Clustering Users by Their Mobility Behavioral Patterns

IRAD BEN-GAL and SHAHAR WEINSTOCK, Tel Aviv University
GONEN SINGER, Afeka College of Engineering
NICHOLAS BAMBOS, Stanford University

45

The immense stream of data from mobile devices during recent years enables one to learn more about human behavior and provide mobile phone users with personalized services. In this work, we identify clusters of users who share similar mobility behavioral patterns. We analyze trajectories of semantic locations to find users who have similar mobility "lifestyle," even when they live in different areas. For this task, we propose a new grouping scheme that is called Lifestyle-Based Clustering (LBC). We represent the mobility movement of each user by a Markov model and calculate the Jensen–Shannon distances among pairs of users. The pairwise distances are represented by a similarity matrix, which is used for the clustering. To validate the unsupervised clustering task, we develop an entropy-based clustering measure, namely, an index that measures the homogeneity of mobility patterns within clusters of users. The analysis is validated on a real-world dataset that contains location-movements of 50,000 cellular phone users that were analyzed over a two-month period.

CCS Concepts: • **Computing methodologies** → **Bayesian network models**; *Cluster analysis*; *Anomaly detection*; • **Information systems** → *Location based services*; *Data analytics*; *Clustering*;

Additional Key Words and Phrases: Clustering trajectories, clustering evaluation

ACM Reference format:
Irad Ben-Gal, Shahar Weinstock, Gonen Singer, and Nicholas Bambos. 2019. Clustering Users by Their Mobility Behavioral Patterns. *ACM Trans. Knowl. Discov. Data* 13, 4, Article 45 (August 2019), 28 pages.
https://doi.org/10.1145/3322126

1 INTRODUCTION

Mobile phones play an integral part in our lives, as we carry them everywhere we go. A mobile device can track its own location at every moment by various means, such as GPS, Wi-Fi, and triangulation algorithms of cellular base stations. Location data are being sent and accumulated by mobile-phone operators and third parties that gain access to the phone's data (e.g., through smart-phone applications). Many companies currently have the ability not only to capture the locations of the user but also to give semantic meaning to those locations using online applications of reverse geocoding, points of interest (POIs), and the user's own personal data. The extraction of semantic locations from raw geographic data is a complex process that has been well described

This work is supported by the Israeli Ministry of Science and Technology, under Grant 3-8709: Learning and mining mobility patterns using stochastic models and by the Koret Foundation Digital Living 2030 Grant.
Authors' addresses: I. Ben-Gal (corresponding author) and S. Weinstock, Industrial Engineering and Management Department, Tel-Aviv University; emails: Bengal@tau.ac.il, shahar.weinstock@gmail.com; G. Singer, Industrial Engineering and Management Department, Afeka College of Engineering; N. Bambos, MS&E, Stanford University; email: singerg@afeka.ac.il; N. Bambos; email: bambos@stanford.edu.

1556-4681/2019/08-ART45 $15.00
https://doi.org/10.1145/3322126

in previous works [19, 29, 48]. Such semantic locations data provide a unique foundation for the learning of human mobility behavior and can serve as an important indication of users' lifestyles and other personal behavioral features.

This new capability of understanding human behavior can play an essential role in the future evolution of "smart cities" [5, 36, 42], in which new technologies use digital data to increase the wellbeing of citizens and reduce costs, pollution, and resource consumption. The more data these technologies have on the citizens' lifestyles, the higher the personalization potential of various public services for the citizens. These services can be coupled with other personalized services and applications, such as telematic-based car insurance (determining a personalized insurance cost for each driver based on his or her lifestyle and behavior), smart homes, and user segmentation for marketing purposes [2]. These trends have gained huge popularity in recent years by many organizations that support location-based services, which offer users various services that are related to their geographic locations [6]. With the identification of a user's lifestyle, as proposed in this study, some of these personalized services can be modified to support also "pattern-based services" that are not necessarily related to the user's geographic location but rather to his mobility patterns and lifestyle. For example, a similarity can be established between two people that live in different cities yet exhibit similar mobility behavior by starting and ending their workdays early 3 days per week, staying at home 2 days per week, and travelling out of town on weekends. Personalized delivery times and convenient time-windows to call these users are two obvious examples for life-style mobility applications that are not directly related to their geographic location.

This work is focused on the task of clustering mobile phone users based on their mobility pattern behaviors, i.e., their movements between semantic locations during different time periods. By using semantic locations instead of geographical points, clusters of users can represent, for example, similar behaviors of users regardless of their physical locations. Therefore, one can identify people with the same lifestyle even when they live in different areas. Following Ye et al. [50], we assume "lifestyle similarity" among people when they have the following:

(1) Do not necessarily share the same location at the same time;
(2) Share a similar frequency of semantic (not necessarily geographic) locations;
(3) Share similar mobility patterns when moving between these locations.

Note that most of the previous works in this area considered users to be similar when they share the same location at the same time and used the various methods, such as the popular Longest Common Subsequence (LCSS) method for sequence alignment, PCA for dimensionality reduction and Mixture Models [52]. However, none of these methods address the second or third assumptions that are listed above. It was previously shown in [46] that user behavior exhibits strong periodic patterns and that given the current time and location of the user, one can predict his next location with a relatively high probability using various data mining methods. In this work, we identify those patterns for each user by profiling the individual's behavior using a stochastic Markovian model. We show that applying a simple first-order Markov model to mobility patterns of users is computationally tractable yet enables us to capture main patterns among adjacent time intervals in the user's trajectory. When using probabilistic profiling, the distance between two users is evaluated based on the distance between their corresponding Markovian transition probability matrices. For this purpose, we apply the Jensen–Shannon (JS) distance, which is well known in information theory and based on a symmetric implementation of the Kullback–Leibler divergence to a pair of multidimensional distributions. Using the JS distance results in a pair-wise distance matrix among the users on which we can apply known clustering algorithms. Three clustering methods, namely, K-medoids, hierarchical clustering and spectral clustering, are evaluated and compared to find the most suitable clustering method for the abovementioned "Lifestyle-Based Clustering" (LBC) task.

As shown later, in a series of experiments it is found that the agglomerative hierarchical clustering is the most efficient clustering method for the considered task over several clustering metrics.

To compare the proposed LBC method to other approaches that use different metrics, and hence do not fit the LBC task, we introduce a metric-independent index that measures how well the clustering outcomes fit the second and third "Lifetime-Similarity" assumptions listed above.

The contribution of the article is fourfold. The first one is related to the addressed challenge, namely, the introduced LBC that groups users with similar mobility patterns over semantic (and not necessarily geographic) locations, hence, enabling to cluster people with the same lifestyle even when they live in different areas and move in different times. The second contribution is the proposed methodology that models the users' mobility behavior by Markov chains (starting with a simple Markov model and up to a non-homogeneous Variable-Order Bayesian Network (VOBN) modeling that is shortly discussed later), apply the JS distance measure between each pair of users to obtain a distance matrix, on which a clustering algorithm can be executed. The proposed method introduces an entropy-based validation index that is metric independent. Moreover, the procedure is shown to be computationally tractable with respect to other conventional methods. The third contribution is the unique dataset that was studied—a real dataset that contains the mobility data of tens-of-thousands of users over two months. This size of the dataset is unique with respect to previous works that usually contain only tens or hundreds of users. The final contribution is the considered "smart city" application that analyzes the distribution of different segments of users, where each segment represents a different mobility behavior, over five different cities. Such an analysis is shown to be correlated with socio-demographic scores and found to be valuable to several "smart-city" services that are discussed in the article.

The rest of this article is organized as follows: Section 2 presents related work. Section 3 introduces the LBS method for clustering users by their behavioral mobility patterns. Section 4 presents the metrics for evaluating the LBS results. Section 5 analyzes and validates the obtained results. Section 6 concludes the work and discusses future research directions.

2 RELATED WORK

In this work, the input to the user mobility behavior model is a trajectory of semantic locations of the user in a specific time period. Thus, the user profile is represented as a sequence of locations, which indicate a path that the user traveled in a specific time interval or, alternatively, very often. Unlike the conventional location-based profiling, by which the user profile is based on the location distribution, the proposed modeling approach considers the time period (the hour in the day) in which the user is present at each location and the patterns of locations that she traverses when moving from one location to another.

There are different ways to extract a trajectory from the user's location history. Most procedures are based on identifying places in which the user did or did not move, by spatiotemporal constraints [1, 55], clustering the places into a small number of locations [3] and even finding the semantic meaning of each place using third-party data [43, 49]. For example, Ashbrook et al. [3] first addressed the task of finding significant places in the user's movements. They suggested that the most logical way to find points that the user might consider significant is to look at where the user spends her time. It is unlikely that the user would consider somewhere where she never stopped (e.g., the middle of the highway) worth consideration. They used this concept to find what they call "places." They defined a place as any logged coordinate with an interval of time of length $t = 10$ seconds between it and the previous point. Next, they created clusters of places using a variant of the k-means clustering algorithm. They called the resulting clusters "locations" and used them instead of places. Yan et al. [49] used another layer on top of locations that annotates the stopping episodes of a trajectory with information about suitable POIs from a different dataset that maps the area. Examples of POIs are restaurants, bars, shops, and movie theaters. Next, they designed a Hidden Markov Model (HMM)-based technique for inferring the semantic annotations of stops.

For the task of finding similar trajectories of users, there are two main ways to treat a trajectory: one is "as it is," i.e., as a deterministic sequence of locations, and the other is to build a stochastic model based on a set of observed trajectories. We will present both approaches and for each one, discuss several methods that were used in the literature.

2.1 Deterministic Models and LCSS Similarity

The most common approach for mining similar trajectories is to define a distance metric between pairs of sequences and apply a clustering algorithm to the precomputed distances.

The *longest common subsequence* (LCSS) is a similarity measure that is frequently used in tasks of clustering trajectories. It is a special case of the Edit distance in which dissimilarity is measured only as the number of insertions and deletions (without substitution) that are required to make two sequences match. Since the LCSS is heavily used to measure distances among location trajectories, the next subsection reviews a few research works that use this similarity measure and its variants.

Valchos et al. [45] utilized LCSS to find similar multidimensional trajectories (our initial focus is on a single-dimensional case): Let A and B be two trajectories with sizes n and m, respectively, where $A = (a_1, a_2, \ldots, a_n)$ and $B = (b_1, b_2, \ldots, b_m)$. For trajectory A, let $Head(A)$ be the sequence $Head(A) = (a_1, a_2, \ldots, a_{n-1})$.

Definition 2.1. Given an integer δ and a real number $0 < \varepsilon < 1$, $LCSS_{\delta,\varepsilon}(A, B)$ can be obtained by a dynamic programming calculation that is defined as follows:

$$\begin{cases} 0 & \boldsymbol{if}\ A \text{ or } B \text{ is empty} \\ 1 + LCSS_{\delta,\varepsilon}(Head(A), Head(B)),\ \boldsymbol{if}\ |a_n - b_m| \leq \varepsilon\ \boldsymbol{and}\ |n - m| \leq \delta & \\ \max(LCSS_{\delta,\varepsilon}(A, Head(B)), LCSS_{\delta,\varepsilon}(Head(A), B)), & \boldsymbol{otherwise} \end{cases}$$

The constant δ controls the time stretching, i.e., how far in time one can go to match a given point from one trajectory to a point in another trajectory. This means that the significance of the time interval is important, but an exact match is not necessary for two trajectories to be similar. The constant ε is the matching threshold—implying that one allows matching of close-enough values and not only exact matches.

Definition 2.2. The similarity function $S1$ between two trajectories A and B, given ε and δ, is defined as follows:

$$S1(\delta, \varepsilon, A, B) = \frac{LCSS_{\delta,\varepsilon}(A, B)}{\min(n, m)}.$$

Given the similarity between two trajectories, which is represented as a real number from 0 to 1, the distance between the trajectories is simply as follows:

$$D1(\delta, \varepsilon, A, B) = 1 - S1(\delta, \varepsilon, A, B).$$

The LCSS has been used in other works for the task of finding similar users; variations of this calculation can be found in [26, 30, 31, 44, 47, 51, 54].

A completely different approach is to use matrix factorization to lower the dimensionality of the data and learn the latent patterns in them [9, 32]. This approach was taken by [15], who addressed the problem of predicting a user's future activity based on his previous activity. They represented the behavioral structure of the user's trajectories by a set of characteristic vectors, which they have termed eigen behaviors. In their model, an individual's behavior over a specific day can be approximated by a weighted sum of her primary eigen behaviors. By conducting a principle component analysis (PCA) on their data with different numbers of components, they managed to obtain promising results for predicting user location and classifying users into predefined classes. By analyzing a person's behavior in the first half of the day, they were able to predict the day's remaining behaviors with 79% accuracy. Using only 6 components, they obtained more than 90% classification accuracy in the classification task they considered. When using 15 components, they reached an accuracy of over 99%. While there are many techniques for creating predictive models that can generate a sequence of future data after training, eigen-decomposition differentiates itself in an important alternative: The main advantage of this model is its ability to capture patterns of users in non-adjacent timestamps. For example, for many subjects, sleeping late in the morning is coupled in the same eigen behavior with going out that evening, a pattern that is highlighted when generating an individual's characteristic behavior spaces. This advantage exists only when a substantial amount of data on a specific user is available, and it will not work well when one does not have enough sequences since the model will not learn the dependence between the (divided) hours. Although the task in this work was not to predict the user's locations at specific hours (but rather to cluster similar users based on their mobility behavior), we did use PCA in order to reduce the dimensionality of the data before clustering the users. Nonetheless, one must consider that when conducting PCA on the entire dataset of trajectories, some of the benefits of the model are lost since the variety of patterns among users makes it difficult to recognize specific dependencies among non-adjacent hours. In addition, PCA is inhomogeneous over time. Thus, in many cases, it cannot find similar users if their calendars do not match in time.

2.2 Stochastic Models

Stochastic models make a probabilistic assumption on the data [10] and usually define each trajectory as a multidimensional point in space, where each dimension represents a time interval of the trajectory. They often assume that each trajectory is produced from a mixture of Gaussian distributions cluster. To find those clusters, they often use an Expectation–Maximization algorithm. Although such a model uses a probabilistic assumption, it assumes that the probabilities of the instances (time intervals) are independent of one another and, thus, loses an important aspect of the trajectory information.

A similar approach was taken by Ferrari et al. [18], which used Latent Dirichlet Allocation (LDA) to cluster semantic trajectories of a user's mobility. They used a method in which each day is viewed as a mixture of "topics" (i.e., clusters) z, where topics are distributions over words that represent a location at a specific time label (i.e., each topic can be represented by the list of locations and time labels that are associated with these probability $p(x|z)$. For each trajectory i, the probability of a word x is given by $p(x) \ = \sum_{t=1}^{K} p(x|z_k)p(z_k)$, where K is the number of topics and $p(x|z_k)$ and $p(z_k)$ are assumed to follow multinomial distributions. LDA uses the EM algorithms to learn the model parameters. A similar model was used by Farrahi and Gatica-Perez [17], which applied an LDA-based model to discover sequences of location that dominantly occur in a user's mobility patterns. LDA finds more than one topic for each user; hence, it does not fit exactly the considered task of associating each user to a single cluster. In comparison, this work implements a mixture model that assigns each user to a single cluster; thus, it might be considered a particular case of the LDA.

In summary, the main gap that we identify between the proposed approach and previous works is that most of the suggested procedures do not consider semantic locations and dependencies among different time intervals in the users' trajectories. Hence, they cannot identify non-continuous complex patterns in the user mobility behavior (as indicated in by the above third assumption). Methods that address this issue define users as similar only if their calendars match exactly (or very similarly) over time. In the next chapter, we address the task of clustering users

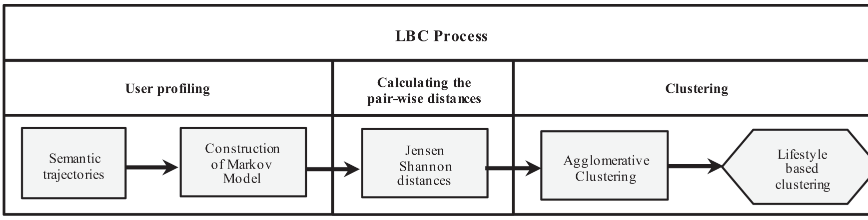


Fig. 1. Lifestyle-Based Clustering procedure.

based on similar transitions that are not necessarily time-identical. This is done by introducing the proposed LBC approach.

3 PROPOSED APPROACH: LIFESTYLE-BASED CLUSTERING (LBC)

The LBC method takes as input a set of semantic locations that represent the mobility traces of users and outputs a cluster label for each user. This LBC follows a three-phase procedure (see Figure 1):

(1) Profiling each user by a stochastic first-order Markov model (transition matrix); where later we consider also higher order or Variable Order Markov models;
(2) Calculating the JS distance for every pair of users over their transition matrices and representing it in a pairwise distance matrix;
(3) Executing a known clustering algorithm on the pairwise distance matrix, where the Hierarchical Agglomerative clustering is found as the preferred method.

In the following sections, we present the three phases of the LBC method.

3.1 User Profiling

The first step in the proposed clustering framework is to represent the deterministic set of location sequences that are associated with each user over a given time period by a stochastic model. In particular, the user mobility behavior, as captured by the location sequences, is described by a conditional distribution function of visited locations given the user's past locations. To capture the dependencies among locations in different time intervals, a first-order Markov model is used to model each user. Although the mobility behavior of users often depends on variable higher order models, as proposed in Ben-Gal et al. [7, 8], a simple first-order model can be used to emphasize important patterns in the user's behavior, while maintaining a relatively lower computational complexity. Let $X_t \in \{x_t\}$ be a random variable that represents the user location at time period t. Then, the conditional probability of the user location at time period $t+1$ given the pattern of past user locations is approximated in this case by a first-order Markov chain, namely, $\Pr(X_{t+1} = x|X_1 = x_1, X_2 = x_2, \ldots, X_t = x_t) \approx Pr(X_{t+1} = x|X_t = x_t)$. Note that the set of locations is semantic and finite. Thus, based on GIS layers and mapping techniques, each location is tagged by the most likely functionality of that location, such as "home," "work," "shop," or "road" [4, 11]. Figure 2 illustrates a mapping of a user's daily sequence of locations at an hourly resolution, where the hourly location is defined by the mode over all visited locations in that hour, to a first-order Markov chain.

The matrix columns' headers represent the locations of the user at hour $t+1$ while rows represent the locations of the user at hour t. The matrix entries are the conditional probability $Pr(X_{t+1} = x|X_t = x_t)$. For example, note that the user visited the "road" location twice that day—at 8am, followed by the semantic "work" location at 9am, and at 6pm, followed by a sematic "shop" location

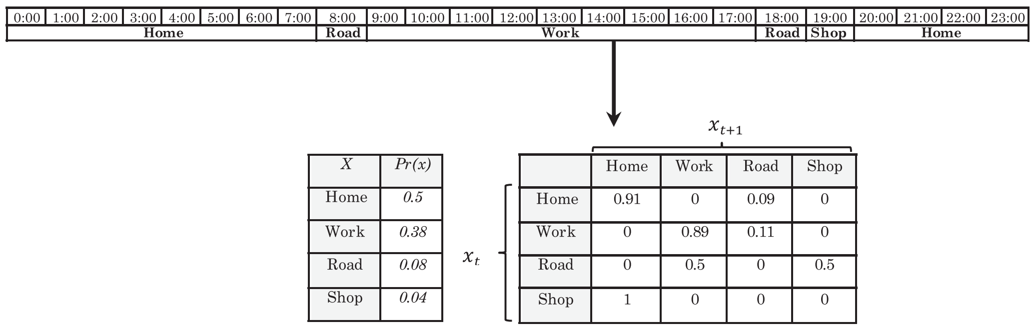


Fig. 2. Construction of a first-order Markov model based on hourly location sequences.

at 7pm. Accordingly, the third row of the transition matrix of the first-order Markov chain shows a 0.5 transition probability from the "road" location to the "work" location and a 0.5 transition probability from the "road" location to the "shop" location. Similarly, given that the user is at work, the probability of her hitting the road in the next hour is $\Pr(X_{t+1} = \text{road}|X_t = \text{work}) = 1/9 = 0.11$.

In addition to the Markov model, we also show the steady-state probability vector, namely, $\Pr(X)$, which indicates the marginal probabilities of the semantic locations, which are measured by the relative frequency of each location for the specific user in a specific time period. Note that the same type of mapping is applied to obtain a Markov model that represents a set of daily location patterns over longer time periods.

A natural extension of the proposed modeling is the replacement of the first-order Markov model by higher order Markov models (that were found to be overfitting in many cases) or by a VOBN [8]. In contrast to the simple Markov model, where the memory order is equal to one and identical for all the locations over all the hours and all days, in VOBN modeling, the memory order can vary for each location, based on the learned patterns. For example, the conditional distribution of the user's location at 6am given is location at 5am does not necessarily depend on his location at 4am (e.g., in most cases, if the user was at home at 5am he was there at 4am as well). Thus, in 6am the required memory order can be limited to 1-hour only. However, usually the conditional distribution of the user's location at 6pm given is location at 5pm does depend on his location at 4pm and, therefore, a model order of 2 or higher is required for this time slot. Another extension of the VOBN is the modeling of non-sequential patterns in the data. Namely, patterns where, for example, the location of the user in 9am affects his location in 5pm but not necessarily his location at 10am (e.g., if the user is in the Gym in 9am he often stays at work at 5pm, yet he is often at work in 10am regardless of his location at 9am). The exact construction algorithm of the VOBN in the context of mobility behavior patterns is long and detailed and, therefore, it is out of the scope of this article and the subject of a future paper.

3.2 Calculating Pair-Wise Distances Among Users

Once each user has been modeled by a Markov model, the distance between two users' models can be calculated as a distance between the corresponding two transition matrices. In particular, the Kullback–Leibler distance, which is also known as the "relative entropy" [24], is a quantity, known in information theory, that measures the difference between two probability distributions. The KL distance between two probability distributions P and Q is defined as follows:

$$D_{kl}(P||Q) \equiv \sum_{x \in X} P(x) log \frac{P(x)}{Q(x)},$$

where $P(x)$ is the probability of x by distribution P and $Q(x)$ is the probability of x by distribution $Q \cdot D_{kl}(P||Q)$ is not a formal metric (e.g., it is not symmetric) and can be viewed as a measure of the "information loss" when P is used to approximate Q.

The KL distance can be used to measure the distance between two users, who are represented by their respective Markov models. The Markov model is treated as a joint probability of two successive locations X_t and X_{t+1}, where $Pr(X_t, X_{t+1}) = Pr(X_t) \cdot Pr(X_{t+1}|X_t)$. Those two last arguments are calculated in the modeling stage, as described above, and their parameters are estimated for each user based on his location log. The KL measure for the relative distance between two joint probability mass functions, namely, $P(X,Y)$ and $Q(X,Y)$, can be partitioned into two terms, with one term representing the distance between the *conditioning* random variables and the other representing the distance between the *conditioned* random variables [25]:

$$D_{kl}[P(X,Y)||Q(X,Y)] \equiv \sum_{x \in X} P(x) log \frac{P(x)}{Q(x)} + \sum_{x \in X} P(x) \sum_{y \in Y} P(y|x) log \frac{P(y|x)}{Q(y|x)}$$

The KL distance is a non-symmetric measure, i.e., $D_{KL}(P||Q) \neq D_{KL}(Q||P)$, and can take on an infinite value, e.g., if for some location x, $Q(x) = 0$. This can damage the clustering results since many of the known clustering algorithms require a formal distance metric and will achieve worse results when symmetry does not hold. To obtain a metric measure, we use the square root of the JS divergence [28], which is a symmetrized and smoothed version of the Kullback–Leibler divergence:

$$D_{JS}(P||Q)^{1/2} \equiv \left(\frac{1}{2} D_{kl}(P||M) + \frac{1}{2} D_{kl}(Q||M)\right)^{1/2}$$

where M is defined as

$$M \equiv \frac{1}{2}(P+Q).$$

The square root of the JS divergence, which is henceforth referred to as the JS distance, enables us to use a formal metric for measurements in the final clustering phase [17], which is described in the next section.

Note that from a computational complexity respective, the construction of a Markov model per each user is of the order $O(J(l^2 + m))$, while the complexity of computing the distance between all pairs of models is of the order $O(J^2 m^2)$, where J denotes the number of models (users' sequences) that can reach tens of thousands of users in a neighborhood; l denotes the number of possible locations' labels, i.e., the alphabet size of the location variable, that is limited in this study to tens of location or less; and m denotes the length of the sequence of locations per user that is limited to hundreds of locations, depending on the analyzed time windows and the sampling frequency. Thus, the most computationally expensive step in the LBC procedure is the construction of the pairwise distances matrix, and the parameters that significantly affect the computational times are the number of modeled users and the length of the sequences. This is also the complexity of the benchmark LCSS model. One way to apply the method in a computationally tractable manner is to parallelize the process over different neighborhoods or areas. For example, analyzing a neighborhood with 50,000 user models, 12 possible semantic locations and sequences of 120 locations per users can take approximately 8–10 hours on a Pentium i-7 6-Core 4.0GHz CPU and can be reduced further by using a GPU server and a fully parallel computing scheme.

Once the pairwise distances matrix is given, both the LBC as well as the LCSS and other methods can use a known clustering algorithm that can fit big-data analytics requirements. For example, a K-means known heuristic algorithms, such as Lloyd's algorithm, can achieve an average linear complexity of $O(d\,J)$, where d is the number of clusters and J is the number of clustered users.

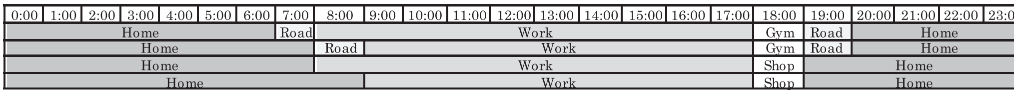


Fig. 3. Simple example of four users' activities.

LCSS Distances				
	User1	User2	User3	User4
User1	0	0.04	0.12	0.17
User2	0.04	0	0.12	0.12
User3	0.12	0.12	0	0.04
User4	0.17	0.12	0.04	0

JS Distances				
	User1	User2	User3	User4
User1	0	0	0.12	0.13
User2	0	0	0.12	0.12
User3	0.12	0.12	0	0
User4	0.13	0.12	0	0

Fig. 4. Pair-wise distances for the simple example using the LCSS and JS metrics.

3.3 Clustering

The final step in the LBC procedure is to cluster the users according to mobility pattern similarity. After calculating the JS distance between each pair of users and obtaining a pairwise similarity matrix, one can perform cluster analysis using various known clustering algorithms. In fact, there are several clustering algorithms and various clustering methods and objectives that can be used. With the understanding that there is no "right" clustering algorithm theoretically for the considered case, but rather a more suitable algorithm for the specific obtained data and the considered application, we applied several known algorithms to find the most suitable one for this study based on real-world data. Each algorithm is based on a different doctrine and relies on different assumptions. The final three selected algorithms, after removing at least dozen methods that were found ineffective, were the K-medoids algorithm [22], the spectral clustering algorithm [35], and the agglomerative clustering algorithm [13]. Our validations on these three clustering algorithms in chapter 5 show a clear advantage for the agglomerative clustering method, which is recommended for use in the LBC case.

3.4 Illustrating Example

Let us refer to a simple example to illustrate the use of the JS distance by the proposed LBC framework, in comparison to the commonly used LCSS method. In this small example, we analyze the mobility behaviors of four users over a single day, as shown in Figure 3.

Each line represents a user's mobility pattern over 24 hours. When clustering these four users into two clusters, one would most likely include user 1 and user 2 in the same cluster as they exhibit similar mobility behavior: both drive to work in the morning and go to the gym immediately after work when driving back home in the evening. User 3 and user 4 will probably be associated in same cluster since they live close to their workplaces, go shopping immediately after work and return home without a long drive.

The distance matrices in Figure 4 present the distances between every pair of users that are obtained using both the LCSS and JS metrics, respectively. The matrix entries are not normalized (this operation is performed later during the clustering process) to give a sense of the distance scales that are used. As expected, the LCSS finds the first two users to be closer to each other, as well as the last two users. The JS metric does the same. However, one can observe its strength as it finds users 1 and 2 to be almost identical (the distance between them is less than 0.005); while the same occurs for users 3 and 4.

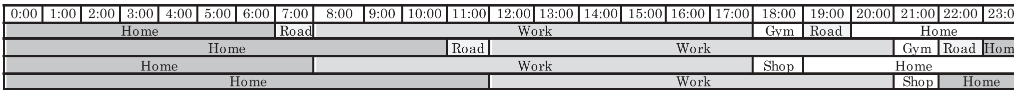


Fig. 5. Time-shifting example.

LCSS Distances				
	User1	User2	User3	User4
User1	0	0.17	0.12	0.25
User2	0.17	0	0.25	0.12
User3	0.12	0.25	0	0.17
User4	0.25	0.12	0.17	0

JS Distances				
	User1	User2	User3	User4
User1	0	0	0.12	0.13
User2	0	0	0.12	0.12
User3	0.12	0.12	0	0
User4	0.13	0.12	0	0

Fig. 6. Pair-wise distances based on the LCSS and JS metrics for the time-shifting example in Figure 3.

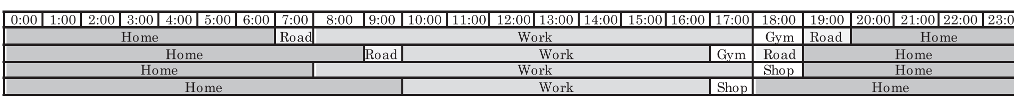


Fig. 7. Decreasing-work-time example.

Recall from our first assumption that we demand that the used similarity metrics be able to define users as similar even if their calendars do not match exactly over time. In other words, the used metric in such a stochastic case should be robust enough and the sensitivity to temporal changes should not be too high. Therefore, for illustration purposes, in Figure 5 we *shift* the mobility behaviors of user 2 and user 4 forward by 3 hours, while keeping all the rest identical to Figure 3.

Thus, according to the first assumption, a robust similarity metric should still place users 1 and 2 in the same cluster and users 3 and 4 in another cluster. Figure 6 shows the LCSS and the JS distance matrices following the considered time-shifts.

This small example shows one of the most important properties of the JS distance: it is less sensitive to the exact time at which each activity occurs if the mobility patterns over the locations are stable or, in other words, it is relatively robust over time. In contrast, the LCSS metric is sensitive to time shifts and reverses the clustering order; now it finds users 1 and 3 to be similar, as well as users 2 and 4.

The second property that we examine is the proposed metric's sensitivity to activity length. If, for example, one finds two users to be similar because they both go shopping after work and then return home, it should be relatively robust to the scenario in which one of them works slightly less/more than the other. In the following illustrative example, we decrease the work times of User 2 and User 4 by 2 hours, as shown in Figure 7.

Accordingly, we expect that a robust metric will not change the relative distances among users following a minor change in the length of a long activity.

Figure 8 shows the distance matrices of both the LCSS and JS distances following the activity length change. Once again, the JS distance is more robust than the LCSS measure, namely, in this case, the LCSS distance does not indicate an exclusive similarity between the first two users and the last two users. In contrast to the LCSS metric, the JS distance is relatively less sensitive to the change in the activities' lengths—the distance values of the JS distance matrix change, but

LCSS Distances				
	User1	User2	User3	User4
User1	0	0.12	0.12	0.25
User2	0.12	0	0.17	0.12
User3	0.12	0.25	0	0.12
User4	0.25	0.12	0.12	0

JS Distances				
	User1	User2	User3	User4
User1	0	0.01	0.12	0.14
User2	0.01	0	0.13	0.12
User3	0.12	0.13	0	0.01
User4	0.14	0.12	0.01	0

Fig. 8. Pair-wise distances of the decreasing work-time example.

by relatively small amounts, such that the final clustering is not affected: users 1 and 2 are still clustered together, as are users 3 and 4.

4 CLUSTERING EVALUATION

This experimental section applies unsupervised learning to an untagged real-world dataset of human mobility patterns that does not contain the lifestyle labels of the users. An important objective of the study is to find "interesting" clusters in the data, including ones that were unknown prior to the clustering process. Accordingly, when validating the obtained clusters, we use internal validity indices. Those indices evaluate the clustering based on the distribution of the entities within the clusters rather than the distribution of the labels among the clusters (external validity indices).

The used clustering indices are divided into two main classes:

(1) Metric-dependent indices: known clustering indices that are used to evaluate the clustering algorithm given a distance metric (LCSS or JS).
(2) Metric-independent index: a newly proposed index that does not depend on any metric but rather on the mobility data patterns themselves.

Both types of indices are discussed next.

4.1 Metric-dependent Indices

It is known that there is no single clustering measure that outperforms all the other measures. Therefore, in many cases, several measures are used together. For example, see [53, 33], in which different clustering measures are discussed. Accordingly, in this study, we use three known measures that are commonly utilized in the clustering literature: the Silhouette index [39], the Dunn index [14], and the Connectivity index [24]. All these indices can be considered different measures of the compactness and the separation of the clusters. These internal validity indices, which are metric-dependent, are applied to evaluate the obtained clusters by using either the JS or the LCSS metric. They are further compared in the evaluation section by using the Hamming distance measure, as explained below.

4.2 Metric-Independent Indices

There are two common challenges when using metric-dependent indices to evaluate clustering results. The first is that these indices do not indicate the suitability and "correctness" of the clustering output, but rather whether the obtained clusters form a reasonable shape in the space. The second challenge is that each of these indices is computed based on a specific metric, thereby making comparison of the different clustering algorithms impossible. Moreover, even if one selects a single distance metric for all the indices, it is often unclear whether this metric is suitable for the considered clustering application. In fact, it has been shown that different metrics can result in opposite clustering ranks, even when the same clustering algorithms are executed [37]. Therefore, a new index measure is proposed here that does not rely on a specific distance measure but rather is based on the mobility data patterns themselves.

4.2.1 Proposed Entropy Clustering Measure. Entropy is viewed as a measure of the uncertainty in the data. The entropy measure can be considered an external clustering evaluation that measures the purity of the clusters' class labels [12, 37, 38, 40]. Thus, if all clusters consist of objects with a single class label, the entropy will be zero. However, when the class labels of the objects in a cluster vary, the entropy increases. To compute the entropy of a dataset, the class distribution of the objects in each cluster must be computed as follows:

$$H(C_k) = \sum_{i \in I_k} p_{ik} \log\left(\frac{1}{p_{ik}}\right),$$

where $H(C_k)$ is the entropy of the kth cluster, p_{ik} is the probability of the ith class label (i.e., location) in the kth cluster, while the sum is taken over all the class labels (all locations) that appear in that cluster and belong to the set I_k. Accordingly, the total entropy for a set of clusters is calculated as the weighted average of the entropies of all clusters, i.e.,

$$H(C) = \sum_{k=1}^{K} \frac{n_k}{n} H(C_k),$$

where n_k is the total number of data points in cluster k, K is the number of clusters, and n is the total number of data points in the analyzed dataset over all the clusters. In practice, small clusters and outliers can be neglected when computing this measure. However, a different measure should be used when one would like to measure the clustering purity over unsupervised datasets, as the goal is to find new and interesting clusters in the data. Therefore, one needs to measure the entropy based on the data patterns themselves rather than any given class label that is not part of the given dataset.

Shannon [40] himself considered the above challenge by proposing a model for measuring the entropy of the English language. He first evaluated the frequencies of every letter, pair of letters, triple of letters, and so on. Given the frequencies of letters, it is possible to measure the entropy of the population for letter sequences of any size. As one increases the length of the sequence, the entropy often grows since there are more combinations of sequence types. However, if the text is not random, one often observes interesting reoccurring patterns in the text. For example, it is known that for an English-language text, given the letter "*q*," the probability that the next letter is "*u*" is higher than the probability that it is any other letter.

We use the above concept to measure the entropy of a dataset that represents a sequence of semantic locations, each of which can be considered a symbol in the location alphabet. The first-order location entropy is simply the entropy of single locations in the cluster. Thus, it is based on the marginal probability of all the locations, regardless of the sequence and time. To measure the location entropy in the clusters, the first step is to slice each trajectory into its locations, as illustrated in Figure 9.

Once the trajectories have been separated into their building-block locations, each cluster can be represented by the distribution over the locations from all its trajectories. The entropy of the kth cluster, which is denoted by $H(C_k)$, is calculated as indicated above. A higher entropy value implies that the probability distribution is more equiprobable with respect to the set of locations. Thus, the associated users spend their total time more equally over the different locations. Since the entropy measure is aggregated over all the users and all their trajectories, the probability being evenly distributed over the locations does not necessarily imply that all the users are similar, nor that each of them spends equal time in each location. Theoretically, the minimum value of the

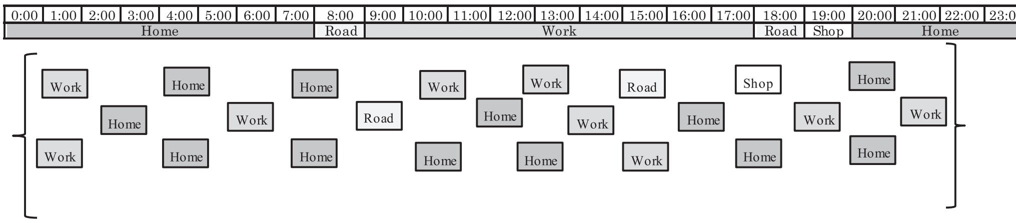


Fig. 9. Separating a sequence trajectory into its locations.

entropy is zero, which is obtained only when the users in the cluster stay in the same location the whole time. In practice, we notice that the entropy value increases as more users and trajectories are added to the cluster since the distributions over the different locations gradually become more even. This is why the entropy decreases as the number of clusters increases, since fewer more-similar users can be clustered together, separately from other more-different users.

In this section, we formalize our notation, assuming that each user is represented by an hourly location trajectory, as shown in Figure 7, and these trajectories are the elements that should be clustered. Let us denote by n^j the number of time-step (i.e., hourly) locations in T_j, which is the trajectory of user j that represents his mobility behavior during the analyzed time period. We assume that all trajectories are of the same length. Hence, all n^j are equal. For cluster C_k, n_k denotes the total number of time-step locations that are associated with the cluster, i.e., $n_k = |C_k| \cdot n^j$, where $|C_k|$ represents the number of users or time-step sequences in the cluster. Finally, n is the total number of time-step locations in the data, i.e., $n = \sum_{k=1}^{K} n_k$, assuming there are K clusters. We denote by $N_{i,j}$ the number of appearances of location i in the trajectory of user $j \cdot N_{i,k}$ is the total number of appearances of location i in cluster k, and N_i is the total number of appearances of location i in the data. The entropy of the data without any clustering is denoted as $H(X)$ and can be calculated by

$$H(X) = -\sum_i \frac{N_i}{n} \log \frac{N_i}{n}.$$

The entropy of clustering C is calculated by the weighted sum of the clusters' entropy:

$$H(C) = \sum_k \frac{n_k}{n} H(C_k) = -\sum_k \frac{n_k}{n} \sum_i \frac{N_{i,k}}{n_k} \log \frac{N_{i,k}}{n_k}.$$

Note that the clustering itself, as seen below, reduces the entropy by definition. This is of importance since it reduces the uncertainty related to mobility patterns in large geographic areas.

Proposition 4.1. *$H(X) \geq H(C)$ for every clustering C.*

Proof (based on [27]).

$$\begin{aligned} H(C) &= -\sum_k \frac{n_k}{n} \sum_i \frac{N_{i,k}}{n_k} \log \frac{N_{i,k}}{n_k} = -\sum_i \sum_k \frac{N_{i,k}}{n} \log \frac{N_{i,k}}{n_k}. \\ &\leq -\sum_i \left(\frac{\sum_k N_{i,k}}{n} \right) \log \frac{\sum_k N_{i,k}}{\sum_k n_k} = -\sum_i \frac{N_i}{n} \log \frac{N_i}{n} = H(X). \end{aligned}$$

The inequality follows from the log-sum inequality [12].

The location entropy measures how well a clusterinalgorithm can group users, sharing the same location distribution, i.e., the users who spend similar amounts of time at the same locations regardless of the patterns or the times of day. Therefore, lower entropy for a specific clustering

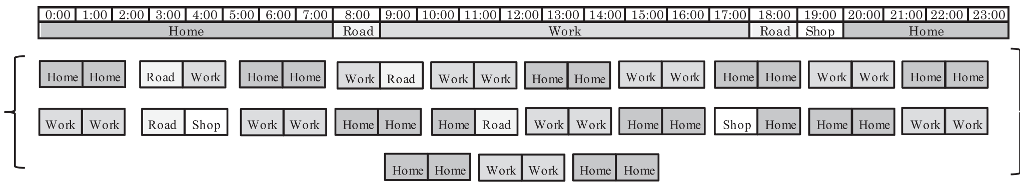


Fig. 10. Separating a trajectory into its pairs of successive locations.

implies that the trajectories of users in each cluster are more similar to one another. Accordingly, we aim to show that the sum of the KL divergences of every trajectory distribution T_j with respect to its cluster distribution, with $j \in C_k$, is proportional to the clustering location entropy.

Proposition 4.2. *Lower location entropy results in a lower sum of KL divergences between every trajectory and its cluster distributions.*

Proof.

$$\sum_k \sum_{j \in C_k} D_{KL}(T_j||C_k) = \sum_k \sum_{j \in C_k} \sum_i \frac{N_{i,j}}{n_j} \log \frac{\frac{N_{i,j}}{n_j}}{\frac{N_{i,k}}{n_k}}$$

$$= \sum_k \sum_{j \in C_k} \sum_i \frac{N_{i,j}}{n_j} \log \frac{N_{i,j}}{n_j} - \sum_k \sum_{j \in C_k} \sum_i \frac{N_{i,j}}{n_j} \log \frac{N_{i,k}}{n_k}$$

$$= - \sum_k \sum_{j \in C_k} H\left(T_j\right) - \sum_k \sum_i \left(\frac{\sum_{j \in C_k} N_{i,j}}{n_j}\right) \log \frac{N_{i,k}}{n_k}$$

$$= - \sum_j H\left(T_j\right) - \sum_k \sum_i \frac{N_{i,k}}{n_j} \log \frac{N_{i,k}}{n_k} = - \sum_j H\left(T_j\right) - \sum_k |C_k| \sum_i \frac{N_{i,k}}{n_j\,|C_k|} \log \frac{N_{i,k}}{n_k}$$

$$= - \sum_j H\left(T_j\right) - \sum_k |C_k| \sum_i \frac{N_{i,k}}{n_k} \log \frac{N_{i,k}}{n_k} = - \sum_j H\left(T_j\right) + \sum_k |C_k|\, H\left(C_k\right)$$

$$= \sum_k \frac{n_k}{n_j} H\left(C_k\right) - \sum_j H\left(T_j\right) = \frac{n}{n_j} \sum_k \frac{n_k}{n} H\left(C_k\right) - \sum_j H\left(T_j\right) = \frac{n}{n_j} H\left(C\right) - \sum_j H\left(T_j\right). \qquad \square$$

$\sum_j H(T_j)$ is the sum of the entropies of all the trajectories in the data. It is defined by the data and does not rely on any clustering. The value of $\frac{n}{n_j}$ is also constant and does not depend on the clustering. Thus, one is left with $H(C)$, which is the location entropy of the clustering and the only value that can affect and decrease the sum of the KL divergences. The KL divergence between distribution T_j and its cluster distribution, when $j \in C_k$, is the amount of information that is lost when one approximates the distribution T_j by the distribution C_k. Thus, a lower location entropy of the clusters implies that the KL divergence is lower, and a more homogeneous clustering is obtained. This is exactly the objective of the proposed clustering procedure.

Higher order pattern entropy. The location entropy can be considered a "zero-order" Markov measure (i.e., without memory) that indicates whether the clusters contain users with similar distributions over the same semantic locations. One can use a similar approach to measure the entropy of patterns of higher order Markov models in the resulting clusters. The concept is similar, with one main difference that each trajectory is now divided not into single locations but into a set of pairs of successive locations, as seen in Figure 10.

By measuring the entropy of location pairs in each cluster, one can learn whether the clusters contain users that not only have similar time distributions over the same locations but also have similar conditional distributions over the same locations, which indicates that they follow similar location patterns. Note that in a similar way, one can measure higher order patterns by using the entropies of triplets, quadruplets, and quintuplets, and so on, of locations.

5 REAL DATA RESULTS

5.1 Data

Our evaluation and analysis were conducted on real-life dataset of a mobile phone provider. This section describes the process of extracting the raw data and transforming them into semantic trajectories of users. The raw data contain the mobility data of 50,000 anonymous cellular users over a period of two months. The users' data was sampled from Radio Network Controllers (RNCs) that cover a relatively large city and its surrounding area and contain hundreds of thousands of users. In the first stage of the study, from security and regulation reasons, we were allowed to track and model the individual mobility patterns of 10% of the users (5,000 users) and segment them accordingly. However, in the second stage of the study, we were allowed to associate the entire set of extracted users with segments that were learned in the first stage of the study. Thus, providing an analysis on a larger scale of a neighborhood or a small city.

The first step of data preprocessing was to group single records into a session record that contains features such as the global index, date, start time, end time, X coordinate, and Y coordinate [11].

In the next step, locations were clustered into tagged "important places" in the user's life by their geographic location and time indices. This step is important for filtering noise from the geographical sample, as well as to reduce the resolution and help assign semantic meanings to the locations in the next step.

The third step assigns semantic meanings to various recorded places. Since there was no use of external data, the sematic tag indicates the frequency of each location for the user in descending order. Thus, each location was ranked by the amount of aggregated time (normalized by the total recorded time) in the schedule. The most frequent location was assigned the label "A," the second one "B," and so on, up to the letter "E." Letter "F" was used to indicate the situation of no signal at a specific hour.

Following such a coding system, the geographical location of "A" for one user is different than the location of "A" for another user. Similarly, the semantic meanings of each label are not necessarily equal for different users—for one user, "A" can indicate "home," whereas for another user "A" can represent "work" (if he spends more time at work than at home, which is in fact quite rare in the given dataset).

We used two months of data that were separated between weekdays and weekends to increase the probability of identifying repetitive patterns in the schedule.

The first analysis that was performed on the numerical data was to reduce its dimensionality by using PCA as proposed in [15]. The analysis revealed that over 50% of the variance in the data can be explained by the first six principle components (PCs), as shown in Figure 11. In fact, the first PC explains 23% of the variance, the second PC accounts for 14%, the third and fourth PCs account for 4.5%, and so on, dropping to a long tail of non-significant PCs.

Some of the properties in the data could be revealed by using the obtained PCs.

For example, Figure 12 shows a sample of users ordered by their first PC value. Every row represents a single user while every column represents an hour in weekdays (Sunday to Thursday). The most frequent location is colored by dark brown, the second frequent location is colored by light brown, and so on. One can see that the first rows (with PC values around −1.76) contain a set of users that stay most of their time at home, while the second set of users spend more time

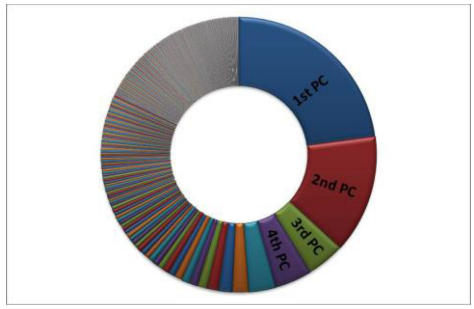


Fig. 11. The explained variance ratio of principle components based on mobility data of 5,000 anonymous cellular users.

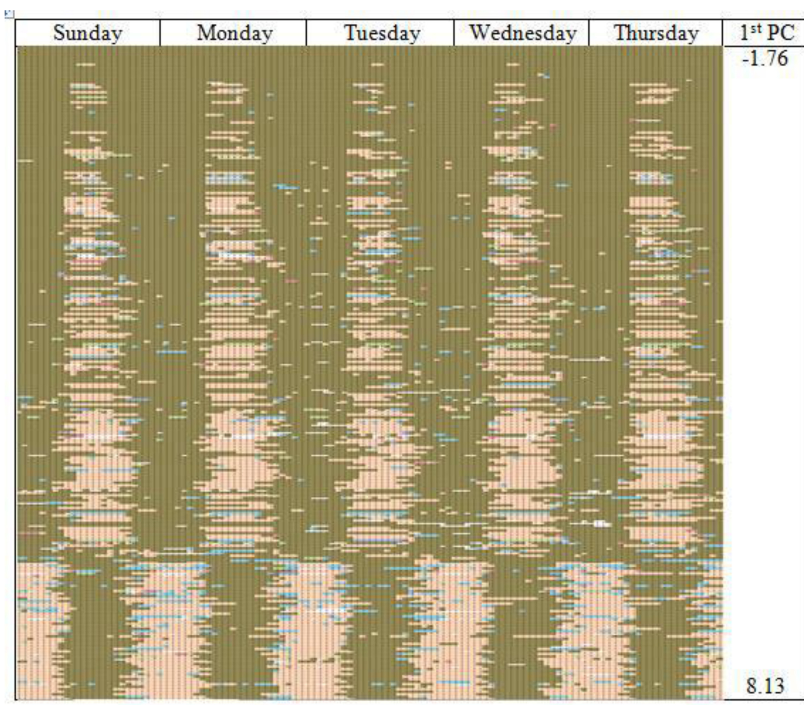


Fig. 12. A sample of the data organized in the range of the first PCs value.

in their work place. Finally, as one continues to users with higher PC values (7 and above) he can find users with inverse colors, implying that they spend more time at work than at home. Thus, one can interpret the first PC as "Time spent at work," since low values in this PC dimension mean that the user spends shorter time of the day at his work place.

Even though the first PC represents a core mobility behavior, using the PCs to cluster the users was found to be impractical in real dataset, specifically when trying to obtain clusters that represent different users' mobility behavior. Figure 13 shows the mobility data over the dimensions of the first and the second PCs. Note that the first PC has a large variance, yet it is not trivial to separate the data into natural clusters based on it alone. The second PC contain some variance but most of the data-points have a low value in this dimension. The figure shows 2 clusters (red

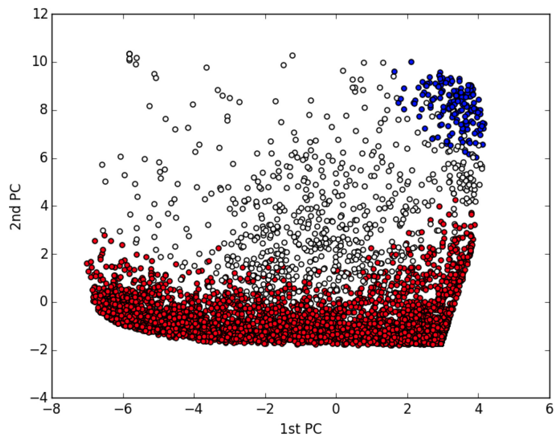


Fig. 13. A scatter of 5,000 users over the first and second PCs dimensions.

and blue) that were extracted by the DBSCAN algorithm, while the white points were labeled by the algorithm as noise. Note that the DBSCAN algorithm is based on data density and, therefore, it allows flexible clustering shapes as well as the presence of noisy data. Despite this flexibility, it was hard to find a natural interpretable clustering based on this dimension-reduction approach. The fact that this approach is inhomogeneous, where every time-interval is treated as an independent feature makes it even harder for such an analysis. Similar challenges were obtained when using other PCs for the analysis, making it difficult to cluster the data accordingly. Note that if the objective was to learn a single user behavior, as done in [15], such a data reduction method can lead to excellent results. But for learning and clustering thousands users or more, the variance of the temporal-based patterns in real-world data is simply too high, resulting in PCs that are very hard to interpret as seen above.

5.2 Evaluation of the Clustering Procedures

In this section, we apply the LBC process with three state-of-the-art clustering methods. Thus, four clustering methods are considered:

(1) *LBC clustering:* This is the proposed LBC, as presented in Section 3.
(2) *LCSS clustering:* This process is similar to the LBC procedure, except that it uses the LCSS distance metric on the deterministic trajectories instead of the JS distance over stochastic Markov models. We applied a similar analysis to the LCSS distance to select the best clustering algorithm and found that Agglomerative clustering [41, 23, 21] is the most suitable method for the considered data.
(3) *PCA clustering:* In this process, we use the PCA method to perform dimensionality reduction on our data. We used a Euclidean distance for the computation of the pair-wise distance matrix and then executed a clustering algorithm. Again, we found the Agglomerative clustering to obtain the best results. The number of PCs was chosen based on the described evaluation indices. The best results were obtained for seven PCs.
(4) *EM clustering:* In this process, we assume that each trajectory is a multidimensional point in space that is produced from a Mixture Model that represents its cluster. To find the clusters, an Expectation-Maximization algorithm was used following [34].

The evaluation and comparison of the four clustering procedures are based on three aspects of mobility behavior as follows:

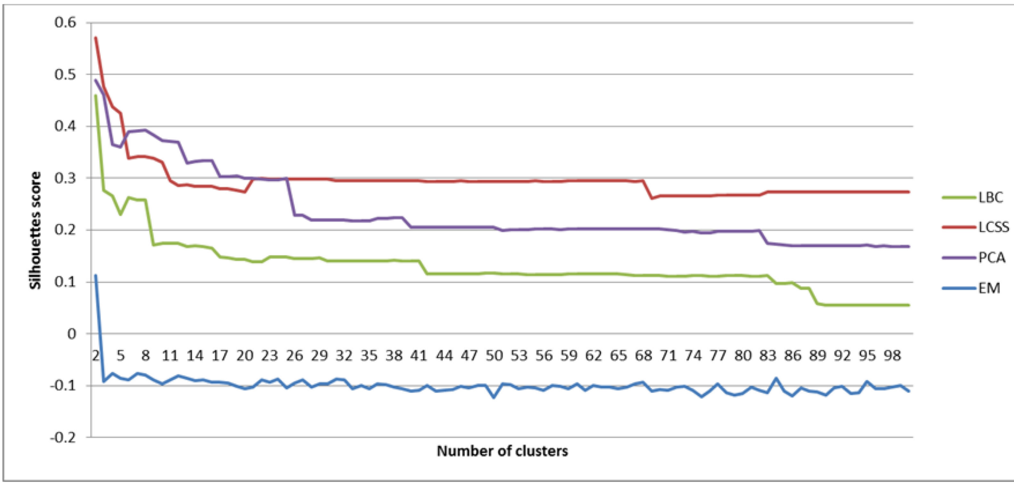


Fig. 14. Silhouettes scores of the four clustering procedures over the Hamming-distance space. The *x*-axis represents the number of clusters and the *y*-axis represents the Silhouettes scores.

(1) Users from the same cluster go to the same locations at the same time;
(2) Users from the same cluster have similar location distributions;
(3) Users from the same cluster share the same patterns among locations.

These three measures are correlated with our initial assumptions and each is related to a different set of applications, as discussed next.

Users who share the same locations at the same time. To evaluate whether the applied clustering approach can cluster users who are present at the same location at the same time, we measure the obtained clusters with metric-dependent indices. In particular, the Hamming distance [20] is used as a metric that simply counts the number of hours in which the two mobility sequences of two different users do not match. Each mobility pattern vector represents the location type of an associated user at a certain hour on a specific date (see Figure 2). The entry-wise Hamming distance value of "0" implies that the two users share the same location type (e.g., "home" or "work") at the same time. Hence, a perfect matching between the two sequences results in a Hamming distance value of zero. The goal is to measure the pair-wise Hamming distance for every pair of users, thereby resulting in a "Hamming space" over all users. In this space, users are distant from one another when they have un-matched locations over time.

To further measure the compactness and separation of the obtained clusters, we use metric-dependent validity indices (e.g., Silhouettes, Dunn, and Connectivity) over the Hamming distance matrix.

Figure 14 shows the Silhouettes scores of the four clustering procedures for every chosen number of clusters, from 2 to 100. The results show that the LCSS clustering procedure obtains a superior Silhouettes score for most numbers of clusters; the PCA process has the second-best scores, the LBC process is third and the EM algorithm obtains the poorest result for this objective. The results imply that when the application requires users that share the same locations at the same times to be clustered together, for example, in carpool applications, a wise choice of clustering algorithm will be based on the LCSS or the PCA method, depending on the number of required clusters as proposed in the literature.

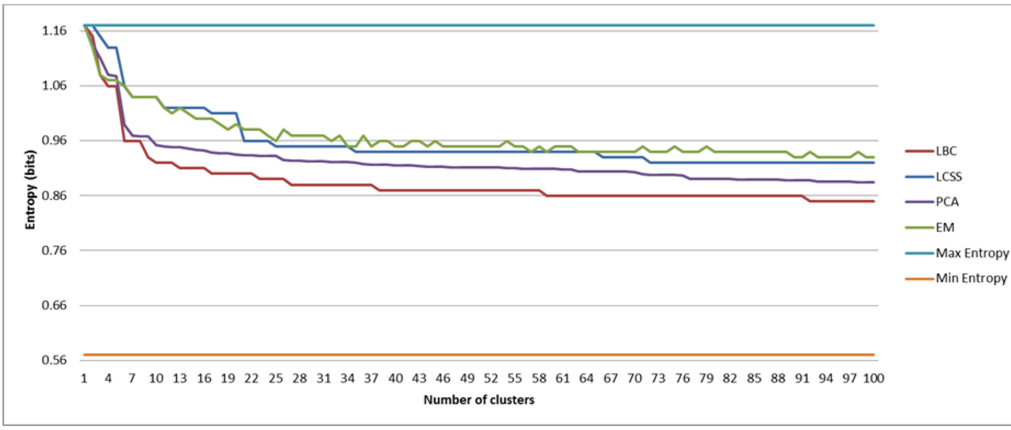


Fig. 15. Location entropies of the four clustering procedures. The x-axis represents the number of clusters and the y-axis represents the location entropy measure.

Users who have similar location distributions. In this section, we use the proposed location entropy as our clustering measure for the same four clustering procedures. The maximum location entropy of the data is found to be 1.17 bits (measured when all users are grouped in a single cluster). As we increase the number of clusters, the entropy of the population decreases, while the minimum entropy is reached when every user is in his own cluster and is equal to 0.57 bits (this is the average entropy of the users' trajectories in the population). The maximum and minimum entropies are shown in Figure 15 by the upper and lower straight lines, respectively.

According to the location entropy measure, the proposed LBC procedure clusters the users in the best way with respect to the location distribution. The next-best procedure is the PCA, followed by the LCSS and the EM, which obtain similar results with a small advantage to the LCSS procedure. These results imply that when the application requires users that share the same location distribution to be clustered together, for example, in location-based mobile coupon planning, a wise choice of clustering procedure will be based on the proposed LBC clustering.

Users that share the same patterns among locations. To learn more about the mobility patterns of the users, we apply a higher order pattern entropy measure for the same clustering procedures. In particular, we consider the pattern entropy of pairs of successive locations in the data, as shown in Figure 16.

The same hierarchy that was obtained in the zero-order location entropy graph appears in the pattern entropy graph of paired locations. The only difference is that the gap between the proposed LBC and the other clustering procedures is slightly wider and more significant since the LBC captures well the pair patterns. Similar results were obtained for location triples.

A T-test was performed over the results and found that the entropy of the LBC procedure is significantly lower than the entropy obtained from other procedures when using series of 2, 3, 4, 5, and 6 time intervals, as shown in Table 1.

A trivial extension of the proposed LBC approach is a replacement of the first-order Markov model by a higher order model, e.g., a second- or third-order Markov models that can potentially capture higher order (longer) patterns. Note that such an extension has several limitations. First, it increases the computational complexity, particularly during the construction of the JS distances matrix. Second, the data contains fewer longer patterns as a support set per each tuple of locations that are represented by the higher order Markov models. Third, it was found in the experiment

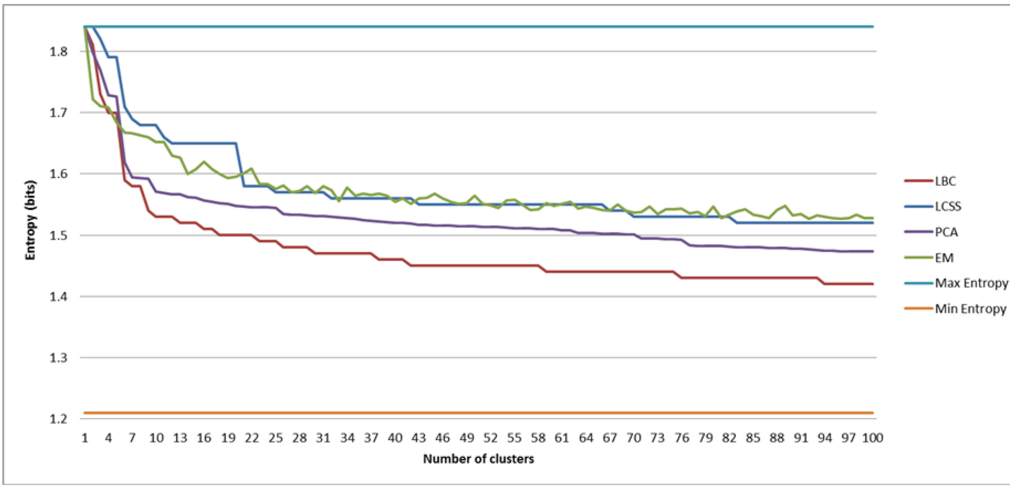


Fig. 16. Patterns' entropy of the four clustering procedures. The *x*-axis represents the number of clusters and the *y*-axis represents the pair-location entropy measure.

Table 1. *P*-values of *T*-tests Between the LBC Entropy Scores and the Other Methods

	Entropy type	LCSS	PCA	EM
Location entropy	1	1.9E-64	2.3E-60	2.0E-62
Pattern entropy	2	1.5E-72	3.5E-62	1.2E-54
	3	1.9E-75	1.1E-59	4.3E-60
	4	1.3E-76	1.6E-57	4.1E-59
	5	9.8E-78	2.5E-56	4.8E-58
	6	4.1E-48	1.2E-66	3.1E-88

The "entropy type" column denotes the length of the time interval that is used to measure the entropy.

that, in fact, the first-order Markov model outperforms the higher order Markov models that are often overfitted, as seen in Figure 17. The figure shows that the patterns' entropy of an LBC-based clustering procedure increases when using higher order Markov models to represent each user.

Note, however, that despite the above, in many cases higher order patterns of three or four semantic locations appear in the data, although they appear in specific time periods. To address the order-variability, while enabling a higher order model only for specific patterns, we suggest to apply the VOBN [8] that was proposed originally for DNA analysis and was never applied to mobility pattern analysis. In contrast to the simple Markov model, where the memory-order is fixed to one and identical for all the locations, over all the users, all the hours, and all the days, in VOBN modeling, the memory order can vary for each location, based on the learned patterns.

Finally, in a series of experiments different clustering methods were compared in order to select the most suitable algorithm for mobility pattern clustering. Several clustering methods, including the *K*-medoids, the Spectral clustering and the Agglomerative clustering were evaluated by several internal validity indices, including the Connectivity index the Dunn index, the Silhouettes index, and the Pattern–Entropy index, while ranging the number of used clusters between 2 to 100. In vast majority of the cases it was found that the Agglomerative clustering outperforms the other

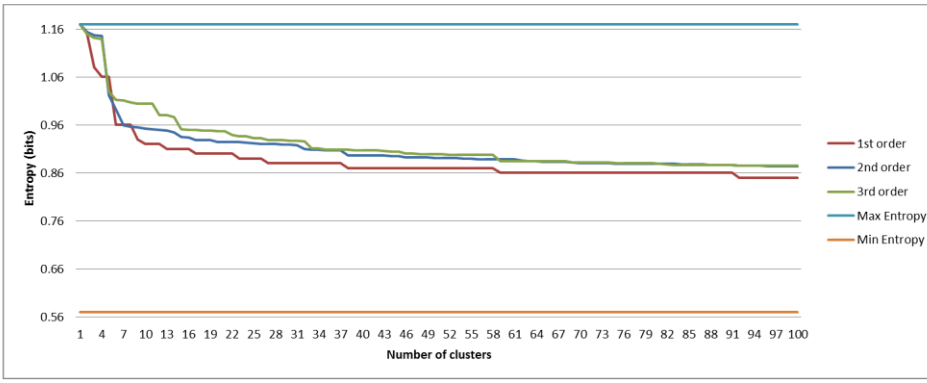


Fig. 17. Patterns' entropy of four LBC clustering procedures. The x-axis represents the number of clusters and the y-axis represents the pair-location entropy measure.

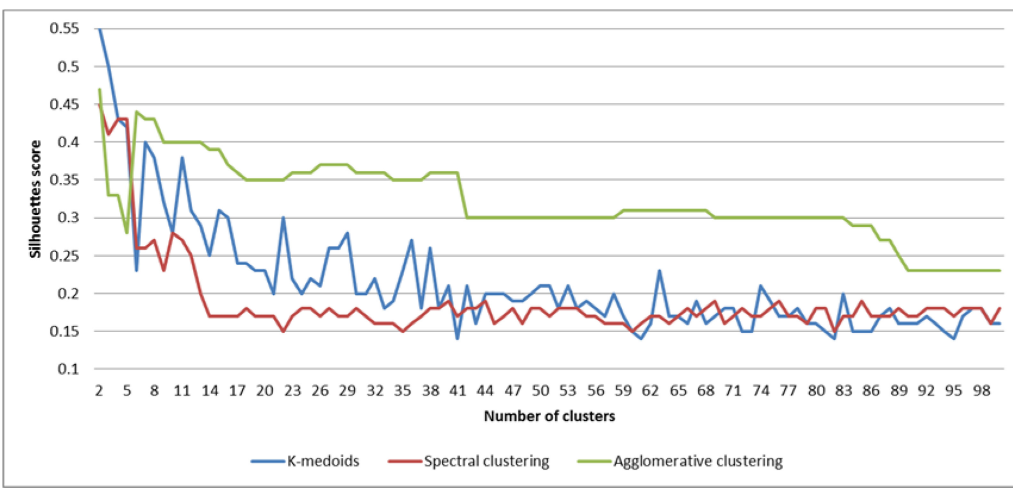


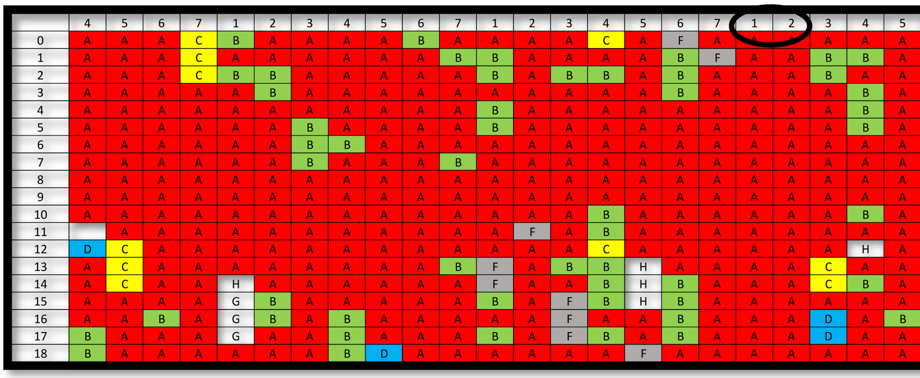
Fig. 18. Comparison of three clustering algorithms, namely the K-medoids, the Spectral clustering and the Agglomerative clustering. The x-axis represents the number of clusters and the y-axis represents the Silhouettes index scores. As seen in majority of the experiments the Agglomerative clustering outperforms the other methods.

clustering methods. Figure 18 presents for example three clustering methods with respect to the Silhouettes index score.

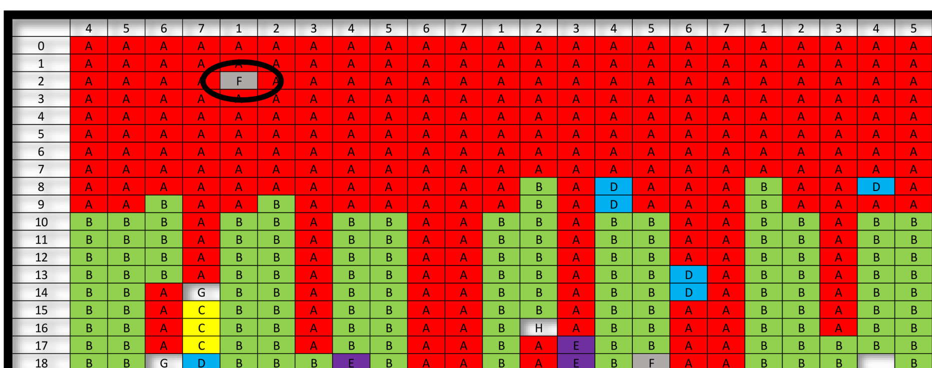
5.3 Examples of Pattern-Based Segmentation

The use of the proposed LBC method enables, as indicated above, to cluster different users that share lifestyle similarity. This segmentation can be executed over different geographic locations since their labeling is probabilistic and semantic, i.e., labels "A," "B," "C," and so on are assigned by their frequencies in decreasing order.

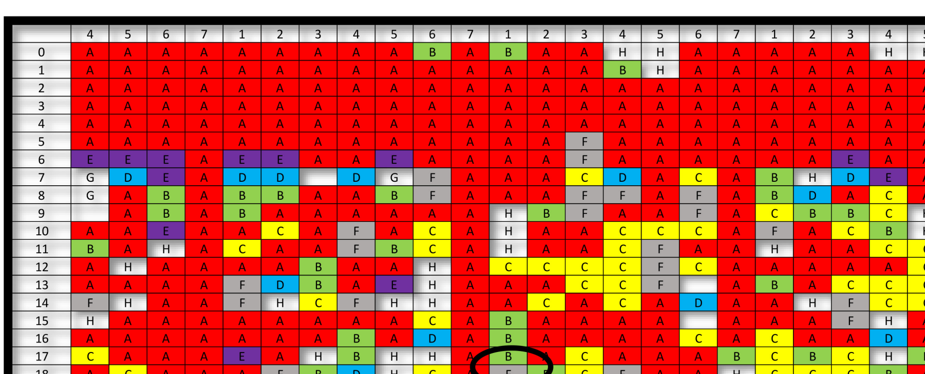
Figures 19(a)–19(d) show for example four users, each of which from a different segment. Each row in the figure represents the hour during the day ("0" stands for midnight down to "18" that stands for 6pm), each column represents a day ("1" stands for a Sunday up to "7" that stands for a Saturday). As seen, each of those segments represent different mobility patterns, some



(a) "Home, sweet home" pattern

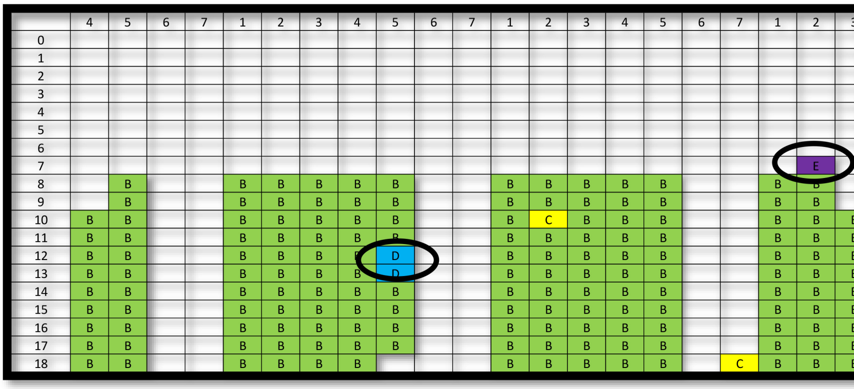


(b) "Working 9 to 5" pattern



(c) "Travelling salesman" pattern

Fig. 19. (a–d) Examples of mobility pattern's diaries of four different users, each of which belongs to a different mobility-behavior segment. Each row represents the hour during the day ("0" stands for midnight down to "18" that stands for 6pm), each column represents a day ("1" stands for a Sunday up to "7" that stands for a Saturday). Each entry denotes a semantic location, labeled by the letter "A" up to "F" in decreasing order of frequency.



(d) "Commuter" ('lives out of the city') pattern

Fig. 19. Continued.

quite different from each other. We name each segment by an intuitive name that provide some insight for a possible qualitative description of its users. For example, the pattern exemplified in Figure 19(a) is named "*Home, sweet home*" and shows a user that rarely leaves his home and when he does so, it is usually in early afternoon hours or very late at night. Figure 19(b) is named by the song "*Working 9 to 5*" although it shows a user who often stays at work between 10am to 6pm (with exceptions). Figure 19(c) is named "*Travelling salesman*" and shows a user who travels during long working hours over various locations. Finally, Figure 19(d) is named "*Commuter*" (or "lives out of the city") and shows a user who lives outside the metropolitan (and this is the reason that many of the entries are blanks—as location data was not recorded outside the city area), works close to the city borderline (hence, no commuting is detected within the city area) and starts during weekdays sharply at 8am until 7pm.

Note that many of the analyzed mobility-patterns are not trivial nor entirely consistent. In each class, one can detect repeating mobility-patterns as well as some anomalies (marked by a black ellipse). Just to show a few, in Figure 19(a), on the third Sunday–Monday sequence the user stays at home along the entire 2 days, which points to a rare mobility-pattern that can indicate, for example, that he does not feel well or that the phone was forgotten at home. In Figure 19(b), one can see that the user often stays at home on Tuesdays, plus an anomaly is detected on the first Sunday at 2pm. In Figure 19(c), location F usually follows location C or A, while in the marked Sunday at 6pm it follows the B location. Finally, in Figure 19(d) a very consistent mobility-behavior is detected for the commuter with some anomalies that are shown in the figure.

Figures 19(a)–19(d) show that the obtained mobility-behavior patterns are not trivial and that they vary significantly not only between users or between different clusters of users but also within a specific user itself. Note again that these mobility-patterns when aggregated over hundreds of thousands of users can provide important information, for example, in a framework of smart cities. Knowing, for example, which or how many users commute every morning to work and from where; which users or how many users stay at home and till when; and what is an anomalous mobility behavior for each user, can provide critical insights to personal security systems, ridesharing plans, smart-grid and energy consumption optimization, location-based services, traffic lights schemes, and many more data-driven applications.

Based on a sampling process of 50,000 users, Figure 20 shows the distribution of different segments of users, where each segment represents a different mobility behavior, over five different cities, while applying the VOBN extension [8]. As seen each city has a different distribution of

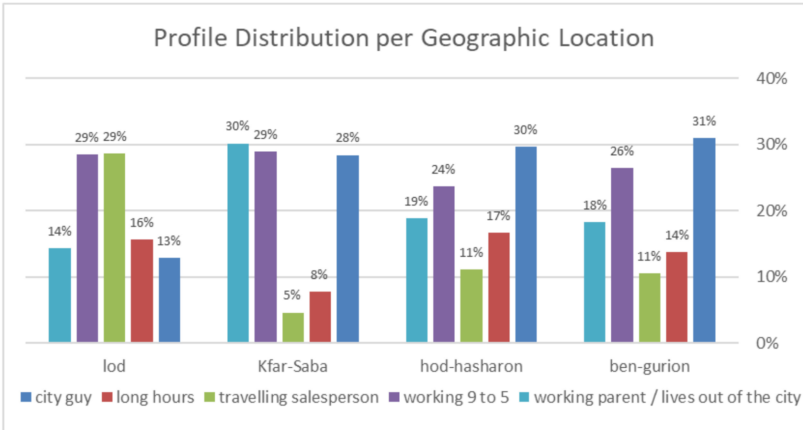


Fig. 20. Distribution of different mobility-behavior segments, over five different cities.

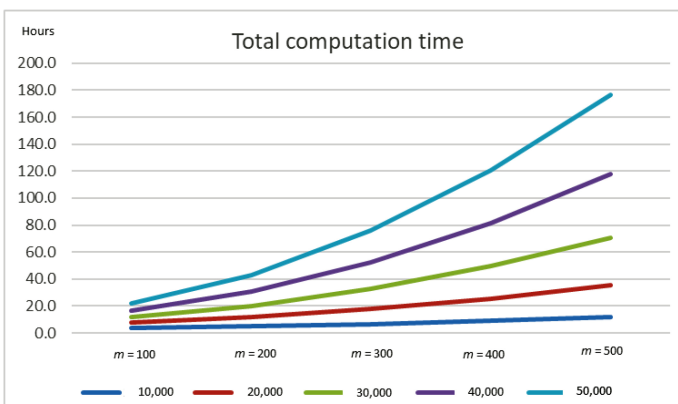


Fig. 21. Total computation time of the proposed procedure (in hours) as a function of both the sequence length $m = 100, \ldots, 500$ and the number of modeled users $J = 10,000, \ldots, 50,000$.

users. To give just one example, in Lod only 14% of the sampled users followed a commuting (lives out of the city) behavioral pattern, while in Kfar-Saba this segment is doubled and reach 30%. Such a distribution was found to be statistically significant with the socio-demographic score of the city, indicating that there is a noticeable correlation between the socio-demographic status of the users and their mobility-behavioral patterns [11]. Such an analysis emphasizes again that many "smart-city" services, such as electric-grid optimization, traffic lights control, road-maintenance working times, and public transportations can rely specifically on the citizens' mobility-behavior distributions in order to provide a tailored service kit per area.

Finally, Figure 21 shows the total computational time of the proposed procedure when it is applied to segments of users of different size, ranging from 10,000 models to 50,000 models, as a function of the location sequence size with $m = 100, \ldots, 500$ locations. Note that both the number of clusters, d, and the alphabet size l (i.e., the possible location labels) do not affect the computation time significantly under the considered settings, although they are critical parameters in terms of the clustering quality and the obtained statistics. As indicated in Section 3.2, the total computational time is combined from three main steps: the construction of a Markov model per each user; the pair-wise distance computation between all the user models resulting in the construction of the similarity matrix; and the clustering procedure itself over the similarity matrix. The construction of the similarity matrix is the most computationally expensive step in the proposed procedure that often consumes more than 85% of the total computation time. This bottleneck step is also required by the benchmark LCSS method that is computationally equivalent to the proposed procedure. Note that by applying parallel computation schemes as well as other computational reduction strategies, such as stochastic search, one can further reduce the computational time of the proposed method.

6 DISCUSSION

6.1 Conclusions

The exponential growth of available location-based data in recent years has resulted in increasing attention of researchers to this area. New research works focus on how to obtain, cleanse, preprocess, and store location data, modify it for different uses and eventually extract meaningful information out of it. The potential of such data in the analysis of human mobility patterns is significant since it can be used in many modern applications, such as smart transportation, location-based services, homeland security, and marketing localization. Many aspects of knowledge discovery from mobility data processes have been contributed in recent years by researchers from both the academic world as well as the industry.

From the literature review, one can identify a real need to develop a new metric that can capture the mobility behaviors of people in a manner that represent their “mobility lifestyles.” The objective of identifying similar people by their semantic-location patterns, regardless of their geographic locations and their specific activity times, become evident in many new applications. The required LBC modeling led to a new challenge regarding the calculation of the distance between mobility patterns. For this purpose, we have proposed the JS metric, which relies on the KL distance between the users’ Markovian models. Several clustering algorithms were examined under the assumption that there is no ground-truth “correct” clustering algorithm, but rather a clustering solution that fits the data well in a specific period.

An important contribution of this research is the proposed clustering procedures, which aim at finding groups of trajectories with similar location distributions and/or similar patterns. To evaluate the proposed LBC procedure, a new validation index was developed based on high-order entropy measures that satisfies the research objective. The proposed entropy measure relies on the data patterns themselves, as an unsupervised task, while measuring the homogeneity of the location distribution and the patterns in the data within the clusters. Results that were obtained from real data showed the efficiency and robustness of the LBC with respect to other clustering procedures, including those that were used in the literature for mobility pattern analysis. This advantage does not depend on a specific clustering algorithm or number of clusters.

6.2 Future Work

Several research directions can be considered when applying the proposed model to different data and/or different objectives.

First, the proposed model can be validated on larger scale datasets when available. Nonetheless, we show that the use of 5,000 randomly sampled users from a set of 50,000 users is enough for obtaining statistical validation of the model. Second, the use of semantic labeled locations rather than the user’s frequent locations appears to be promising for validation purposes and for obtaining a true sense of human mobility behavior. For validation purposes, we would like to cross the obtained results with demographic data, which can provide important information on the types of users in each cluster [11].

A natural extension of the proposed method is the generalization of the Markov model to a non-homogeneous VOBN [8]. This extension enables to model and extract non-homogenous and non-sequential patterns in the data, as discussed in Section 3.1.

Another interesting research direction is to use the LBC model on digital trajectory data that represent other aspects of human behavior; for example, clustering people by their web surfing histories or clustering people by their actions on their mobile phones.

Using human mobility data, several new analytic applications can be considered and addressed. One example is an application in which the exact times of the users' activities and the location patterns are used rather than the location matching over time. Another possible approach is to adopt a hybrid model that uses more than one clustering procedure. Finally, several modifications can be suggested for extending the LBC procedure to address real-life modeling challenges, such as further use of VOBN models [8] for profiling users instead of using the simple fixed-order Markov models.

REFERENCES

[1] Luis Otavio Alvares, Vania Bogorny, Bart Kuijpers, Bart Moelans, Jose Antonio Fernandes de Macedo, and Andrey Tietbohl Palma. 2007. Towards semantic trajectory knowledge discovery. Technical Report, Hasselt University, Belgium.

[2] Asier Aztiria, Alberto Izaguirre, and Juan Carlos Augusto. 2010. Learning patterns in ambient intelligence environments: A survey. *Artif. Intell. Rev.* 34, (2010), 35–51.

[3] Daniel Ashbrook and Thad Starner. 2003. Using GPS to learn significant locations and predict movement across multiple users. *Pers. Ubiquitous Comput.* 7, 5 (2003), 275–286. DOI: http://dx.doi.org/10.1007/s00779-003-0240-0

[4] Omer Barak, Gabriella Cohen, and Eran Toch. 2016. Anonymizing mobility data using semantic cloaking. *Pervasive Mobile Comput.* 28 (2016), 102–112.

[5] Michael Batty, Kay W. Axhausen, Fosca Giannotti, Alexey Pozdnoukhov, Armando Bazzani, Monica Wachowicz, Georgios Ouzounis, and Yuval Portugali. 2012. Smart cities of the future. *Eur. Phys. J. Spec. Top.* 214 (2012), 481–518.

[6] Paolo Bellavista, Axel Kupper, and Sumi Helal. 2008. Location-based services: Back to the future. *IEEE Pervasive Comput.* 7, 2 (2008), 85–89.

[7] Irad Ben-Gal, Gail Morag, and Armin Shmilovici. 2003. CSPC: A monitoring procedure for state dependent processes. *Technometrics* 45, 4 (2003) 293–311.

[8] Irad Ben-Gal, Ayala Shani, Andre Gohr, Jan Grau, Sigal Arviv, Armin Shmilovici, Stefan Posch, and Ivo Grosse. 2005. Identification of transcription factor binding sites with variable-order Bayesian networks. *Bioinformatics* 21, 11 (2005), 2657–66. DOI: http://dx.doi.org/10.1093/bioinformatics/bti410

[9] Chen Cheng, Haiqin Yang, Irwin King, and Mr Lyu. 2012. Fused matrix factorization with geographical and social influence in location-based social networks. In *Proc. AAAI.* 17–23.

[10] Darya Chudova, Scott Gaffney, Eric Mjolsness, and Padhraic Smyth. 2003. Translation-invariant mixture models for curve clustering. In *Proc. 9th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. (KDD'03).* 79–88. DOI: http://dx.doi.org/10.1145/956755.956763

[11] Gabriella Cohen, Eran Toch, and Irad Ben-Gal. 2017. *"Analyzing Mobility Patterns with Zero-Knowledge Routine Diaries."* Working paper based on unpublished thesis.

[12] Thomas M. Cover and Joy A. Thomas. 2012. *Elements of Information Theory.* John Wiley & Sons, Inc, 2012.

[13] William H. E. Day and Herbert Edelsbrunner. 1984. Efficient algorithms for agglomerative hierarchical clustering methods. *J. Classif.* 24 (1984), 7–24.

[14] Joseph C. Dunn. 1973. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *J. Cybern.* 3 (1973) 32–57. DOI: http://dx.doi.org/10.1080/01969727308546046

[15] Nathan Eagle and Alex Sandy Pentland. 2009. Eigenbehaviors: Identifying structure in routine. *Behav. Ecol. Sociobiol.* 63, 7 (2009) 1057–1066. DOI: http://dx.doi.org/10.1007/s00265-009-0739-0

[16] Dominik M. Endres and Johannes E. Schindelin. 2003. A new metric for probability distributions. *IEEE Trans. Inf. Theory* 49, 7 (2003), 1858–1860.

[17] Katayoun Farrahi and Daniel Gatica-Perez. 2012. Extracting mobile behavioral patterns with the distant n-gram topic model. In *Proc. 2012 16th Int. Symp. Wearable Comput.* 1–8. DOI: http://dx.doi.org/10.1109/ISWC.2012.20

[18] Laura Ferrari, Alberto Rosi, Marco Mamei, and Franco Zambonelli. 2011. Extracting urban patterns from location-based social networks. In *Proc. 3rd ACM SIGSPATIAL Int. Work. Locat. Soc. Netw. (LBSN'11)*. 9–16. DOI: http://dx.doi.org/10.1145/2063212.2063226

[19] Barbara Furletti, Paolo Cintia, Chiara Renso, and Laura Spinsanti. 2013. Inferring human activities from GPS tracks. In *Proc. 2nd ACM SIGKDD Int. Workshop Urban Comput.* Article 5. DOI: http://dx.doi.org/10.1145/2505821.2505830

[20] Richard W. Hamming. 1950. Error detecting and error correcting codes. *Bell Syst. Tech. J.* 29, 2 (1950), 147–160.

[21] George Karypis, Eui-Hong Han, and Han Vipin Kumar. 1999. Chameleon: A hierarchical clustering algorithm using dynamic modeling. *IEEE Comput.* 32, (8):68–75.

[22] Leonard Kaufman and Peter Rousseeuw. 1987. Clustering by means of medoids. *Stat. Data Anal. Based L 1-Norm Relat. Methods.* Y. Dodge (Ed.). Amsterdam, 405–416.

[23] Benjamin King. 1967. Step-wise clustering procedures. 1967. *J. Am. Stat. Assoc.* 69 (1967) 86–101.

[24] Joshua Damian Knowles and J. Handl. 2005. Exploiting the trade-off: The benefits of multiple objectives in data clustering. *Evolutionary Multi-Criterion Optimization*. Lecture Notes in Computer Science, Vol. 3410. 547–560.

[25] Solomon Kullback. *Information Theory and Statistics*. John Wiley & Sons, Inc, 1959.

[26] Kenneth Wai-ting Leung, Dik Lun Lee, and Wang-chien Lee. 2011. CLR: A collaborative location recommendation framework based on co-clustering categories and subject descriptors. In *Proc. 34th Int. ACM SIGIR Conf. Res. Dev. Inf. Retrieval (SIGIR'11)*. 305–314. DOI: http://dx.doi.org/10.1145/2009916.2009960

[27] Tao Li, Sheng Ma, and Mitsunori Ogihara. 2004. Entropy-based criterion in categorical clustering. In *Proc. 21st Int. Conf. Mach. Learn. (ICML'04)*. 68. DOI: http://dx.doi.org/10.1145/1015330.1015404

[28] Jianhua Lin. 1991. Divergence measures based on the Shannon entropy. *IEEE Trans. Inf. Theory* 37, 1 (1991), 145–151.

[29] Juhong Liu, Ouri Wolfson, and Huabei Yin. 2006. Extracting semantic location from outdoor positioning systems. *Proc. IEEE Int. Conf. Mob. Data Manag.* 2006 (2006), 1–8. DOI: http://dx.doi.org/10.1109/MDM.2006.87

[30] Eric Hsueh-Chan Lu and Vincent S. Tseng. 2009. Mining cluster-based mobile sequential patterns in location-based service environments. In *Proc. 2009 10th Int. Conf. Mob. Data Manag. Syst. Serv. Middlew.* 273–278. DOI: http://dx.doi.org/10.1109/MDM.2009.40

[31] Mingqi Lv, Ling Chen, and Gencai Chen. 2013. Mining user similarity based on routine activities. *Inf. Sci.* 236 (2013), 17–32. DOI: http://dx.doi.org/10.1016/j.ins.2013.02.050

[32] Haiping Ma, Huanhuan Cao, Qiang Yang, Enhong Chen, and Jilei Tian. 2012. A habit mining approach for discovering similar mobile users. In *Proc. 21st Int. Conf. World Wide Web (WWW'12)*. 231–340. DOI: http://dx.doi.org/10.1145/2187836.2187868

[33] Ujjwal Maulik and Sanghamitra Bandyopadhyay. 2002. Performance evaluation of some clustering algorithms and validity indices. *IEEE Trans. Pattern Anal. Mach. Intell.* 24, 12 (2002), 1650–1654.

[34] Todd K. Moon. 1996. The expectation-maximization algorithm. *IEEE Signal Process. Mag.* 13, 47–60, 1996.

[35] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. 2002. On spectral clustering: Analysis and an algorithm. *Adv. Neural Inf. Process. Syst.* 2 (2002), 849–856.

[36] Gang Pan, Guande Qi, Wangsheng Zhang, Shijian Li, Zhaohui Wu, and Laurence Tianruo Yang. 2013. Trace analysis and mining for smart cities: Issues, methods, and applications. *IEEE Commun. Mag.* 51, 6 (2013), 120–126.

[37] Ido Priness, Oded Maimon, and Irad Ben-Gal. 2007. Evaluation of gene-expression clustering via mutual information distance measure. *BMC Bioinform.* 8 (2007), 111. DOI: http://dx.doi.org/10.1186/1471-2105-8-111

[38] Eréndira Rendón, Itzel Abundez, Alejandra Arizmendi, and Elvia M. Quiroz. 2011. Internal versus External cluster validation indexes. *Int. J. Comput. Commun. Control* 5, 1 (2011), 27–34.

[39] Peter J. Rousseeuw. 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* 20 (1987), 53–65. DOI: http://dx.doi.org/10.1016/0377-0427(87)90125-7

[40] Claude Shannon. 1948. Prediction and entropy of printed English. *Bell Syst. Tech. J.* 30 (1948), 50–64. DOI: http://dx.doi.org/10.1002/j.1538-7305.1951.tb01366.x

[41] Peter P. H. Sneath and Robert R. Sokal. 1973. *Numerical taxonomy*. Freeman, London, UK.

[42] John Steenbruggen, Emmanouil Tranos, and Peter Nijkamp. 2015. *Data from mobile phone operators: a tool for smarter cities? Telecommun. Policy* 39, 3–4 (2015), 335–346.

[43] David Taniar and John Goh. 2007. On mining movement pattern from mobile users. *Int. J. Distrib. Sens. Netw.* 3, 1 (2007), 69–86. DOI: http://dx.doi.org/10.1080/15501320601069499

[44] Vincent S. Tseng and Kawuu W. Lin. 2005. Mining sequential mobile access patterns efficiently in mobile web systems. In *Proc. 19th Int. Conf. Adv. Inf. Netw. Appl.* 1–6.

[45] Michail Vlachos, George Kollios, and Dimitrios Gunopulos. 2002. Discovering similar multidimensional trajectories. In *Proc. 18th Int. Conf. Data Eng.* 673–684. DOI: http://dx.doi.org/10.1109/ICDE.2002.994784

[46] Jingjing Wang and Bhaskar Prabhala. 2012. Periodicity based next place prediction. In *Proceedings of the Nokia Mobile Data Challenge (MDC'12) Workshop*, Vol. 2.

[47] Xiangye Xiao, Yu Zheng, Qiong Luo, and Xing Xie. 2010. Finding similar users using category-based location history. In *Proc. 18th SIGSPATIAL Int. Conf. Adv. Geogr. Inf. Syst. (GIS'10)*. 442–445.
[48] Zhixian Yan, Dipanjan Chakraborty, Christine Parent, Stefano Spaccapietra, and Karl Aberer. 2013. Semantic trajectories: Mobility data computation and annotation. *ACM Trans. Intell. Syst. Technol.* 4, 3 (2013), 49:1–49:38. DOI: http://dx.doi.org/10.1145/2483669.2483682
[49] Zhixian Yan, Dipanjan Chakraborty, Stefano Spaccapietra, Christine Parent, and Karl Aberer. 2011. SeMiTri : A framework for semantic annotation of heterogeneous trajectories. In *Proc. 14th Int. Conf. Extending Database Technol.*
[50] Yang Ye, Yu Zheng, Yukun Chen, Jianhua Feng, and Xing Xie. 2009. Mining individual life pattern based on location history. In *Proc. IEEE Int. Conf. Mob. Data Manag.* 1–10. DOI: http://dx.doi.org/10.1109/MDM.2009.11
[51] Josh Jia-ching Ying, Eric Hsueh-chan Lu, Wang-chien Lee, and Vincent S. Tseng. 2010. Mining user similarity from semantic trajectories. In *Proc. 2nd ACM SIGSPATIAL Int. Work. Locat. Based Soc. Netw. (LBSN'10)*. 19–26.
[52] Jin Soung Yoo and Shashi Sekhar. 2009. Similarity-profiled temporal association mining. *IEEE Trans. Knowl. Data Eng.* 21, 8 (2009), 1147–1161.
[53] Andreas Weingessel, Evgenia Dimitriadou, and Sara Dolničar. 2002. An examination of indexes for determining the number of clusters in binary data sets. *Psychometrika* 67, (2002), 137–159.
[54] Yu Zheng. 2011. Location-based social networks : Users. In *Computing with Spatial Trajectories. Springer.*
[55] Yu Zheng and Xing Xie. 2010. Learning location correlation from GPS trajectories. In *Proc. 2010 11th Int. Conf. Mob. Data Manag.* 49 (2010). 27–32. DOI: http://dx.doi.org/10.1109/MDM.2010.42

Received April 2018; revised March 2019; accepted March 2019