process. In this model, a *user trajectory* is simply a walk on the graph G from a starting vertex s to a destination vertex d.

In this paper, we are interested in inferring the set of waypoints given a trajectory. We formulate this as a classification problem for which every vertex on the trajectory as either a waypoint or an intermediate point. This uses a statistic that captures the evolution of the uncertainty about the trajectory given only the waypoints. Specifically, we assume a Markovian mobility model on the graph, and we compute the conditional trajectory entropy given the candidate waypoints [6], [8]. This conditional entropy captures the degree to which knowledge of only the waypoints predicts the full trajectory.

We make two main contributions in this paper. First, we find an empirical connection between the class of a vertex on a trajectory (waypoint or intermediate point) and the conditional Markov entropy of that vertex. More specifically, given a trajectory from s to d that passes through some vertex u, we compare H_{sd}, the unconditional uncertainty over the family of all trajectories from s to d, to the conditional entropy $H_{sd|u}$ of all trajectories $s \rightarrow d$ that pass through u. We show that, through an extensive analysis of real mobility traces, waypoints are those with a high ratio $H_{sd|u}/H_{sd}$. We provide the intuition behind this observation, and establish a connection with random walk hitting times in the special case

[1]With some exceptions, e.g., leisure travel, where sometimes "it's the journey, not the destination", or sports activities.

of a regular graph. It is remarkable that waypoints can be found from trajectories in the mobility graph G alone, i.e., in the absence of any timing information and any absolute geographic locations.

Our second contribution builds on this insight. We develop a segmentation algorithm that infers the likely waypoints of a trajectory, based on conditional trajectory entropy. The entropy is computed with respect to a Markovian mobility model, which in a practical implementation would be trained from a database of mobility traces. We evaluate this algorithm over a large dataset of real mobility traces; we show that the points corresponding to high conditional entropy tend to be those with high residence time, which is much more likely for a waypoint than an intermediate point. The entropy based heuristic used by our algorithm outperforms alternative approaches: it is 43% more accurate than a geometric approach and 20% more accurate than path stretch based approach. Moreover, it is computationally efficient at online segmentation of trajectories, given that offline computations are performed only once.

The remainder of the paper is structured as follows. In Section II, we describe the trajectory entropy model. In Section III, we mine a dataset of nearly 18,000 GPS trajectories in order to explore the link between trajectory entropy and waypoints. We build on these findings and present an algorithm for trajectory segmentation in Section IV. We conduct, in Section V, empirical experiments in order to evaluate the performance of our approach at segmenting trajectories and compare it to the performance of alternative approaches. In Section VI, we present two empirical experiments that enable us to gain more insight into the link between the evolution of trajectory entropy, mobility predictability and the nature of intermediate location. In Section VII, we present the related work and we conclude in Section VIII.

II. Trajectory Entropy Model

Human mobility is governed by both subjective principles (e.g., personal preferences and habits, social relationships, environment perception) and objective principles (roads and geographic constraints). For an observer who knows only partially the subject's motivation and history, the mobility process remains ambiguous and is therefore suitably modeled as a stochastic process.

In our work, we model the mobility of a user as a random walk on a finite graph $G(V, E)$. A vertex of this graph represents a decision point where the user can choose where to move next (a semantic place such as home or work), and an edge represents a direct physical path between two vertices (work home routine). The advantage of this model, over a continuous geographic model, is that it essentially represents the space of possible user decisions, but abstracts away any finer but irrelevant details of the mobility process. Moreover, it has the advantage of being general enough to be representative of most scenarios. A random walk on graph G is equivalent to a finite state Markov chain (MC) $\{X_i\}$ characterized by the matrix of transition probabilities P. The order of the MC determines the memory of the process that it models: For a first order MC, each state (vertex) represents a decision point, whereas for higher order MCs, a state represents a sequence of decision points. Naturally, a state can encode more information than a sequence of points because we are able to add features such as place semantics and time. For the rest of this paper, we assume, if no stated otherwise, that a vertex of the graph G and the corresponding state of the MC represent a location.

In this model, the sequence of locations a user visits as a random trajectory T_{sd} of a MC. This trajectory T_{sd}, as defined by Ekroot and Cover [6], is a path with initial state s, final state d and no intermediate state d, i.e., the trajectory is terminated as soon as it reaches state d. Using the Markov property, the probability of a realization $t_{sd} = sx_2...x_kd$ given that $X_1 = s$ is

$$p(t_{sd}) = P_{sx_2}P_{x_2x_3}\dots P_{x_kd}.$$

Let $\mathcal{T}_{sd}$ be the set of all trajectories that start at state s and end as soon as they reach state d. As the MC defined by the matrix P is finite and irreducible, we have

$$\sum_{t_{sd}\in\mathcal{T}_{sd}} p(t_{sd}) = 1 \qquad \text{for all } s, d.$$

So T_{sd} is a discrete random variable that has as support the set $\mathcal{T}_{sd}$, with the probability mass function $p(t_{sd})$.

The entropy of a random variable quantifies the uncertainty or expected surprise of its realization. It is therefore a natural choice when we are interested in quantifying the predictability of a random variable. The entropy of a random variable is maximized when all its realizations are equiprobable —high uncertainty so low predictability —while it is minimized when the random variable is deterministic —no uncertainty so maximum predictability —. Consequently, the uncertainty of the user's trajectory, between the states s and d, is captured by

$$H_{sd} \equiv H(T_{sd}) = -\sum_{t_{sd}\in\mathcal{T}_{sd}} p(t_{sd}) \log p(t_{sd}).$$

Let H denote the matrix of trajectories entropy where $H_{ij} = H(T_{ij})$. Ekroot and Cover [6] provide a general closed-form expression for the matrix H for the case of an irreducible, aperiodic, and finite state MC.

Furthermore, if the user provides location updates along his trajectory, the predictability of his mobility evolves and is now captured by the entropy of his trajectory, conditional on the intermediate locations revealed. We represent location updates as a sequence of intermediate states $\boldsymbol{u} = u_1u_2\dots u_l$, and the entropy of the user's trajectory becomes

$$\begin{aligned} H_{sd|\boldsymbol{u}} &\equiv H(T_{sd}|T_{sd}\in\mathcal{T}_{sd}^{\boldsymbol{u}}) \\ &= -\sum_{t_{sd}\in\mathcal{T}_{sd}^{\boldsymbol{u}}} p(t_{sd}|T_{sd}\in\mathcal{T}_{sd}^{\boldsymbol{u}}) \log p(t_{sd}|T_{sd}\in\mathcal{T}_{sd}^{\boldsymbol{u}}), \end{aligned} \tag{1}$$

where $\mathcal{T}_{sd}^{\boldsymbol{u}}$ is the set of all trajectories in $\mathcal{T}_{sd}$ that exhibit the intermediate states $\boldsymbol{u}$

$$\mathcal{T}_{sd}^{\boldsymbol{u}} = \{t_{sd}\in\mathcal{T}_{sd} : t_{sd} = s\dots u_1\dots u_2\dots u_l\dots d\}.$$

Computing the entropy $H_{sd|u}$ is challenging. Even the costly approach of computing all the terms of the sum (1) is not always possible because the set $\mathcal{T}_{sd}^{u}$ has an infinite number of members in the case where, after removing state d, the transition graph of the MC is not a DAG. Kafsi et al. [8] provide a general closed-form expression to compute the entropy of Markov trajectories conditional on multiple intermediate states.

Their approach is based on a transformation of the original MC into a MC that exhibits the desired conditional distribution $p(T_{sd}|T_{sd} \in \mathcal{T}_{sd}^{\boldsymbol{u}})$. It is important to emphasize that the entropy $H_{sd|u}$ is not the entropy of the random variable T_{sd} given another random variable, but the entropy of T_{sd} conditional on the realization of a dependent random variable. This implies that the trajectory entropy does not necessarily decrease as we condition on intermediate locations: the conditional entropy $H_{sd|u}$ might be larger the *unconditional* trajectory entropy H_{sd}.

In this work, we explore the link between the evolution of conditional trajectory entropy and waypoints. Our intuition is simple: An increase of the trajectory entropy due to conditioning on a intermediate location is an indicator that the posterior distribution of trajectories $p(T_{sd}|T_{sd} \in \mathcal{T}_{sd}^{\boldsymbol{u}})$ is very different from the prior distribution $p(T_{sd})$. In other words, revealing this evidence contradicts our prior belief about the family of trajectories. More precisely, it contradicts the assumption of location d being the destination of the trajectory: the intermediate location that most increases trajectory entropy is the most likely to be an intermediate destination (waypoint) of the user whose final destination is d. This intuition is confirmed by both theoretical and empirical results. From the theoretical side, we prove (detailed proof in Appendix A) that, for a regular graph, the trajectory entropy H_{sd} is proportional to the hitting time S_{sd} .i.e., the expected length of the trajectory T_{sd}. Consequently, the ratio $H_{sd|u}/H_{sd}$ captures the stretch of the length of trajectory T_{sd} when we condition on going through state u. Naturally, for the case of non-regular graph, the trajectory entropy captures properties that are richer than the hitting time. We confirm this emprically in Section V.

In the next section, we explore this direction further and show that there is, indeed, an empirical connection between high conditional entropy and waypoints.

III. Trajectory conditional entropy and waypoints

In this section, we explore the link between trajectory conditional entropy and waypoints: We show empirically, using a dataset of around $18,000$ trajectories, that intermediate locations that increase trajectory entropy are more likely to be waypoints where users spend a significant amount of time.

A. Dataset and Mobility Model

Geolife dataset: Exploring the link between waypoints and trajectory entropy necessitates a dataset that associates trajectories with time information. The Geolife project [17] consisted in collecting the mobility traces of 182 users over a five-year period. The collected dataset contains around $18,000$ trajectories (more than $1,300,000$ km) mainly located in China. A trajectory of this dataset is represented as a sequence of time-stamped points described by their latitude, longitude and altitude. The sampling rates vary between users but remain very high in general: 91.5% of the trajectories are logged in a dense representation, e.g., every $1-5$ seconds. These trajectories present the advantage of being very diverse because they are associated with different activities and transportation modes. In fact, according to Zheng et al. [17] who collected these traces, some trajectories are associated with home-to-work routines, whereas some others are associated with shopping and sightseeing.

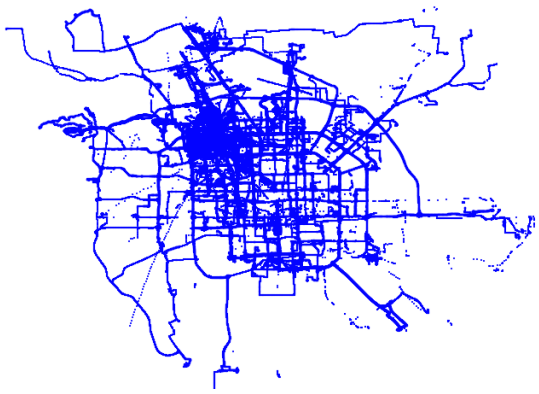

Fig. 1. The majority of the GPS trajectories in the Geolife dataset are within the city of Beijing, China.

Pre-processing: We process the data as follows: First, we discretize the GPS records by dividing the surface of the globe into identical areas (squares whose side lengths are 1km). A square basically represents the set of locations enclosed within the area it covers. Then we represent each trajectory in the dataset as the sequence of areas visited and the associated *residence* time in each area, i.e., the total time spent by the user in this area. We will use the information about residence time in order to detect waypoints. In Figure 2, we show the empirical distribution of trajectory length given as the number of areas covered. The trajectory lengths range from very short trajectories (1 or 2 locations) that correspond to short urban trips to very long trajectories that correspond to inter-city trips. In fact, in the raw dataset, 36% of the trajectories span a distance that is less than 5 km, whereas 5% of the trajectories span a distance superior to 100 km. Moreover, as the majority of the trajectories are geographically within the city of Beijing [17], we choose to focus on the data produced in this capital.

Constructing the mobility MC: After the pre-processing phase, we construct a weighted graph $G(V, E)$ whose vertices represent geographical areas and edges represent *direct* transitions between areas. As we are interested in actual transitions between areas, we exclude jumps that are due to a loss of GPS signal and also exclude self-transitions that reflect only multiple location samples within the same area. As a result, the weight of an edge $(i, j) \in E$ is equal to the number of direct transitions from area i to area j. We infer, using a maximum likelihood estimator, the first order MC that has generated the observed data. The training set $\mathcal{T}_{\text{train}}$ contains the trajectories of all users and the MC obtained is therefore a low order mobility model that captures the population mobility pattern. Note that we choose a low order MC because the training data available does not allow for training a higher order MC, similar to the one we present in Section VI-A. In fact, the number of samples needed to train a MC increases exponentially with the order of the MC. Having samples fewer than the number needed for a correct training of a high order MC leads to severe over-fitting. Having constructed the MC that captures the patterns of

population mobility, we can analyze the link between trajectory entropy and waypoints.

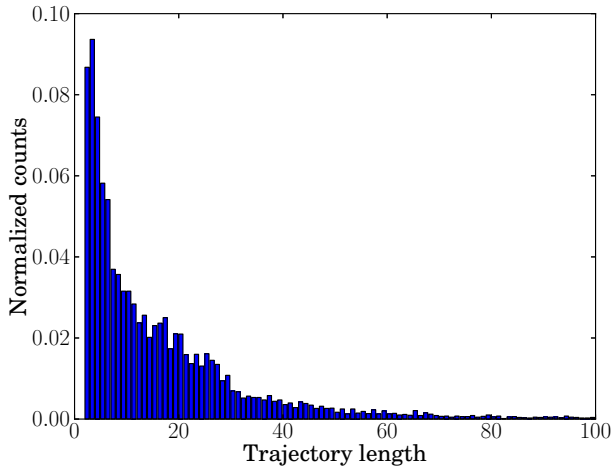

Fig. 2. Histogram of the length of Geolife trajectories after pre-processing. The trajectory lengths range from very short trajectories that correspond to short urban trips to very long trajectories that correspond to inter-city trips.

B. Waypoints Increase Trajectory Entropy

As discussed in Section II, revealing some intermediate locations might increase trajectory entropy, because it implies a conditioning that drastically changes the distribution of trajectories. A plausible explanation for such a Bayesian surprise —the posterior distribution of trajectory is very different from the prior distribution—is that the location revealed is not simply an intermediate location: it is a waypoint in itself. In order to explore this direction, we conduct the following experiment: we observe the mobility of a user whose trajectory t_{sd} starts at location s and ends at location d. Suppose that this user has a waypoint u along his trajectory. As this waypoint is more important than the intermediate locations that lead to it, the time the user would spend at this waypoint u is presumably larger than the average time spent at the other intermediate locations. If the locations that increase the entropy are more likely to be waypoints, the average time spent by the users at these locations —a proxy for their importance —should be larger than the average time spent at other intermediate locations. If our hypothesis is true, we would be able to quantify more accurately the importance of a location, even when the location records are not associated with time steps; observing the evolution of trajectory conditional entropy evolution would enable us to detect these important waypoints.

In order to test this hypothesis, we conduct the following experiment: we associate with each trajectory t_{sd} the set of locations $\mathcal{U}_\alpha(t_{sd})$ that is defined as

$$\mathcal{U}_\alpha(t_{sd}) = \{u \in t_{sd} | H_{sd|u} > \alpha H_{sd}\}. \quad (2)$$

This set contains the intermediate locations u whose conditional entropy $H_{sd|u}$ is larger than αH_{sd}. For $\alpha = 0$, the set $\mathcal{U}_\alpha(t_{sd})$ is equal to the trajectory t_{sd}, whereas for $\alpha = 1$, the set $\mathcal{U}_\alpha(t_{sd})$ is the set of locations in t_{sd} that increase the trajectory entropy.

We introduce the continuous random variable $R(u)$ that represents the residence time at location u. We are interested in analyzing the evolution of the expected residence time at a location as a function of the change of trajectory entropy if we reveal this location. More formally, we analyze the evolution of

$$\mu_\alpha = \mathbb{E}[R(u) | u \in \mathcal{U}_\alpha(T_{sd})] \quad (3)$$

as we increase the value of the parameter α.

We approximate the quantity (3) by the empirical average

$$\hat{\mu}_\alpha = \frac{\sum_{t_{sd} \in \mathcal{T}_{\text{train}}} \sum_{u \in t_{sd}} r(u, t_{sd}) \mathbb{1}_{u \in \mathcal{U}_\alpha(t_{sd})}}{\sum_{t_{sd} \in \mathcal{T}_{\text{train}}} \sum_{u \in t_{sd}} \mathbb{1}_{u \in \mathcal{U}_\alpha(t_{sd})}}, \quad (4)$$

where $r(u, t_{sd})$ is the residence time at location u along the trajectory t_{sd}.

Figure 3 illustrates the evolution of $\hat{\mu}_\alpha$ as a function of α. As we increase the value of α, we become increasingly

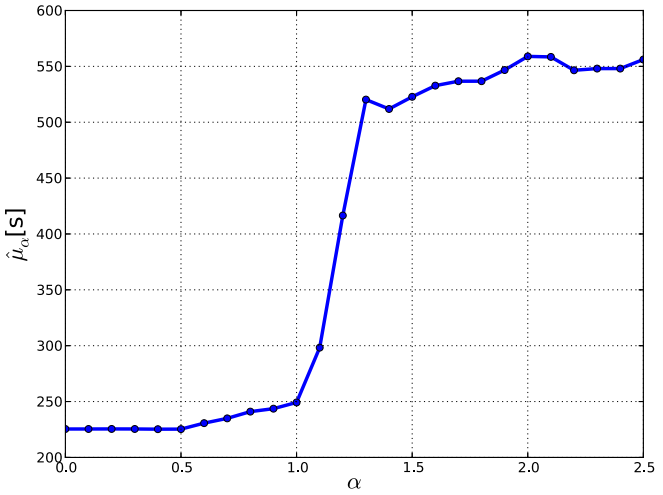

Fig. 3. Evolution of the average time spent at a location $\hat{\mu}_\alpha$ as a function of the entropy ratio α. We observe a sharp transition in the value of $\hat{\mu}_\alpha$ as we consider only the points that increase trajectory entropy. This strongly supports our hypothesis stating that locations that increase trajectory entropy are more likely to be waypoints.

restrictive and consider only the intermediate locations that satisfy the inequality $H_{sd|u} > \alpha H_{sd}$. We notice that the average time $\hat{\mu}_\alpha$ increases with α. More importantly, we observe a sharp transition in the value of $\hat{\mu}_\alpha$ as soon as we consider only the points that increase the trajectory entropy. In fact, the average time a user spends at a location is less than 4 minutes, whereas the average time spent at locations that increase the trajectory entropy ($H_{sd|u} > 1.3\, H_{sd}$) is longer than 8 minutes.

In order to explore further this direction, we compare the distribution of the residence time $R(u)$ at locations that decrease the entropy ($H_{sd|u} \leq H_{sd}$) with the distribution of residence time at locations that increase it ($H_{sd|u} > H_{sd}$). Figure 4 shows two CCDFs (complementary cumulative distribution functions) of the residence time $R(u)$ for both situations. We clearly observe that the CCDFs have the same evolution for low values of residence time but diverge starting at $r = 8$ minutes. Above this value, there will more likely be a large residence time in a location that increases the entropy

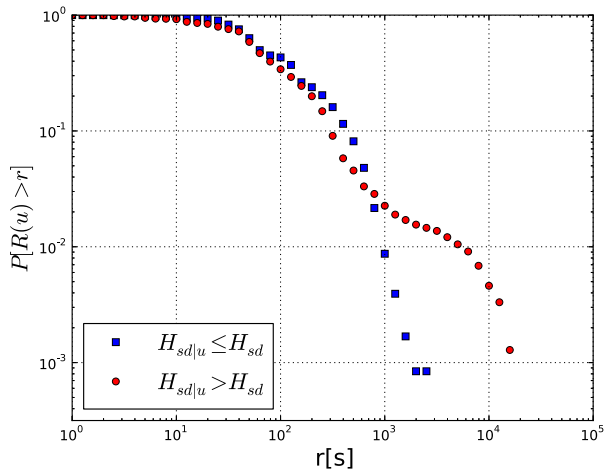

Fig. 4. Log-log plot of the complementary cumulative distribution function (CCDF) of the residence time R.

rather than in a location that decreases it. For example, it is 10 times more likely that a user spends more than 30 minutes at a location that increases the entropy rather than at a location that decreases it. Taken together, our results strongly support our hypothesis stating that locations that increase trajectory entropy are more likely to be waypoints where a user will spend more time. Furthermore, these results imply that, using a low order mobility model that is based on the patterns of population mobility, we are able —with no time information —to segment individual trajectories by detecting waypoints.

In the next section, we build on these findings and propose an algorithm that is able to automatically segment a given trajectory by using a heuristic based on trajectory entropy.

IV. Algorithm

We propose here a recursive algorithm that segments a trajectory by finding intermediate locations that increase the conditional entropy.

Algorithm for trajectory segmentation: We show in Algorithm 1 the pseudo-code of our segmentation algorithm. The input of the algorithm is a trajectory t_{sd}, a MC transition probability matrix P and a sensitivity parameter $\alpha > 0$. The algorithm recursively segments the trajectory t_{sd} by finding the intermediate point u that maximizes the conditional entropy $H_{sd|u}$ (line 21). If this conditional entropy $H_{sd|u}$ is larger than αH_{sd}, the point u is added to the sequence of waypoints U. Note that the sensitivity parameter α controls to which extent the segmentation process is conservative: The higher alpha is, the more selective we are at declaring a point as a waypoint. If a waypoint u is chosen, the segmentation algorithm continues by applying the same procedure to the two sub-trajectories t_{su} and t_{ud}.

Our algorithm bears similarity with the Ramer-Douglas-Peuker algorithm [11] that is used for polygonal approximation of plane curves. The conditional trajectory entropy in our algorithm is analogous to the Euclidean distance between the original curve and the simplified curve in [11].

Algorithm 1: Trajectory segmentation

Input: trajectory $traj$, transition probabilities matrix P, sensitivity α
Output: indices of waypoints U

```
1  begin
2      U ⟵ ∅          // global variable
3      if len(traj) > 2 then
4          segment(traj, 0, len(traj) − 1)
5      end
6      return U
7  end
8  Function segment(traj, i, j)
9      k ⟵ partition(traj, i, j)
10     if k ≥ 0 then
11         U ⟵ U ∪ {k}
12         if i + 1 < k then
13             segment(traj, i, k)
14         end
15         if k + 1 < j then
16             segment(traj, k, j)
17         end
18     end
19 Function partition(traj, i, j)
20     s ← traj[i], d ← traj[j]
21     k ⟵ arg max_{i<k<j} H_{sd|traj[k]}      // finding
       the element that maximizes
       conditional entropy
22     u ⟵ traj[k]
23     if H_{sd|u} > αH_{sd} then
24         return k
25     else
26         return −1
27     end
```

Complexity: We study the average case complexity of our algorithm for a N states MC and a trajectory of length l. The expected number of nested calls is upped bounded by $\log l$: in the most balanced situation, we divide the trajectory into two sub-trajectories with approximately the same length. Typically, the number of nested calls is much lower than $\log l$ because the number of waypoints in a trajectory of length l is much lower than l. For each call, we compute the conditional entropy for $\mathcal{O}(l)$ candidates; this necessitates the computation of the fundamental matrix associated with the Markov chain.

A naive computation of the fundamental matrix has a $\mathcal{O}(N^3)$ complexity because it necessitates the inversion of a matrix of size $\mathcal{O}(N)$. However, for a given MC, we can pre-compute offline the conditional entropies $H_{sd|u}$ and then use these results in order to segment all the trajectories on this MC. In such a situation, the expected time complexity of segmenting a trajectory of length l is $\mathcal{O}(l \log l)$, which allows for a very efficient online segmentation of trajectories.

Another interesting direction we explore is the approximation of the conditional entropy $H_{sd|u}$ by the sum of entropies $H_{su} + H_{ud}$. Such an approximation would reduce the complexity of computing the conditional entropies $H_{sd|u}$ because we would be able to use the matrix of trajectory

entropies H —$\mathcal{O}\left(N^3\right)$ complexity to compute the entropy between all pairs $s, d \in V^2$ —to approximate the conditional entropy $H_{sd|u}$. In fact, a short development of the results of Kafsi et al. [6], [8] gives

$$H_{sd|u} - (H_{su} + H_{ud}) = H_{su} - H_{su|\bar{d}},$$

which implies that quantities $H_{sd|u}$ and $H_{su} + H_{ud}$ are very close if the distributions $p(T_{su})$ and $p(T_{su}|T_{su} \notin \mathcal{T}_{su}^d)$ are very similar. This is the case, for example, if the probability that the trajectory T_{su} goes through the state d is very low. In such a situation, the pre-computation, which is performed once, has a complexity $\mathcal{O}\left(N^3\right)$, and the expected time complexity of segmenting a trajectory of length l is $\mathcal{O}\left(l \log l\right)$. For future work, we plan to work on this approximation and provide bounds for the difference between the $H_{sd|u}$ by the sum of entropies $H_{su} + H_{ud}$.

V. Experimental evaluation

In this section, we apply the entropy-based segmentation to the GPS trajectories of the Geolife Project, and show that it is able to accurately infer waypoints along a trajectory, without having access to time information.

A. Trajectory Segmentation

Detecting waypoints: As we do not have data that classifies intermediate locations as waypoints, we use the available time information in order to detect potential waypoints. This bears similarity with the approach taken by Zheng et al. [17] who analyze the same dataset and classify a location as a stay point if an individual stays within an area around it for more than 20 minutes. We take this idea further and improve it by comparing the individual behavior to the collective behavior: We assume that a location visited by a user along his trajectory is likely to be a waypoint if this user spends *significantly* more time at this location than the other users typically do. As a consequence, locations where most of the users spend a relatively long time (e.g., crowded areas or train stations) will not be declared as waypoints. More formally, we associate with each location x a Gaussian distribution of residence time $\mathcal{N}\left(\mu_x, \sigma_x\right)$, whose parameters are learnt from behavior of the whole population observed. For a user moving along a given trajectory t_{sd}, an intermediate location $u \in t_{sd}$ is considered to be an intermediate destination if the time this user spends at u is classified as an outlier by the Chauvenet's criterion [15] applied on the distribution $\mathcal{N}\left(\mu_x, \sigma_x\right)$. This criterion states that, given a dataset of n observations produced by a Gaussian distribution, we consider a data point as an outlier only if the probability of observing its deviation from the mean is less than $\frac{1}{2n}$. In order to check whether our results are consistent independently of the choice of outlier detection method, we tested different outlier detection methods and obtained consistent results. We denote by $\mathcal{W}(t_{sd})$ The set of waypoints associated with the trajectory t_{sd}.

We apply this waypoint detection procedure to the Geolife GPS trajectories and observe that the majority (more than 87%) of the trajectories has no waypoints, i.e., $\mathcal{W}(t_{sd}) = \emptyset$. Among the trajectories that admit at least one waypoint, the clear majority (around 90%) has only one waypoint. We will therefore focus on assessing the performance of different segmentation methods on finding, for a given trajectory t_{sd} *a*) whether this trajectory admits waypoints, and *b*) if yes, finding the waypoint where the user spends most of her time

$$w = \underset{u \in \mathcal{W}(t_{sd})}{\arg\max}\, r(t_{sd}, u)$$

Baseline methods: In order to assess the performance of our approach at trajectory segmentation, we consider different baseline methods that rely on different heuristics for trajectory segmentation. As the challenge is to uncover waypoints with no information about time, all the methods presented here share the fact that their heuristics are based on the structure of the trajectory only. Each method first constructs a set of candidate waypoints $\hat{\mathcal{W}}(t_{sd})$, and then chooses the way point $\hat{w}$ that maximizes a given heuristic . The baseline methods are as follows:

Random (R) assumes that each trajectory admits waypoints and selects uniformly at random one the points of the trajectory t_{sd}.

Geo Stretch (GS) The set of candidate waypoints $\hat{\mathcal{W}}(t_{sd})$ is composed of the intermediates locations that are not on the direct line from s to d. The waypoint is the intermediate location that is the furthest from the segment with s and d as end points. This simple yet strong baseline is used in the very popular Ramer-Douglas-Peuker algorithm [11] to select the point on a trajectory that is the furthest from the approximating line segment between s and d.

Path Stretch (PS) We consider the weighted mobility graph introduced in III-A. The weight associated with an edge (i, j) is equal to $-\log(P_{ij})$ where P_{ij} is the probability of visiting location j given that we are at location i. These weights favor the transitions that are the most frequently observed. A simple computation gives that the cost of a trajectory t_{sd} is equal to $-\log p\left(t_{sd}\right)$, which implies that the more probable a path is, the less costly it is. The set of candidate waypoints $\hat{\mathcal{W}}(t_{sd})$ is composed of the intermediate locations that are not along the shortest path from s to d. The waypoint is the candidate location $\hat{w}$ that maximizes the path cost.

Entropy (E) The set of candidate waypoints $\hat{\mathcal{W}}(t_{sd})$ is composed of the intermediate locations whose conditional entropy $H_{sd|u}$ is larger than the trajectory entropy H_{sd}. The waypoint is the candidate location $\hat{w}$ that maximizes conditional entropy

$$\hat{w} = \underset{u \in \hat{\mathcal{W}}(t_{sd})}{\arg\max}\, H_{sd|u}.$$

Empirical evaluation: In order to evaluate the performance of the different segmentation methods, we repeat the following process 100 times: we divide randomly the dataset of trajectories in a training set (90 % of the data) and a test set (10 % of the data). Then, we train a Markovian mobility model based on the trajectories of the training set and we apply the different segmentation methods to the trajectories of the test set. As we do not have access to a ground truth about waypoints, we consider the results produced by the time-based classification procedure, introduced in the beginning of Section V-A, as target values. We assess the performance of each segmentation method by measuring *a*) its average

	Residence time (std) [s]	Distance (std) [hops]
R	209 (73)	6.1 (1.2)
GS	570 (121)	2.5 (0.6)
PS	700 (114)	1.76 (0.27)
E	**1151** (150)	**1.41** (0.28)

TABLE I. THE AVERAGE PERFORMANCE OF THE ENTROPY BASED SEGMENTATION (E) COMPARED TO BASELINE METHODS (R, GS, PS). THE AVERAGE DISTANCE BETWEEN THE ENTROPY-BASED METHOD'S GUESS AND THE ACTUAL WAYPOINT IS 1.4 HOPS ON AVERAGE, WHICH REPRESENTS A 43% IMPROVEMENT OVER THE GS SEGMENTATION, AND 20% IMPROVEMENT OVER THE PS SEGMENTATION.

	Accuracy	F_1 score
R	0.1	0.16
GS	0.12	0.18
PS	0.47	0.25
E	**0.7**	**0.6**

TABLE II. THE AVERAGE CLASSIFICATION ACCURACY AND AVERAGE F_1 SCORE OF THE ENTROPY BASED SEGMENTATION (E) COMPARED TO BASELINE METHODS (R, GS, PS).

classification accuracy and the average F_1-score (harmonic mean of precision and recall), *b*) the average residence time $r(t_{sd}, \hat{w})$ at the location classified as waypoint, and *c*) the average distance (number of hops) between the waypoint guess $\hat{w}$ and the actual waypoint w.

We report the results, obtained by averaging the results of the process presented above, in Tables I and II. The methods whose heuristics are based on the statistics related to the population mobility (PS and E) perform better than purely geographical heuristics (GS). This is not surprising as heuristics that are based on geographical distances fail to capture paths that are geographically costly but very popular (i.e., a long route that includes many points of interests is more popular than a short route that has none).

Among the methods that are based on the mobility graph, the entropy-based segmentation is clearly the best: It takes advantage of the entropy-based heuristic that describes the evolution of whole the distribution of trajectories, as opposed to the PS that is based on the evolution of path probability only. This also confirms that trajectory entropy captures much more than simply the evolution of the cost of the shortest path.

By looking at Table I, we see that the average residence time at the locations classified as waypoints by our method is larger than the residence time at the locations classified as such by the baseline methods (102% larger than GS, 64% larger than PS). This indicates clearly that the entropy based segmentation is the best at retrieving intermediate locations where users spend significant amounts of time. More importantly, the average distance between our method's guess and the actual waypoint is 1.4 on average, which is a 43% improvement over the GS segmentation, and a 20% improvement over the PS segmentation. We can see this average distance as a measure of the waypoints' privacy [12], which implies that the privacy of waypoints is the lowest when the adversary's estimate of the waypoint is based on trajectory entropy.

Table II shows the average accuracy and F_1 score of each method. The fact that R and GS perform poorly is not surprising if we know that the majority of the trajectories in our dataset admits no waypoints: the random segmentation method declares systematically that a trajectory admits a waypoint while GS declares the same as soon as the trajectory t_{sd} deviates from the direct line from s to d. Our method is the most accurate —48% more accurate than PS —and is able to classify correctly an important proportion (70%) of the trajectories that have a waypoint. Moreover, it offers the best trade-off between precision and recall, with a F_1 score equal to 0.60. Note that, for all methods, the precision is lower than the recall: observing an intermediate location that deviates significantly from the most probable path, or increases trajectory entropy does not always imply, with certainty, that this location is a waypoint.

Taken together, these results strongly support the possibility of uncovering waypoints of a trajectory without having access to time information: Computing the conditional trajectory entropy associated with a parsimonious mobility model enables us to detect structural outliers that are very likely to be waypoints and not simple intermediate locations.

VI. INTERPRETATION

In this section, we explore further the link between trajectory conditional entropy and the nature of intermediate locations and analyse an additional dataset in order to confirm the results found. In the first part of this section, we consider a mobility graph with a very high location-resolution: We consider a real city map and represent a user's mobility using a route choice model borrowed from the research literature on urban mobility [7]. In the second part of this section, we focus on a subset of the Geolife datatset [17] and illustrate which intermediate locations increase/decrease trajectory entropy for a specific pair of source and destination.

A. Trajectories on City Maps

Open Street Map dataset: The urban environment in which we evolve is usually described by a road map that is represented as a graph. In order to obtain this representation, we use the data freely available from the project OpenStreetMap (OSM). OSM is a collaborative project, with over 1.9 million registered users, created in order to provide free geographic data. The maps provided by the OSM project are represented using a standard geospatial vector data format called shapefile. We download these shapefiles and process them in order to represent the city road map as a graph $G(V, E)$ similar to the one shown in Figure 5. In addition to being connected, the graph G is weighted: we associate with each edge $(i, j) \in E$ a cost $c_{ij} > 0$ equal to its length.

Route choice model: For a given source vertex s and destination vertex d, we associate with each edge (i, j) a weight $\omega_{(i,j)|s,d}$ defined as

$$\omega_{(i,j)|s,d} = 1 - \left(1 - \left(\frac{D_{sd}}{D_{si} + c_{ij} + D_{jd}}\right)^{b_1}\right)^{b_2}, \quad (5)$$

where D_{ij} is the cost of the shortest path between vertices i and j. The weight (5) is inspired from the cumulative

Fig. 5. The graph extracted from the geospatial data of OSM about an European city. The vertices and edges of the graph enables the representation of the geometrical shape of the roads.

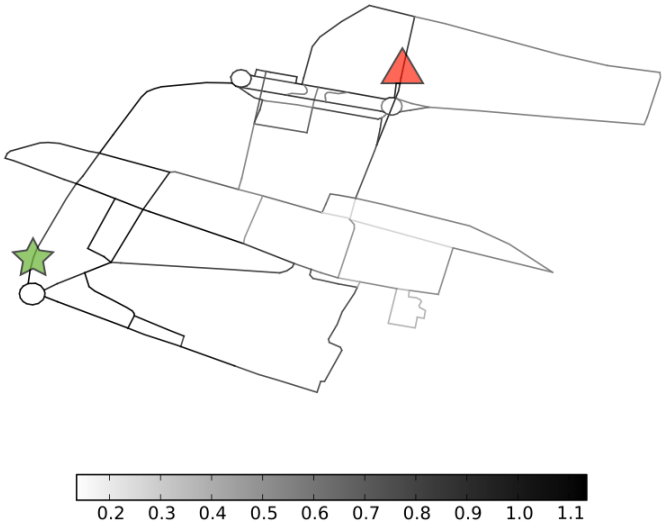

Fig. 6. The graph $G_L = (V_L, E_L)$ represents an area around the train station of an European city. We focus on the trajectory between the vertices s (green star) and d (red triangle), and color each directed edge (u, v) with a color proportional to the value of the conditional entropy $H_{sd|(u,v)}/H_{sd}$. Light gray represents a low value of entropy and hence a high trajectory predictability.

distribution function of Kumaraswamy's double-bounded distribution [9], defined on the interval $[0, 1]$ and having two non-negative shape parameters b_1 and b_2. It indicates to which extent the cost of a path going through the edge (i, j) deviates from the cost of the shortest path between the vertices s and d. The weights definition is based on the paper [7] in which the authors propose a method to stochastically generate paths for a given origin-destination pair, without having to enumerate all paths between these points. We model the user mobility as a *second* order MC whose state space is the set of vertices V. We choose a second order MC because we want to keep in memory the momentum of the mobility and to have a more realistic behavior: The next location a user will visit is different from the one she just left. Equivalently, we can represent this second order MC as a first order Markov chain X_i with an extended state space E: a state represents a sequence of two connected vertices (i, j). Therefore, the transition probabilities are given by

$$P_{(i,j),(k,l)} = P(X_{n+1} = (k, l)|X_n = (i, j)).$$

As we are interested in the mobility between two fixed vertices s and d, we define the transition probabilities between the states $(i, j), (k, l) \in E$ as

$$P_{(i,j),(k,l)} = \begin{cases} \frac{\omega_{(k,l)|s,d}}{\sum_{l' \in \Gamma(k) \setminus l} \omega_{(k,l')|s,d}} & \text{if } j = k, \\ 0 & \text{otherwise,} \end{cases} \quad (6)$$

where $\Gamma(k)$ is the neighborhood set of vertex k. In Figure 6, we plot the road map $G_L = (V_L, E_L)$ of an area surrounding the train station in an European city. We are interested in the trajectories between two locations represented by the vertices s (green star) and d (red triangle). Using the mobility model defined in (6) with $b_1 = 2$ and $b_2 = 1$, we obtain a second order MC, where a state represents a sequence of two vertices or, more simply, a directed edge in E_L.

Trajectory entropy and mobility predictability: The entropy H_{sd} is equal to 7.13 bits, the expected number of bits needed to represent the random trajectory T_{sd}. We plot in the Figure 6 the graph G_L and color each edge $(i, j) \in E_L$ with a color that is proportional to the value $H_{sd|(i,j)}/H_{sd}$. First, we notice a high variability of this quantity whose range is the interval $[0.13, 1.1]$. Unsurprisingly, this means that we cannot consider location updates as having an equal effect on the trajectory predictability: revealing one location can have almost no effect on the predictability of a trajectory, whereas revealing another location can be very threatening to privacy as it drastically increases trajectory predictability. In order to understand the cause of an important decrease in the entropy value, we have to dig a bit deeper and study the trajectory conditional distribution. We observe that the distribution of trajectories conditioned on the directed edge that minimizes entropy is dominated by two trajectories with very close probabilities. If we reveal this intermediate edge, the randomness of the trajectory would be equivalent to the randomness of Bernoulli random variable with $p \simeq 0.5$. Naturally, a location along this edge will not be classified by our algorithm as a waypoint because it decreases trajectory entropy.

B. Entropy and Mobility Predictability

We take the same approach presented in Section III-A but focus on the pair of locations (s, d) with the largest set of trajectory realizations. As we are able to visualize these raw trajectories, this enables us to gain more intuition into the link between conditional trajectory entropy and the nature of intermediate locations. We plot the *raw* trajectories in Figure 7 and observe that two main roads, with similar lengths, allow for reaching the destination d.

We show in Figure 8 the set of locations that, when revealed, decreases most significantly the trajectory entropy. They are within the square that is along one of the two main roads leading from s to d. Knowing this, we are not surprised that the entropy decreases to 73% of its initial value by just conditioning the trajectory on going through this blue square. Revealing this intermediate location excludes the

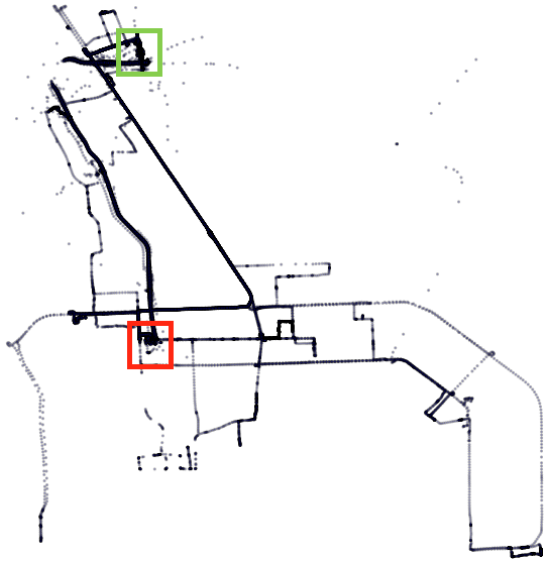

Fig. 7. We plot all the *raw* trajectories, starting inside the area delimited by the green square (upper part) and ending inside the area delimited by the red square (lower part). Observe that the starting and ending areas are connected by two main roads.

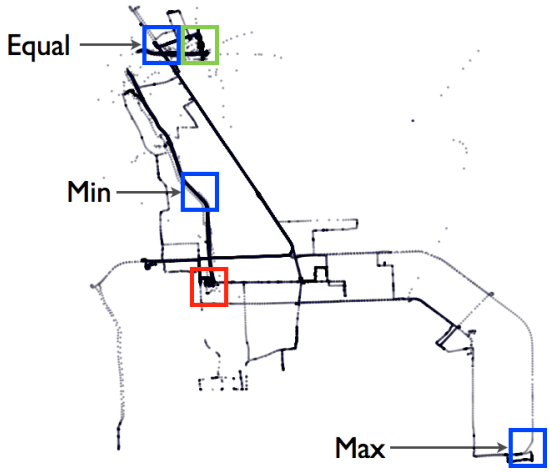

Fig. 8. It is not surprising that revealing an intermediate location that is adjacent to the starting location has a small effect on the trajectory distribution and decreases the entropy by only 2%. However, revealing an intermediate location in an area along one of the two main paths decreases the trajectory entropy to 73% of its initial value, which significantly increases trajectory predictability. Moreover, revealing the location in the lower-right corner changes completely the distribution of trajectories and maximizes conditional entropy.

trajectories that go through the second main road leading to d, thus increasing significantly the predictability of the trajectory taken.

In contrast, revealing some other locations has a small effect on the predictability of the trajectory. In fact, revealing the blue square just next to the starting location when heading towards the destination has a small effect on the trajectory distribution and decreases the entropy by only 2%.

Figure 8 shows how revealing an intermediate location —this location is on a lengthy path between the source and destination —increases the entropy by 70%. Visualizing the raw trajectories enables us to see that this intermediate destination is not on the most popular paths between the source and destination. Our segmentation algorithm, applied to a trajectory that goes through this intermediate location, will correctly classify this intermediate location as a waypoint.

VII. Related Work

In this section, we briefly present additional references to the related work about trajectory segmentation, trajectory entropy and mobility predictability.

Trajectory segmentation: The classic idea of trajectory segmentation [2], [5], [16] is to obtain segments where movement characteristics inside each segment are similar. Movement characteristics might be attributes of the points of the trajectory such as speed, direction, or curviness. For example, the Ramer-Douglas-Peuker algorithm [11] segments a trajectory based on the deviation of its points from the segmented curve. Buchin et al. [2] propose a segmentation algorithm that associates a profile to every point based on its attributes (speed, heading and curvature). Then, they express the segmentation process as an optimization problem where the goal is to have a similarity within a segment higher than a given threshold. In this work, our goal is to segment trajectories by measuring the "deviation" from the typical distribution of trajectories followed by the population. To the best of our knowledge, we are the first to segment trajectories, using a model of population mobility and with no time information.

Trajectory entropy: In [6], Ekroot and Cover study the computational aspect of the depth measure as introduced by Lloyd and Pagel [10]. In order to quantify the number of bits of randomness in a Markov trajectory, they propose a closed-form expression for the entropy of trajectories of an irreducible Markov chain. Kafsi et al. [8] provide a general closed-form expression to compute the entropy of Markov trajectories conditional on multiple intermediate states.

Mobility predictability: Song et al [13] study the predictability of human mobility using a mobile phone dataset of $45,000$ mobile phone users. They quantified the users' mobility predictability by approximating their mobility entropy rate and found out that, on average, 1 bit of information is needed to describe the next cell tower visited by a user given his whole mobility history. In our work, we go beyond the predictability of the next location visited and are able to quantify in the predictability of the whole trajectory. Moreover, we quantify the impact of revealing a subset of the locations visited on mobility predictability.

VIII. Conclusion

In this work we have proposed a trajectory segmentation method based on the computation of the entropy of conditional Markov trajectories. We have shown empirically that the entropy of a trajectory conditioned on a particular location is a powerful metric for estimating whether this location is likely to be a waypoint or not, and more generally for revealing whether knowing this location makes the trajectory more or less predictable. Using this observation, we have developed an algorithm that is able to efficiently segment trajectories: We take advantage of a model of population mobility and quantify to which extent this individual trajectory deviates from the

plausible behaviors. The advantage of our approach is that little information about the individual trajectory is needed. In particular, no timestamps nor absolute geographic locations are used. This implies that data miners would be able to uncover important intermediate locations for privacy-preserved trajectories that are not associated with time information or for very sparse trajectories.

More generally, we believe the entropy of conditional trajectories is a powerful tool to study dynamics on graphs, because it captures the evolution of the distribution of trajectories as intermediate locations are revealed. The results presented in Section VI also open up interesting directions for future research. For example, we are able to quantify, using the trajectory entropy framework, the privacy risk of revealing a subset of the locations visited (check-ins) along a trajectory.

APPENDIX

We study the particular case of a random walk on a regular graph and show how the entropy H_{sd} is proportional to the hitting time S_{sd} .i.e., the expected number of steps before the state d is visited, starting from state s. This equality is particularly interesting because it links trajectory entropy to the hitting times, a popular measure in graph theory and used, among others, to quantify the similarity between vertices [3].

Proposition 1: Consider a random walk on a δ-regular graph $G(V, E)$ defined by the transition probability matrix P whose entries read

$$P_{ij} = \begin{cases} 1/\delta & \text{if } (i,j) \in E, \\ 0 & \text{otherwise.} \end{cases}$$

Then, we have the following equality for all $s, d \in V, s \neq d$

$$H_{sd} = \log(\delta)\, S_{sd}.$$

Proof: By definition, we have

$$H_{sd} = -\mathbb{E}\left[\log p(T_{sd})\right].$$

Since the trajectory is generated by a random walk on a δ-regular graph, we have

$$p(t_{sd}) = \frac{1}{\delta^{l(t_{sd})}},$$

where $l(t_{sd})$ be the length of the trajectory t_{sd}. Thus

$$\begin{aligned} H_{sd} = -\mathbb{E}\left[\log\left(\frac{1}{\delta^{l(t_{sd})}}\right)\right] &= -\mathbb{E}\left[l(T_{sd}) \log\frac{1}{\delta}\right] \\ &= \log(\delta)\, \mathbb{E}\left[l(T_{sd})\right]. \end{aligned}$$

The expected length $\mathbb{E}\left[l(T_{sd})\right]$ of the random trajectory T_{sd} is equal to the hitting time S_{sd} because the trajectory T_{sd} is a path from s to d that terminates as soon as it reaches state d. Therefore

$$H_{sd} = \log(\delta)\, S_{sd}.$$

■

In other words, since all nodes have the same degree, the entropy of a trajectory generated by a random walk on a regular graph is proportional to its expected length.

REFERENCES

[1] P. Banerjee, S. Ranu, and S. Raghavan. Inferring uncertain trajectories from partial observations. In *Data Mining (ICDM), 2014 IEEE International Conference on*, pages 30–39, Dec 2014.

[2] Maike Buchin, Anne Driemel, Marc van Kreveld, and Vera Sacristán. An algorithmic framework for segmenting trajectories based on spatio-temporal criteria. In *Proceedings of the 18th SIGSPATIAL*, GIS '10, pages 202–211, New York, NY, USA, 2010. ACM.

[3] Mo Chen, Jianzhuang Liu, and Xiaoou Tang. Clustering via random walk hitting time on directed graphs. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2*, AAAI'08. AAAI Press, 2008.

[4] Eunjoon Cho, Seth A. Myers, and Jure Leskovec. Friendship and mobility: user movement in location-based social networks. In *KDD '11*, New York, NY, USA, 2011.

[5] Somayeh Dodge, Robert Weibel, and Ehsan Forootan. Revealing the physics of movement: Comparing the similarity of movement characteristics of different types of moving objects. *Computers, Environment and Urban Systems*, 33(6):419 – 434, 2009.

[6] L. Ekroot and T. M. Cover. The entropy of markov trajectories. *IEEE Trans. Inf. Theor.*, 39(4):1418–1421, September 2006.

[7] E. Frejinger, M. Bierlaire, and M. Ben-Akiva. Sampling of alternatives for route choice modeling. *Transportation Research Part B: Methodological*, 43(10):984 – 994, 2009.

[8] M. Kafsi, M. Grossglauser, and P. Thiran. The entropy of conditional markov trajectories. *Information Theory, IEEE Transactions on*, 59(9):5577–5583, Sept 2013.

[9] P. Kumaraswamy. A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1 - 2):79 – 88, 1980.

[10] Seth Lloyd and Heinz Pagels. Complexity as thermodynamic depth. *Annals of Physics*, 188(1):186 – 213, 1988.

[11] Urs Ramer. An iterative procedure for the polygonal approximation of plane curves. *Computer Graphics and Image Processing*, 1(3):244 – 256, 1972.

[12] R. Shokri, G. Theodorakopoulos, J. Le Boudec, and J. Hubaux. Quantifying location privacy. In *2011 IEEE Symp. on Security and Privacy (SP)*.

[13] Chaoming Song, Zehui Qu, Nicholas Blumm, and Albert-Laszlo Barabasi. Limits of Predictability in Human Mobility. *Science*, 327(5968):1018–1021, 2010.

[14] Dashun Wang, Dino Pedreschi, Chaoming Song, Fosca Giannotti, and Albert-Laszlo Barabasi. Human mobility, social ties, and link prediction. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 1100–1108, New York, NY, USA, 2011. ACM.

[15] Wikipedia. Chauvenet's criterion— Wikipedia, the free encyclopedia, 2015. [Online; accessed June-2015].

[16] Hyunjin Yoon and Cyrus Shahabi. Robust time-referenced segmentation of moving object trajectories. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, ICDM '08, pages 1121–1126, Washington, DC, USA, 2008. IEEE Computer Society.

[17] Yu Zheng, Lizhu Zhang, Xing Xie, and Wei-Ying Ma. Mining interesting locations and travel sequences from gps trajectories. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 791–800, New York, NY, USA, 2009. ACM.