International Journal of Innovative
Computing, Information and Control
Volume **9**, Number **11**, November **2013**

ICIC International ©2013 ISSN 1349-4198
pp. 4263–4284

CLUSTERING MOBILITY PATTERNS IN WIRELESS NETWORKS WITH A SPATIOTEMPORAL SIMILARITY MEASURE

Thuy Van T. Duong[1] and Dinh Que Tran[2]

[1]Faculty of Information Technology
Ton Duc Thang University
Nguyen Huu Tho Street, Tan Phong Ward, 7th District, Ho Chi Minh City, Vietnam
vanduongthuy@yahoo.com

[2]Faculty of Information Technology
Post and Telecommunication Institute of Technology
Km10, Nguyen Trai Street, Ha Dong District, Hanoi City, Vietnam
tdque@yahoo.com

Received October 2012; revised February 2013

Abstract. *Clustering is a technique in data mining whose task is to classify objects into groups. In the recent years, it has been utilized to predict mobility behaviors of users for improving the quality and the management of services in wireless networks. Most of the current solutions focus on extending the traditional k-means approach with the numerical data to the categorical ones. However, such an extension paradigm may result in the loss of semantics of the spatio-temporal mobility patterns of users in the wireless network. Moreover, applying the random choice of initial values (or seeds) of the k-means technique may produce a different local optimum in every run time and thus lead to various partitionings. In this paper, we first propose a model for estimating the similarity among mobility patterns based on the weighted combination of Spatial and Temporal Pattern Similarity measures (STPS) of mobile users in wireless networks. Then we introduce the algorithm of Similarity Mobility Pattern based Clustering (SMPC), which is an alternative extension of the traditional k-means technique. Our approach focuses on using the proposed measure STPS to define a new concept of "cluster center" and to construct two novel procedures: a center updating procedure and a seed initialization procedure. We have conducted experiments with various conditions and parameters to investigate the suitability of the proposed similarity measure STPS and the quality of clusters generated from the algorithm SMPC for mobility patterns in the wireless environment. Experimental results have demonstrated that: (i) Integrating the spatial and temporal characteristics of mobility patterns in the similarity model improves considerably the clustering quality; (ii) Our seed initialization and center updating procedures achieve the stability and the computational speed better than ones with the traditional random initialization; (iii) Our clustering algorithm SMPC with the proposed combination similarity measure is more effective in computation than the other ones.*

Keywords: Clustering, Mobility group, Mobility patterns, Mobile user, Similarity measure, Wireless networks

1. **Introduction.** The popularity of Wireless Local Area Networks (WLANs) and mobile devices such as cellular phones, laptops, PDAs enables WLAN users to utilize services more and more easily and effectively for their daily activities. WLAN logs which are collected from such mobile devices may provide useful information resources for various application areas such as traffic management, location management, purchasing behavior analyzing, location-aware advertising. In the recent years, discovering knowledge from WLAN logs has become a major focus in studies about WLANs. Hsu et al. [1, 2] analyse WLAN logs to understand the nature of user preferences which is a fundamental task

for designing efficiently mobile networks. Some other research works, e.g., [3, 4, 5], try to exploit the information of mobile users in order to provide the next Location-Based Services (LBS) for their movement. Most of these studies focus on discovering mobility behaviors from the WLAN logs to predict future movement of mobile users [6, 7, 8, 9, 10, 11, 12]. Since most of mobility prediction models are based only on the individual own movement history, the incompleteness on information of movement history may limit the extraction of mobility rules and in turn, the lack of extracted rules may affect the accuracy in prediction. Hsu et al. [2] state that WLAN users may follow various mobility patterns but their movements frequently exhibit mobility characteristics of group. Thus, the movement prediction problem is reduced to the one of clustering mobility patterns in the wireless domain into seperate groups. And several techniques for clustering these patterns have been proposed in the literature.

Some studies make use of the Euclidean distance to determine the similarity among categorical sequences. Ma et al. [13] have utilized it to discover sequential patterns from mobile user histories for improving location management in wireless communication systems. Wang and Li [14] used it to introduce a simple sequential clustering algorithm for predicting the group mobility and partition in wireless Ad-hoc networks. However, Do and Kim [24] state recently that Euclidean distance may be a poor measure of similarity for categorical sequences such as mobility patterns. The other studies are based on the hierarchical approach to build a hierarchy of clusters by using merging or splitting technique of mobility patterns. For instance, Oh and Kim [16] proposed a method in order to cluster categorical sequences into groups. They first defined a measure to compute the similarity between two sequences and then constructed a hierarchical clustering algorithm based on the new similarity measure. Hsu et al. [2] also made use of hierarchical clustering method to discover behavioral group based on the eigenbehavior vectors. These authors have classified WLAN users into groups of similar behaviors to understand the nature of user mobility preferences in WLANs. However, Huang [17] demonstrated that hierarchical clustering may be not efficient for application domain with large datasets.

The interesting point is that the traditional k-means algorithm has been widely used in many application areas. The benefit of this technique is that it is scalable to large datasets and thus suitable for discovering knowledge from various data resources [17]. However, its original framework is for numerical data and makes use of the random initialization for k cluster centers. In order to get over the limitation, several researches [17, 18, 19, 20, 21] have focused on developing the traditional k-means approach for the following issues:

- *Extending to categorical data*: using a similarity measure to define a new cluster concept in categorical domain instead of "mean" as in the numerical domain [17, 18]. Huang [17] extended the k-means algorithm by introducing a new concept of "mode" based on a dissimilarity measure for categorical objects. He proposed a technique of updating modes to minimize the clustering cost function in clustering process. It is due to non-unique mode of each cluster, clustering results of k-modes algorithm depend strongly on the selection of modes in clustering process. San et al. [18] also defined a new notion of "representative" for categorical domain. The k-representatives algorithm has only one representative for each cluster and thus deal with the drawback of the k-modes algorithm. However, the k-representatives algorithm may produce a locally optimal partitioning because of random initialization.
- *Proposing an initialization procedure*: In clustering algorithm, the random choice of starting values (or seeds) may produce a different local optimum in every run time and further lead to various partitionings [19]. This problem may be overcome by a good initialization procedure [20] and has attracted a great deal of research

interests. The good initialization is crucial for finding globally optimal partitionings [20]. Some recent works [19, 20, 21] have focused on improving the initialization procedure in order to find the globally optimal partitionings. Ranjan et al. [20] demonstrated that there is no initialization procedure for k-means paradigm that is the best across all datasets. Therefore, it is of great interest to understand which initialization procedure is good for our scenarios.

In this paper, we first propose a model of similarity measure for mobility patterns, which is based on the weighted combination of temporal and spatial similarity measures (STPS: Spatial and Temporal Pattern Similarity). Then we introduce the clustering algorithm SMPC (Similarity Mobility Pattern based Clustering) which is an alternative extension of the traditional k-means technique. Our proposed clustering approach focuses on using measure STPS to define a new concept of "cluster center" and to construct two novel procedures. The first one is the center update procedure which is constructed to compute the optimum cluster centers every time mobility patterns are reassigned to their nearest clusters. The second one is the initialization procedure which is constructed to find a good initial set of cluster centers instead of a set of randomly chosen cluster centers.

The contributions of our work are three-fold.

- Proposing a novel model of similarity measure STPS, which is the basis for constructing our clustering approach. The model STPS computes the similarity between two mobility patterns in wireless networks by exploiting both spatial and temporal properties of them.
- Introducing a new clustering algorithm SMPC for classifying mobility patterns into groups. The algorithm SMPC extends the k-means paradigm to the categorical domain of mobility patterns with a new concept "center" and two novel procedures. The cluster center is defined as an optimum center of each cluster; the center update procedure is constructed by means of a minimum of dissimilarities of patterns in a cluster; the initialization procedure is made from the maximization of dissimilarities among seeds.
- Conducting experiments to demonstrate:
  - The weighted combination of spatial and temporal characteristics of mobility patterns in the model STPS improves considerably the quality of the clustering;
  - The new concept of "cluster center" and two novel procedures of initialization and center updating contribute to reducing the time cost of computation and to achieving clustering results as stable as possible;
  - Both model STPS and algorithm SMPC are effective and efficient by examining the various effect of internal parameters on them and in comparison with some other works.

The remainder of this paper is organized as follows. Section 2 presents a measure model for estimating the similarity between the two mobility patterns in wireless networks. Section 3 introduces an alternative extended k-means algorithm for clustering mobility patterns. Section 4 is devoted to describing experiments and results for evaluating the proposed similarity measure and clustering algorithm. Section 5 gives a discussion and related works introduction. Finally, Section 6 draws concluding remarks and our further research work.

2. **Computational Model of Similarity.**

2.1. **Mobility patterns in wireless networks.**

This section presents briefly the model of wireless network and its pattern representation (For more detail, see [12, 22]), which is utilized for constructing the similarity measure described in the next subsection. In

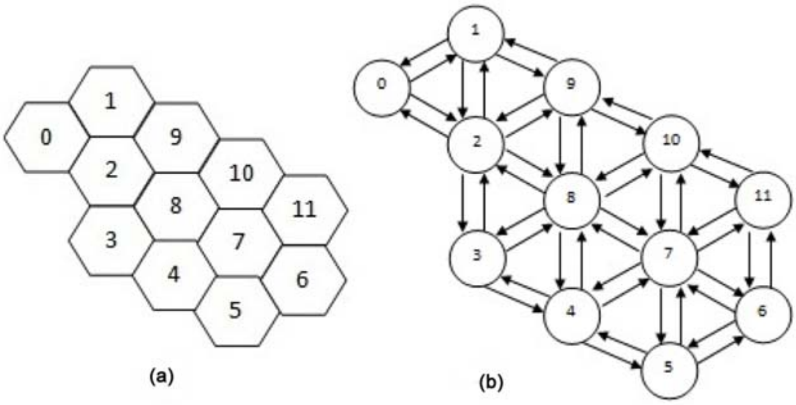


FIGURE 1. The coverage region (a) and the corresponding graph G (b)

the wireless network environment, the mobile users can travel around the radio coverage region that is illustrated by a hexagonal shaped network (see Figure 1). Each hexagon is a cell which is served as a Base Station (BS) in the communication space. The mobility of WLAN users is represented as an unweighted directed graph $G = (V, E)$, where the vertex set V is the set of cells in the coverage region and the edge set E represents the adjacency between pairs of cells. Each cell in V is determined with the ID number c and and then a location of a mobility user at the timestamp t is defined as a couple (c, t). The bidirected edges illustrate the fact that mobile users may move from one cell to another directly and vice versa.

Definition 2.1. *Let C and T be two sets of ID cells and predefined timestamps, respectively. The ordered pairs $q = (c, t)$, in which $c \in C$ and $t \in T$, is called a point. Denote Q to be the set of all points, then $Q = C \times T = \{(c, t) | c \in C \text{ and } t \in T\}$.*

Two points $q_i = (c_i, t_i)$ and $q_j = (c_j, t_j)$ are said to be equivalent if and only if $c_i = c_j$ and $t_i = t_j$. Point $q_i = (c_i, t_i)$ is defined to be earlier than point $q_j = (c_j, t_j)$ if and only if $t_i < t_j$, and it is denoted as $(c_i, t_i) < (c_j, t_j)$ or $q_i < q_j$.

Definition 2.2. *The mobility pattern is defined as a finite sequence of temporally ordered points $p = \langle (c_1, t_1), (c_2, t_2), \dots, (c_k, t_k) \rangle$ in $C \times T$ space, where ID cells of two consecutive points must be neighbors in the coverage region.*

Note that the value of each timestamp t_j is not unique in a mobility pattern, i.e., t_j may be equal to t_i if they are timestamps of two consecutive points of a mobility pattern. For example $\langle (c_1, t_1), (c_2, t_2), (c_3, t_2), (c_4, t_4) \rangle$ is a mobility pattern.

Definition 2.3. *The length $L(P)$ of a mobility pattern p is the number of points in P.*

2.2. **Model of similarity.** This section is devoted to describing a similarity model called STPS (Spatial and Temporal Pattern Similarity) for estimating the similarity between mobility patterns in wireless networks, which is the basis for constructing the clustering algorithm in Section 3. Our similarity model is motivated by the fact that the location of mobility users in wireless networks should be characterized via both spatial and temporal similarities:

- Two mobility patterns are considered to be more similar in space if they share more common cells;
- Two patterns passing through the same cells at the same times must be considered more similar in time than the case they stayed at the different times.

Formally, our similarity model is represented by a 4-tuples (P, S, H, O), in which:

- $P = \{P_1, P_2, \dots, P_n\}$ – a set of mobility patterns;
- $S =| P | * | P |$ – a matrix of spacial similarity. Elements S_{ij} represent the spacial similarity between mobility patterns P_i and P_j that is calculated via Definition 2.6.
- $H =| P | * | P |$ – a matrix of temporal similarity. Elements H_{ij} represent the temporal similarity between mobility patterns P_i and P_j that is calculated via Definition 2.7.
- $O =| P | * | P |$ – a matrix of overall similarity or combination similarity. Elements O_{ij} represent the overall similarity between mobility patterns P_i and P_j that is a weighted combination of spatial similarity and temporal similarity as presented in Definition 2.9. The smaller the value O_{ij} is, the more similar the two mobility patterns P_i and P_j are.

In order to determine components S, H and O of the similarity model, we define three corresponding similarity measures, which are updated and extended from our previous work [23] for estimating the values of S_{ij}, H_{ij} and O_{ij}.

Definition 2.4. *Let P be a set of mobility patterns. A similarity measure $d : P \times P \to [0, 1]$ is a function from a pair of patterns to a real number between zero and one such that the following conditions are satisfied:*
(i) Reflexivity: for all $P_i \in P$ $d(P_i, P_i) = 0$;
(ii) Symmetry: for all $P_i, P_j \in P$ $d(P_i, P_j) = d(P_j, P_i)$.

2.2.1. *Spatial similarity.* In order to estimate the value of each element S_{ab} in component S, we define a measure for computing the similarity in space between two mobility patterns P_a and P_b. For simplicity of presentation, this section makes use of the mobility pattern without the time factor.

Given such two mobility patterns: $P_a = \langle c_{a1}, c_{a2}, \dots, c_{an}\rangle$ and $P_b = \langle c_{b1}, c_{b2}, \dots, c_{bm}\rangle$, for all $1 \le i \le n$, $1 \le j \le m$ and $c_{ai}, c_{bj} \in V$. Spatial similarity measure can be defined in terms of spatial dissimilarity between two mobility patterns. The more uncommon cells there are in two patterns, the more spatially dissimilar they are.

Definition 2.5. *Let $g : P \times P \to R$ be a function representing the number of cells in pattern P_a but not in pattern P_b. Then, g is determined by the formula:*

$$g(P_a, P_b) = card(\{c_{ai} | c_{ai} \in P_a,\ c_{ai} \notin P_b\}) \tag{1}$$

It is easy to prove the following proposition:

Proposition 2.1.
(i) $0 \leqslant g(P_a, P_b) \leqslant L(P_a)$, where $L(P_a)$ is the length of mobility pattern P_a;
(ii) $g(P_a, P_a) = 0$, for all $P_a \in P$;
(iii) $g(P_a, P_b) = L(P_a)$, if P_a and P_b do not share any cells.

A spatial similarity measure between two patterns is then defined as follows.

Definition 2.6. *The spatial similarity measure $d_{space}(P_a, P_b)$ between two patterns P_a and P_b, is defined as follows:*

$$S_{ab} = d_{space}(P_a, P_b) = \frac{g(P_a, P_b) + g(P_b, P_a)}{L(P_a) + L(P_b)} \tag{2}$$

It is clear that if two patterns are equal, $d_{space}(P_a, P_a) = g(P_a, P_b) + g(P_b, P_a) = 0$. Conversely, if the two patterns do not share any cells, the number of uncommon cells in these patterns is $g(P_a, P_b) + g(P_b, P_a) = L(P_a) + L(P_b)$. Thus, $d_{space}(P_a, P_b) = 1$. It is easy to check the reflexivity and symmetry properties of $d_{space}(P_a, P_b)$. We have the following proposition.

Proposition 2.2. *The function d_{space} is a similarity measure.*

2.2.2. *Temporal similarity.* The temporal similarity measure is constructed to compute the value of each element H_{ab} in component H of computational model. The temporal dependency of mobile objects has been widely considered in the recent researches (e.g., [3, 5, 12]). Our alternative approach is based on intuition that two patterns must be considered temporally similar when they pass through the same cells at the same time in the wireless network. For example, two mobility patterns $P_1 = \{(1, t_1), (0, t_3), (5, t_4), (6, t_6), (7, t_9)\}$ and $P_2 = \{(0, t_3), (5, t_4), (7, t_9)\}$ have three common cells 0, 5 and 7 at times t_3, t_4 and t_9, respectively. And they are considered to be temporally similar.

Then, our temporal similarity measure is defined by means of the temporal dissimilarity between two mobility patterns. And in turn, the temporal dissimilarity is calculated to be the sum of all temporal differences between the timestamps of the common cells in two patterns. The smaller the total temporal difference is, the more temporally similar the two patterns are. The formalization of the temporal similarity measure between patterns P_a and P_b is given in the following definition.

Definition 2.7. *Suppose $P_a = \langle(c_{a1}, t_{a1}), \ldots, (c_{an}, t_{an})\rangle$ and $P_b = \langle(c_{b1}, t_{b1}), \ldots, (c_{bm}, t_{bm})\rangle$ are two mobility patterns, where $c_{ai}, c_{bj} \in C$ and $t_{ai}, t_{bj} \in T$ for all i, j. The temporal similarity measure $d_{time}(P_a, P_b)$ between two patterns P_a and P_b is given by*

$$H_{ab} = d_{time}(P_a, P_b) = \frac{1}{k} \sum_{i=1,j=1}^{n,m} \frac{|t_{ai} - t_{bj}|}{\max(t_{ai}, t_{bj})}, \text{ where } c_{ai} = c_{bj} \tag{3}$$

where k is the number of common cells of P_a and P_b.

It is easy to prove the following proposition.

Proposition 2.3. *The function $d_{time}(P_a, P_b)$ is the similarity measure.*

2.2.3. *Combination similarity.* Resulting from the above partial similarity measures, we may construct the definition of the weighted combination similarity measure. Intuitively, the combination similarity of spatial and temporal similarities must satisfy the following constraints:

- It must be neither lower than the minimal and nor higher the maximal of spatial similarity and temporal similarity;
- The higher the partial similarity is, the higher the combination similarity is.

These constraints may be formulated by the following *combination function*:

Definition 2.8. *A function $u : [0, 1] \times [0, 1] \to [0, 1]$ is called the combination function, denoted com-function, if and only if it satisfies the following conditions:*
(i) $\min(s, h) \leqslant u(s, h) \leqslant \max(s, h)$*;*
(ii) $u(s_1, h) \leqslant u(s_2, h)$ *if* $s_1 \leqslant s_2$*;*
(iii) $u(s, h_1) \leqslant u(s, h_2)$ *if* $h_1 \leqslant h_2$*.*

Proposition 2.4. *The function $u : [0, 1] \times [0, 1] \to [0, 1]$ defined by the formula:*

$$u(x, y) = w_{space} * x + w_{time} * y, \text{ where } w_{space} + w_{time} = 1$$

is the com-function.

Proof: We will prove that:

$$\min(x, y) \leqslant u(x, y) \leqslant \max(x, y)$$

We have:

$$u(x, y) = w_{space} * x + w_{time} * y$$

If $x \leqslant y$ then $\min(x, y) = x$, $\max(x, y) = y$ and $y = x + \epsilon$ where $\epsilon \geqslant 0$.
By replacement:

$$\begin{aligned} u(x, y) &= w_{space} * x + w_{time} * (x + \epsilon) && (4) \\ &= w_{space} * x + w_{time} * x + w_{time} * \epsilon && (5) \\ &= x * (w_{space} + w_{time}) + w_{time} * \epsilon && (6) \end{aligned}$$

Due to $w_{space} + w_{time} = 1$, $u(x, y) = x + w_{time} * \epsilon$ and consequently $u(x, y) \geqslant x$ when $\epsilon \geqslant 0$. Additionally, due to $w_{time} \leqslant 1$, $w_{time} * \epsilon \leqslant \epsilon$. Thus, $u(x, y) = x + w_{time} * \epsilon \leqslant x + \epsilon = y$. Similarly, we prove that $y \leqslant u(x, y) \leqslant x$, when $x > y$. Thus, the first condition (1) has been proven.

The second condition (2) is proven as follows. We have:

$$u(x_1, y) = w_{space} * x_1 + w_{time} * y$$

and

$$u(x_2, y) = w_{space} * x_2 + w_{time} * y$$

If $x_1 \leqslant x_2$ then $w_{space} * x_1 \leqslant w_{space} * x_2$. Thus,

$$u(x_1, y) = w_{space} * x_1 + w_{time} * y \leqslant w_{space} * x_2 + w_{time} * y = u(x_2, y)$$

The second condition (2) has been proven. Similarly, it is easy to prove the third condition (3). Thus, the proposition is proven.

Definition 2.9. *Combination similarity measure $d(P_a, P_b)$ between two patterns P_a and P_b is defined by the formula:*

$$d(P_a, P_b) = w_{space} * d_{space}(P_a, P_b) + w_{time} * d_{time}(P_a, P_b) \qquad (7)$$

in which $w_{space} + w_{time} = 1$, and $d_{space}(P_a, P_b)$ and $d_{time}(P_a, P_b)$ are spatial and temporal similarity measures, respectively.

It is easy to prove the following proposition:

Proposition 2.5. *Function $d(P_a, P_b)$ is similarity measure.*

3. Clustering Mobility Patterns Based on Spatiotemporal Similarity.

3.1. **The proposed clustering algorithm.** The purpose of the clustering algorithm is to partition the set of mobility patterns into k clusters such that patterns within the same cluster have a high degree of similarity, whereas patterns belonging to different clusters have a high degree of dissimilarity. Our new algorithm – SMPC (Similarity Mobility Pattern based Clustering) – is an alternative extension of the k-means method in clustering categorical data (e.g., [18, 24]). Our approach focuses on improving an initialization procedure and on constructing a procedure of updating cluster center.

Instead of the traditional random initialization, k seeds are chosen by using the initialization procedure in Subsection 3.2. After choosing k initial centers, we apply the assignation procedure in Subsection 3.3 to assign each pattern in dataset to the nearest cluster (as showed in Algorithm 1, line 5). Then, the center of each cluster will be updated by means of the combination similarity presented in Definition 3.2 (Algorithm 1, line 9). Based on the updated cluster centers, we reassign each pattern in dataset to cluster according to assignation procedure (line 11). The center updating procedure and the assignation procedure are repeated until no pattern has changed clusters via a test cycle of the whole dataset (lines 6-12). The proposed clustering procedure is presented in Algorithm 1.

In Algorithm 1, we use array $A[n]$ to contain the cluster ID of each pattern. For example, $A[i] = j$ means pattern p_i is assigned to jth-cluster. Furthermore, the center of

Algorithm 1 SMPC Similarity Mobility Patterns Clustering Algorithm

```
Input: The dataset $P = \{P_1, P_2, \ldots, P_n\}$
Output: $k$ clusters
for all $i = 0$ to $n$ do
    $A[i] = 0$                              // initialize array $A[n]$
end for
$C \leftarrow$ InitializationProcedure(P)      // initialize $k$ seeds
// Assign each pattern to the nearest cluster using $k$ seeds
$A[n] \leftarrow$ AssignationProcedure($P, C, A[n]$)
repeat
    $changed = 0$                           // flag to decide repeat or stop
    for all cluster X do
        UpdateProcedure(X)                  // update the center of each cluster
    end for
    $changed =$ AssignationProcedure($P, C, A[n]$)   // reassign each pattern to cluster
until $changed = 0$                          // no pattern has changed its cluster
return $A[n]$
```

clusters may be changed whenever there is at least one pattern that changes its cluster. In order to realize whether a pattern is moved to the other cluster, we use flag *changed*. If there is a pattern changing clusters, the value of *changed* will be 1. Otherwise, the value of *changed* will be 0.

3.2. **A new method for initializing k seeds in algorithm SMPC.** In this section, we introduce a new initialization method which is based on the dissimilarity measure among seeds. The main idea of this method is that a seed is randomly selected firstly from the dataset and each of the remaining seeds is chosen by maximizing the sum of all dissimilarities between it and all previous seeds.

Let $P = \{P_1, P_2, \ldots, P_n\}$ be a set of mobility patterns called *dataset* and k be a positive integer specifying the number of clusters. Denote c_i to be the i^{th}-seed, $1 \leq i \leq k$ and then $C = \{c_1, c_2, \ldots, c_l\}$ be a set of current seeds, $1 \leq l \leq k$.

Definition 3.1. *The next seed is a mobility pattern P_i in the dataset such that it maximizes*

$$D_i = \sum_{j=1}^{l} d(P_i, c_j) \tag{8}$$

where $d(P_i, c_j)$ is the combination similarity measure between pattern P_i and seed c_j as in Definition 2.9.

The initialization procedure is outlined as follows:

1. The first seed c_1 is randomly selected from the dataset.
2. For each pattern P_i in the dataset, $P_i \neq c_1$, $1 \leq i \leq n$, calculating the dissimilarity between P_i and c_1, $d(P_i, c_1)$.
3. The second seed c_2 is the pattern P_i with the maximum of $d(P_i, c_1)$.
4. Let l be the number of current seeds. For each pattern P_i in the dataset, $P_i \notin C$ where C is the set of current seeds, calculating the sum of all dissimilarities between P_i and current seed as in Equation (8), denoted as D_i. The next seed c_{l+1} is the pattern P_i with the maximum of D_i.

Algorithm 2 Initialization Procedure

```
Input: The dataset, $P = \{P_1, P_2, \dots, P_n\}$
Output: $k$ seeds, $C = \{c_1, c_2, \dots, c_k\}$
$c_1 = \text{random}(P)$                // generating a random pattern from dataset $P$
max = 0
$l = 1$
for all mobility pattern $P_i \in P$, $P_i \neq c_1$ do
    if max $< d(P_i, c_1)$ then
        max $= d(P_i, c_1)$
        $c_2 = P_i$
    end if
end for
$l = l + 1$
repeat
    max = 0
    for all mobility pattern $P_i \in P$, $P_i \notin C$ do
        $D_i = 0$
        for all current center $c_j \in C$ do
            $D_i = D_i + d(P_i, c_j)$
        end for
        if max $< D_i$ then
            max $= D_i$
            $c_{l+1} = P_i$
        end if
    end for
    $C = C \cup \{c_{l+1}\}$
    $l = l + 1$
until $l > k$
return $C$
```

5. If $l < k$ then assign $C = C \cup \{c_{l+1}\}$, $l = l + 1$, and return to Step 4. Otherwise stop.

The detail of the initialization procedure is given in Algorithm 2.

3.3. **Assigning patterns to clusters and updating cluster centers.** Let $P = \{P_1, P_2, \dots, P_n\}$ and $C = \{c_1, c_2, \dots, c_k\}$ be a dataset and a set of current cluster centers, respectively. Since each cluster is represented by a cluster center, the clustering problem resulted in assigning each pattern P_i, $1 \leq i \leq n$ in cluster c_j, $1 \leq j \leq k$ such that the dissimilarity between P_i and c_j is least. The assignation procedure is outlined as follows:

1. Calculating the dissimilarity between P_i and c_j, $d(P_i, c_j)$, for each pattern P_i in the dataset, $P_i \notin C$, $1 \leq i \leq n$ and each cluster center $c_j \in C$;
2. Choosing c_j such that $d(P_i, c_j)$ is minimized;
3. Assigning P_i to the cluster that is represented by c_j.

The detail of the assignation procedure is presented in Algorithm 3.

After all patterns have been assigned to clusters, the center of each cluster must be updated. Intuitively, the center of cluster X is a pattern in X such that the sum of all dissimilarities between it and remaining patterns in X is minimum. The remainder of this subsection is devoted to describing the construction of the center update algorithm.

Algorithm 3 Assignation Procedure

```
Input: The dataset, $P = \{P_1, P_2, \dots, P_n\}$
       The set of current cluster centers, $C = \{c_1, c_2, \dots, c_k\}$
       The current assignation, $A[n]$ //array $A$ contains the cluster ID of each pattern
Output: The new assignation, $A[n]$
changed = 0                                  // flag to decide repeat or stop
for all mobility pattern $P_i \in P$, $P_i \notin C$ do
    min = 1                                  // because of $d(P_i, c_j) \le 1$
    for all cluster center $c_j \in C$ do
        if $d(P_i, c_j) <$ min then
            min = $d(P_i, c_j)$
            index = j
        end if
    end for
    if $A[i] \ne index$ then
        $A[i] = index$
        changed = 1                          // marking move pattern $P_i$ to indexth-cluster
    end if
end for
return changed
```

Algorithm 4 Center Update Procedure

```
Input: The cluster $X = \{P_1, P_2, \dots, P_m\}$
Output: The new center of $X$
min = m                                      // because of $O_i = \sum_{j=1}^{m} d(P_i, P_j) \le m$
for all mobility pattern $P_i \in X$ do
    $O_i = 0$
    for all mobility pattern $P_j \in X$ do
        if $P_j \ne P_i$ then
            $O_i = O_i + d(P_i, P_j)$
        end if
    end for
    if $O_i \le$ min then
        min = $O_i$
        $c = P_i$                            // move pattern $P_i$ to indexth-cluster
    end if
end for
return c
```

Definition 3.2. *Let $X = \{P_1, P_2, \dots, P_m\}$ be a cluster. The center of cluster X is a pattern P_i in X such that it minimizes*

$$O_i = \sum_{j=1}^{m} O_{ij=} \sum_{j=1}^{m} d(P_i, P_j) \tag{9}$$

where $P_j \ne P_i$, for all j, $1 \le j \le m$ and $d(P_i, P_j)$ is the combination similarity between patterns P_i and P_j as in Definition 2.9.

The center update procedure is outlined as follows:

1. Initializing $O_i = 0$ for the sum of all dissimilarities between P_i and remaining patterns in X, for each pattern P_i in the cluster X, $1 \le i \le m$;
2. Adding $d(P_i, P_j)$ to the sum O_i: $O_i = O_i + d(P_i, P_j)$, for each remaining pattern P_j in X;
3. Choosing P_i such that O_i is minimized and then P_i is the center of cluster X.

The detail of the center update procedure is given in Algorithm 4.

4. Experimental Evaluation.

This section is devoted to presenting the following issues:

- The statement of the problem and the traditional measures for experimental evaluation;
- Set up a dataset for experimental evaluation;
- Experimentally evaluating the proposed similarity model STPS and the clustering algorithm SMPC;
- Comparing with other approaches.

4.1. Problem statement and basis for experimental evaluation.

4.1.1. *Problem statement.* Given a wireless network with the coverage region as in Figure 1, which is composed of a set of mobile users. Each mobile user has a movement history which records its movements from one cell to another in the coverage region. The movement histories of all mobile users are used to discover the set of mobility patterns. Our purpose is to provide an efficient approach for clustering mobility patterns into mobility groups of similar behaviors. This is the basis for constructing an effective mobility prediction technique based on mobility behaviors in group to deal with the incompleteness on information of individual movement history.

In order to evaluate the proposed clustering approach, we first investigate both similarity measure STPS and clustering algorithm SMPC via internal parameters and then assess how effective it may be. Second, we compare the proposed approach with some other research works in respect of clustering quality. Which measures are utilized for evaluating the quality of the clustering are discussed in the next subsection.

4.1.2. *Measures of clustering quality: Basis for experimental evaluation.* Intuitively, the indication of a good clustering result is that the distance between data objects in the same cluster is low, whereas the distance between data objects in different clusters is high. Currently, three measures have been widely used to investigate the clustering quality ([17, 18, 25, 26]): Overall Entropy (OE), Variation of Information (VI) and the clustering accuracy measure for category data. We also utilize these measures to evaluate our clustering algorithm and they are briefly presented in the remainder of this subsection.

First, clustering quality is traditionally evaluated by *internal measure* and *external measure* [25]. The internal measure reflects the average semantic distance between data objects within each cluster. In contrast, the external measure reflects the average semantic distance between the clusters themselves. In [25], *cluster entropy* (E_c) and *class entropy* (E_l) are defined as the internal and external measures, respectively, and the *Overall Entropy* (OE) is then their linear combination:

$$OE = \beta.E_c + (1-\beta).E_l \tag{10}$$

where $\beta \in [0, 1]$ is empirically determined. The smaller the overall entropy is, the better clustering quality is. And then entropy measures are defined as follows [25].

Suppose $C = C_1 \cup C_2 \cup \cdots \cup C_k$ is a partition on the set of N data objects with the set of labels $\{l_1, l_2, \ldots, l_{k^*}\}$. Let n_j be the total number of data objects with label l_j in the dataset, and n_{ij} be the number of data objects labeled l_j in cluster C_i. Then, the cluster entropy E_c and the class entropy E_l are defined by the following formulae:

$$E_c(C) = -\sum_{i=1}^{k}\sum_{j=1}^{k^*} \frac{n_{ij}}{N} \log \frac{n_{ij}}{|C_i|} \tag{11}$$

$$E_l(C) = -\sum_{j=1}^{k^*}\sum_{i=1}^{k} \frac{n_{ij}}{N} \log \frac{n_{ij}}{n_j} \tag{12}$$

Second, in the case that the labels of data objects are not pre-defined, Gomez-Alonso and Valls [28] proposed the Variation of Information (VI) measure for measuring the cluster quality as follows. Let $C^* = C_1^* \cup C_2^* \cup \cdots \cup C_{k^*}^*$ be the pre-constructed correct partition of the dataset of discourse and $C = C_1 \cup C_2 \cup \cdots \cup C_k$ be the partition which is generated by a clustering algorithm. The value of $VI(C, C^*)$ determines information variation between C and C^* and is defined by Equation (13). The smaller $VI(C, C^*)$ is, the more similar C and C^* is.

$$VI(C, C^*) = H(C|C^*) + H(C^*|C) = H(C) + H(C^*) - 2.I(C, C^*) \tag{13}$$

$$I(C, C^*) = \sum_{i=1}^{k}\sum_{j=1}^{k^*} \frac{|C_i \cap C_j^*|}{N} \log \frac{|C_i \cap C_j^*|/N}{(|C_i|/N).(|C_j^*|/N)}$$

$$H(C) = -\sum_{i=1}^{k} \frac{|C_i|}{N} \log \frac{|C_i|}{N}$$

$$H(C^*) = -\sum_{j=1}^{k^*} \frac{|C_j^*|}{N} \log \frac{|C_j^*|}{N}$$

In this paper, we also discover that VI and OE measures whose classes and cluster entropies with the same weight are equivalent when all data objects own predefined labels (See Subsection 4.3.2).

Third, for clustering categorical data, these measures OE, VI are considered to be unsuitable. Huang [17] and San et al. [18] adopted an extension which may measure the degree of correspondence between the clusters obtained from the algorithm and the class assigned previously. This measure of the clustering accuracy is defined as follows:

$$r = \frac{1}{n}\sum_{l=1}^{k} a_l \tag{14}$$

in which a_l is the number of data objects that occur in both cluster C_l and its corresponding labeled class, and n is the number of objects in the data set. The larger the clustering accuracy is, the better clustering quality is.

4.2. **Synthetic dataset generation.** In the scope of this paper, our experiments are conducted with a dataset generator which describes the movement behaviors of mobile users around the coverage region as in Figure 1. At first, we manually generate 5 movement behaviours which are represented in sequences, called the set of initialized patterns. For each movement behavior, approximately 100 mobility patterns are automatically generated by following procedure. Each mobility pattern is a movement around the graph G in Figure 1. So, mobility patterns are generated by inserting some points $q_i = (c_i, t_i)$ into the movement behavior which satisfy two constraints. The first one is the network constraint, i.e., consecutive cells in patterns are neighbor in graph G and the second one is the temporal constraint which satisfies the ascending order of timestamps in patterns.

Table 1. Datasets

Datasets	**Set of initialized patterns**	**Datasets**	**Set of initialized patterns**
DS1	$\langle(1,t_1),(2,t_3),(8,t_6),(4,t_9)\rangle$	DS3	$\langle(0,t_1),(2,t_3),(8,t_6),(4,t_9)\rangle$
	$\langle(1,t_3),(9,t_4),(8,t_9),(7,t_{10})\rangle$		$\langle(0,t_3),(2,t_4),(8,t_9),(7,t_{10})\rangle$
	$\langle(6,t_3),(7,t_5),(10,t_7),(8,t_9)\rangle$		$\langle(6,t_3),(7,t_5),(10,t_7),(8,t_9)\rangle$
	$\langle(6,t_2),(7,t_6),(4,t_8),(8,t_9)\rangle$		$\langle(6,t_2),(7,t_6),(4,t_8),(8,t_9)\rangle$
	$\langle(3,t_4),(4,t_6),(5,t_8),(6,t_{10})\rangle$		$\langle(3,t_4),(4,t_6),(5,t_8),(6,t_{10})\rangle$
DS2	$\langle(0,t_3),(2,t_4),(8,t_6),(7,t_8)\rangle$	DS4	$\langle(1,t_1),(2,t_3),(8,t_6),(4,t_9)\rangle$
	$\langle(3,t_4),(4,t_6),(5,t_8),(6,t_{10})\rangle$		$\langle(1,t_3),(9,t_4),(8,t_9),(7,t_{10})\rangle$
	$\langle(1,t_3),(9,t_5),(10,t_7),(11,t_8)\rangle$		$\langle(11,t_3),(7,t_5),(10,t_7),(8,t_9)\rangle$
	$\langle(6,t_3),(7,t_5),(10,t_7),(8,t_9)\rangle$		$\langle(11,t_2),(10,t_6),(9,t_8),(8,t_{10})\rangle$
	$\langle(4,t_4),(8,t_6),(2,t_7),(1,t_9)\rangle$		$\langle(3,t_4),(4,t_6),(5,t_8),(6,t_{10})\rangle$

All of 100 generated mobility patterns are assigned the same label. Repeating the procedure for each movement behavior, we obtain a testing dataset which consists of 500 mobility patterns with 5 clusters. Such a size of a testing dataset is common in clustering experiments [27]. We can generate different datasets by using different sets of initialized patterns. In this work, we generated four different datasets DS1, DS2, DS3 and DS4 as in Table 1.

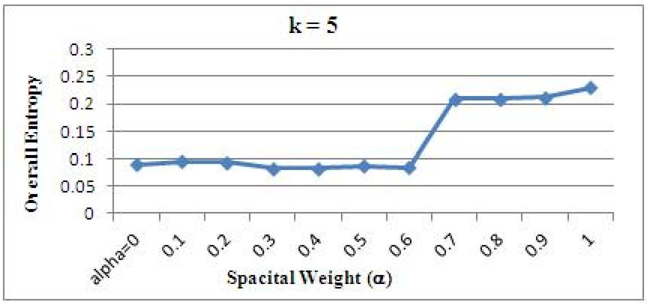
4.3. **Evaluation of proposed approach.** In this section, we examine the effect of each following parameters in order to evaluate how suitable and effective both the similarity measure **STPS** and the clustering algorithm **SMPC** are for classifying mobility patterns into groups of similar behaviors:

- The combination weights in the measure STPS;
- The number of clusters in the algorithm SMPC;
- The random initialization in the clustering algorithm;
- The number of mobility patterns (datasets size);
- The various datasets.

4.3.1. *Effect of weighted combination (α).* In this section, we study how suitable the proposed model STPS is if it reflects the similarity between mobility patterns not only in spatial but also in temporal aspects. For this purpose, we examine the effect of the weighted combination in order to answer the question that the similarity between mobility patterns should be measured based on both spatial and temporal properties or based only either on spatial or temporal property. The experiment is performed on the measure STPS of the form $\alpha.d_{space}(P_a,P_b)+(1-\alpha).d_{time}(P_a,P_b)$ as in Definition 2.9, where $w_{space}=\alpha$ and $w_{time}=(1-\alpha)$. The value of α is varied in the experiments to find how significant the spatial and temporal properties of mobility patterns are to clustering quality; $\alpha=0$ means clustering based on purely temporal similarity, while $\alpha=1$ means clustering based on purely spatial similarity.

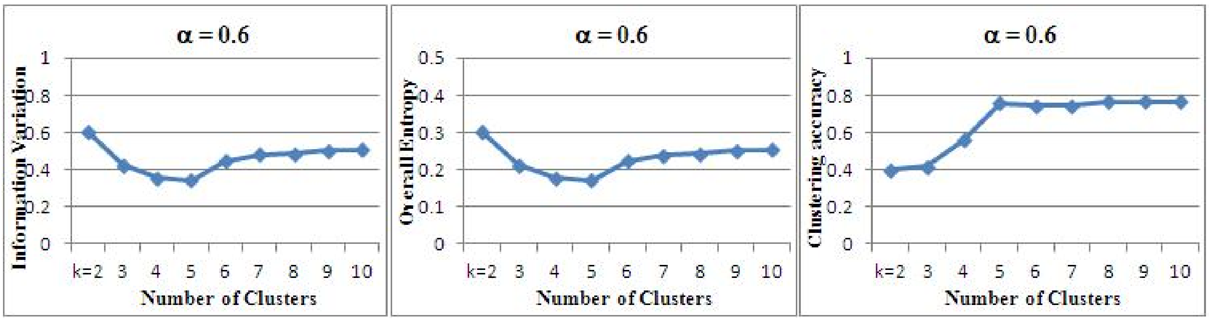
The algorithm SMPC with the measure STPS is run on the dataset DS1 and the number of generated clusters is fixed at $k=5$. Varying α from 0 to 1 on 0.1 incremental steps, we obtain experimental results as in Figure 2 which shows that the clustering quality is improved with α varying from 0.3 to 0.6. This indicates that both spatial and temporal properties are important to the clustering quality. It means that our weighted combination similarity model STPS was well defined.

4.3.2. *Effect of number of clusters (k).* For obtaining the best clustering quality, the optimal value of the number of generated clusters k should be determined by experiments. We run the proposed algorithm SMPC on the dataset DS1 with the optimal value $\alpha=0.6$



Measures	$\alpha = 0$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
OE	0.09	0.095	0.094	0.084	0.084	0.087	0.084	0.21	0.21	0.212	0.23

Figure 2. The effect of combination weight α on clustering quality



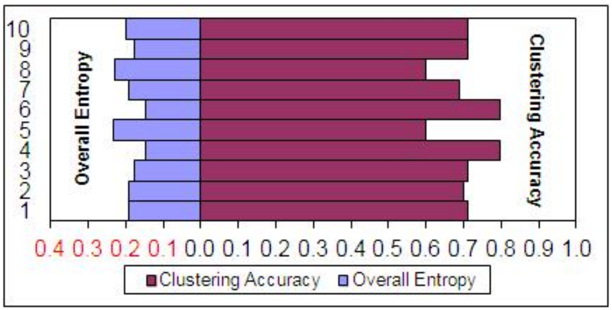
Measures	$k = 2$	3	4	5	6	7	8	9	10
OE	0.3015	0.2125	0.1772	0.171	0.224	0.239	0.242	0.251	0.253
VI	0.603	0.425	0.3544	0.343	0.448	0.479	0.484	0.503	0.506
Accuracy r	0.4	0.418	0.558	0.76	0.744	0.744	0.766	0.766	0.768

Figure 3. The effect of number of clusters (k)

and k varying from 2 to 10. In order to exactly evaluate, we study the effect of the number of generated clusters k with respect to not only Overall Entropy (OE) and Variation of Information measures (VI) but also Clustering Accuracy measure (CA).

For the Overall Entropy measure, we take the equal weights for the cluster entropy and the class entropy, i.e., $\beta = 0.5$ for Equations (10). Figure 3 demonstrates the significant difference of clustering quality with varied k in three measures OE, VI and CA. As expected, the best clustering quality is obtained when $k = 5$, which is the same as the number of clusters of the testing dataset. It implies that our dataset generator is suitable for the proposed clustering method. Furthermore, Figure 3 also shows that the corresponding VI and OE curves actually have the same shape. This demonstration is also in accordance with the equivalence of VI and OE measures.

4.3.3. *Effect of the random initialization in clustering algorithm.* The question now is why an initialization procedure should be proposed instead of random initialization for k cluster centers. To answer this question, we conduct an experiment to study the effect of the random selection for initial values of seeds in clustering algorithm. We first run our clustering approach 10 times on the dataset DS1 with the optimal value $\alpha = 0.6$ and $k = 5$. Then, we will compare clustering results to each other among run times with respect to both overall entropy and clustering accuracy. The difference of clustering quality among run times is presented in Figure 4 which shows that the random initialization for the first seed in algorithm SMPC may affect clustering quality. Therefore, we can say that the clustering quality may be much different among run times if k seeds are randomly chosen.



Run times	Overall Entropy	Clustering Accuracy
1	0.19	0.71
2	0.19	0.70
3	0.178	0.71
4	0.145	0.796
5	0.234	0.6
6	0.145	0.796
7	0.19	0.69
8	0.23	0.6
9	0.178	0.71
10	0.2	0.71

FIGURE 4. Difference of clustering quality among run times

TABLE 2. Improvement in time cost of the proposed initialization procedure

	Initialization Procedure	Random Initialization
Run Time (second)	0.02	0.13

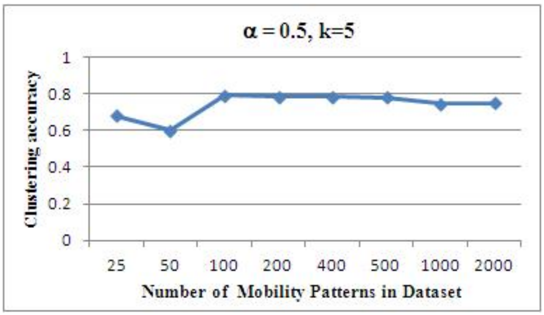
This means that it is necessary to construct an initialization procedure for k seeds in order to achieve clustering quality as stable as possible.

Another experiment should be also conducted to evaluate the proposed initialization procedure. We set up two scenarios:

- *Scenario 1*: based on the proposed initialization procedure
- *Scenario 2*: based on the random initialization

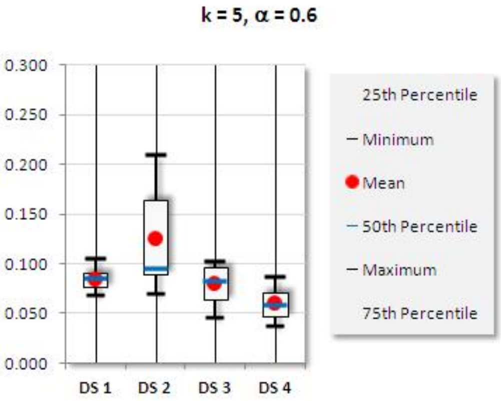
The experimental results in Table 2 indicate that the run time of the Scenario 1 is significant smaller than that of the Scenario 2. The reason is that k seeds in the Scenario 1 are chosen as dissimilar as possible and then each mobility pattern is assigned to cluster exactly. This advance may reduce the number of times to reassign each mobility pattern to another cluster when the cluster centers are updated. Therefore, the clustering process in the scenario based on the proposed initialization procedure may be converged more quickly compared with the scenario based on random initialization for seeds.

4.3.4. *Effect of the number of mobility patterns (dataset size).* The question is that in evaluating clustering quality, what dataset size should we use? In order to answer this question, we will perform experiments with different dataset sizes. We run the algorithm SMPC on the dataset DS2 with different number of mobility patterns at the optimal values $\alpha = 0.5$ and $k = 5$. Figure 5 shows experimental results of comparison which suggests that there is no significant difference of clustering quality among different number of mobility patterns in datasets. However, the light increase of clustering quality at dataset



Number of Patterns	25	50	100	200	400	500	1000	2000
Accuracy (r)	0.68	0.6	0.79	0.785	0.7825	0.782	0.747	0.75

FIGURE 5. Non-significant difference of clustering quality on various number of mobility patterns



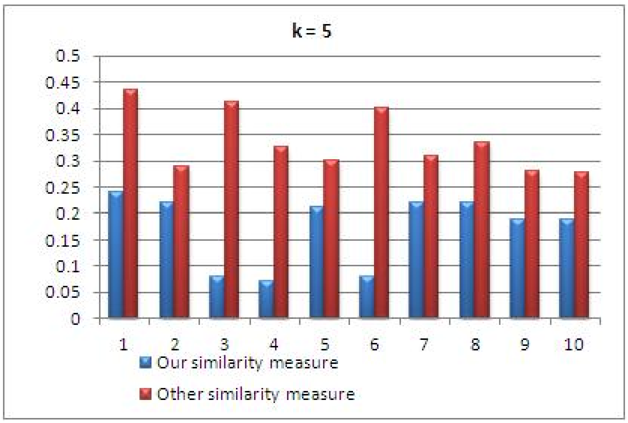
Run times	**DS1**	**DS2**	**DS3**	**DS4**
1	0.087	0.129	0.097	0.078
2	0.092	0.088	0.062	0.054
3	0.083	0.210	0.084	0.041
4	0.084	0.176	0.063	0.037
5	0.087	0.089	0.094	0.086
6	0.074	0.096	0.068	0.053
7	0.068	0.207	0.102	0.063
8	0.071	0.070	0.097	0.071
9	0.092	0.093	0.046	0.045
10	0.105	0.095	0.082	0.07

FIGURE 6. Comparison of clustering quality on various datasets

size in approximately hundreds of mobility patterns indicates suitability of datasets of 500 mobility patterns in above experiments.

4.3.5. *Effect of the various datasets.* In order to test the effect of datasets, we perform experiments on four above datasets DS1, DS2, DS3 and DS4. For each dataset, we run the algorithm SMPC 10 times with respect to the Overall Entropy (OE) and calculate the average value of these clustering results. The obtained data is given in Figure 6 which shows that the change on datasets could affect the clustering quality. However, the difference of clustering results is not too much. Furthermore, if we consider OE < 0.35 as a "good" clustering result, then we can say that the clustering quality is good on various datasets.

4.4. **Comparison with other approaches.** Our clustering approach utilizes the algorithm **SMPC** with the similarity measure **STPS**. Therefore, the clustering quality of the proposed approach depends on both of them.



Run times	Measure STPS	Other Measure
1	0.24	0.435
2	0.22	0.29
3	0.08	0.412
4	0.07	0.326
5	0.21	0.3
6	0.08	0.399
7	0.22	0.31
8	0.221	0.334
9	0.189	0.28
10	0.189	0.278

FIGURE 7. Significant difference of clustering quality between STPS and other similarity measure

4.4.1. *Comparison STPS with other similarity measures.* The question now is whether or not it is necessary to introduce a new similarity model. In order to answer this question, we perform algorithm SMPC on two different similarity measures:

- *Case 1*: The proposed model STPS
- *Case 2*: The similarity measure in [28]

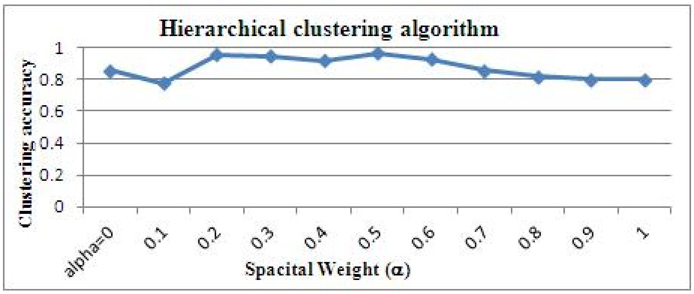
For each case, we run algorithm SMPC 10 times on the dataset DS1 with $k = 5$. In Case 1, the combination weight is taken the optimal value, i.e., $\alpha = 0.6$. Figure 7 illustrates the comparison of clustering quality in two cases. The experimental results show that the clustering quality of Case 1 is better than that of Case 2. Our new similarity model STPS may contribute considerably to the improvement in the clustering quality. In conclusion, we can say that the proposed measure STPS is suitable and effective for measuring the similarity between mobility patterns in wireless networks.

4.4.2. *Comparison SMPC with other clustering algorithms.* In order to compare the proposed algorithm SMPC with the other ones, we implement both algorithm SMPC and the hierarchical clustering algorithm in [16] on the same measure STPS. The experiment results in Table 3 show that the difference of average clustering quality between the two algorithms is not so much. However, the average run time of algorithm SMPC is significantly smaller than that of the other, approximately hundredfold. Hence, we can say that our algorithm SMPC is effective in the time cost of the clustering computation.

4.4.3. *Evaluating measure STPS on other clustering algorithms.* Intuitively, one of the key steps which affects clustering quality is the measure used to compute the similarity between data objects. Thus, this experiment is conducted to make use of other clustering algorithm to evaluate the proposed measure STPS via clustering results. We implement

Table 3. Significant difference of computational speed

	Algorithm SMPC	**Other Algorithm**
Average Clustering Accuracy	0.807	0.85
Average Run Time (second)	0.020	4.940



Measures	$\alpha = 0$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Accuracy (r)	0.86	0.78	0.96	0.95	0.92	0.97	0.93	0.86	0.82	0.8	0.8

Figure 8. The suitability of our similarity model

Table 4. Evaluation of our clustering algorithm

Overall Entropy	**Number of results**	**Good results**
$0 \rightarrow 0.099$	0	Good
$0.1 \rightarrow 0.199$	0	Good
$0.2 \rightarrow 0.299$	30	Good
$0.3 \rightarrow 0.349$	41	Good
$0.35 \rightarrow 0.399$	10	
$0.4 \rightarrow 0.499$	19	
$0.5 \rightarrow 0.599$	0	
$0.6 \rightarrow 1.0$	0	

and run the hierarchical clustering algorithm in [16] using the measure STPS on dataset DS3. The good clustering results (see Figure 8) suggests that the measure STPS are well constructed. Furthermore, the improvement of clustering quality with α varying from 0.2 to 0.6 indicates that it is reasonable to take into account simultaneously spatial and temporal properties of mobility patterns in our proposed measure STPS.

4.4.4. *Evaluating algorithm SMPC on other similarity measures.* In order to avoid the effect of our similarity model on clustering quality, we will implement and run the proposed algorithm SMPC 100 times on the similarity measure introduced in [28]. In this experiment, we use dataset DS1 and fix the number of generated clusters at the optimal value $k = 5$. The experiment produced 100 clustering results which contained 71 good results (OE < 0.35) as in Table 4. This means that there is a 71% chance to obtain a good result by using our algorithm SMPC. Therefore, we can say that the algorithm SMPC is effective for clustering mobility patterns.

4.4.5. *General comparison.* This section is devoted to a summary of the comparison of the performance between our clustering approach and the others, in which the performance of the clustering includes the quality of the clustering and the time cost of the clustering computation.

Evaluation Criteria

Table 5. Input parameters

Parameters	**Values**
Combination weight (α) for measure STPS	0.5
Number of clusters (k) for algorithm SMPC	5
Number of mobility patterns (dataset size)	500
Dataset	DS1
Number of runs for each experiment	100

Table 6. Summary of comparison with other approaches

	Measure STPS	**Other Measure**
Algorithm SMPC	OE = 0.08, Time = 0.02 (s) **Good**	OE = 0.33, Time = 0.02 (s) **Rather Good**
Other Algorithm	OE = 0.06, Time = 4.94 (s) **Rather Good**	OE = 0.21, Time = 4.94 (s) **No Good**

In order to make the results comparable, we use the same values for input parameters in all experiments as in Table 5. For each experiment, we will run 100 times and calculate the average values of clustering Quality (Overall Entropy) and run Time (Second).

Experimental Results

Table 6 shows that the clustering quality in the cases based on measure STPS is significant higher than in the cases based on the measure in [28]. In addition, the results also suggest that using the algorithm in [16] do not increase much clustering quality but increase much time cost compared with the algorithm SMPC. In summary, the measure STPS is good at the quality of the clustering and the algorithm SMPC is good at the time cost of the clustering. Hence, we can say that it is better to use the algorithm SPMC based on measure STPS to classify mobile users into groups of similar mobility behaviours.

5. **Discussions and Related Works.** This section is devoted to discussing how our clustering approach may deal with the problems of the current clustering approaches presented in Section 1.

Firstly, our clustering approach determines the similarity among mobility patterns by using the proposed similarity measure STPS instead of the Euclidean distance as in some studies [13, 14]. Do and Kim [24] demonstrated that the Euclidean distance may be a poor measure of similarity for categorical attributes which are frequently involved in data mining applications. The measure STPS has exploited the characteristics of dependency in space and time of mobility patterns in wireless networks. The suitability and the effectiveness of STPS have been evaluated in experiments. The good quality of clustering results have demonstrated that STPS is well defined for measuring the similarity between the two mobility patterns of mobile users in the wireless environment.

Secondly, since our clustering approach takes advantage of the simplicity and the computational speed of the original k-means algorithm, the time cost of the clustering is significantly smaller than ones based on hierarchical clustering [2, 16]. Huang et al. [31] and the other works [20, 29, 30] have also shown that comparing with conventional approaches, the k-means clustering technique offers several key advantages and most importantly it can be applied to forecast the future. Thus, it has become the most popular clustering algorithm in scientific and industrial applications with large data sets as our work.

Thirdly, our clustering approach works well for categorical data and avoids the locally optimal partitioning, which are drawbacks of most clustering approaches based on k-means algorithm. The benefit of our construction may stem from the following factors:

- We have introduced a new concept of "cluster center" instead of "mean". Some earlier works [32, 33] simply converted categorical data into numeric values and then used the concept of "mean" as the center of cluster, which is computed by Euclidean distance. The disadvantage of these approaches is that they may lead to the loss of semantics in category concept [24]. Therefore, it is necessary to construct a new concept of cluster center which is determined by the similarity measure of categorical data. Moreover, our clustering approach also avoids the dependency of the selection of the cluster centers as in k-modes technique [17]. It is due to the fact of the proposed clustering algorithm SMPC has only one cluster center for each cluster. This advance leads clustering results of SMPC to be more stable than that of k-modes. A cluster center updating procedure has been constructed to find the optimum center of each cluster whenever mobility patterns are reassigned.
- The algorithm SMPC has utilized the proposed initialization procedure for the choice of the initial values (or seeds) instead of random initialization for seeds, which may produce locally optimal partitioning and lead to varying partitionings [19, 20, 21]. In the proposed initialization procedure, the seeds are chosen such that they are as dissimilarily as possible with the aim of reducing the delay of clustering process. It is due to the fact that the more dissimilar the set of seeds is, the more separate the generated clusters are. Thus, each mobility pattern is assigned to the closest cluster exactly and further reducing the number of times to move patterns from one cluster to another because of cluster center update. This advance may enable the clustering process to be converged more quickly compared with some approaches [18], which is based on random initialization for seeds.

Our clustering approach is to classify mobility patterns of mobile users into groups with similar behaviors. Discovering similar mobility groups may lead to that more mobility rules are extracted from the mobility patterns within the same cluster, and thus the prediction of the user movement is more accurate. Utilizing clustering techniques to improve the accuracy in predicting the future locations of mobile users has also been widely studied [2, 3, 4, 6, 7, 8, 10, 34]. The most similar point of view to ours is given by Yang et al. [34]. The authors have found out that most patterns only appear in a group of sequences and more distinctive patterns can be mined from the same cluster. In addition, the patterns which are mined from the sequences within the same cluster are more reliable for prediction. From such observation, they proposed a new method to cluster sequences into different groups, and then make prediction based on the patterns mined from the groups separately. The authors also proposed a new similarity measure for finding common subsequences between two sequences. Their measure is then applied to k-medoids algorithm – a variation of k-means one. In order to improve the performance of k-medoids algorithm, they used the initialization method by Krishnapuram et al. [35]. However, their purpose is to mine sequences of web sections in web logs to understand user behaviors on the web environment. Thus, their pattern representation is completely different from the mobility patterns in time and space in our wireless environment.

6. Conclusions and Future Work.

In this paper, we have presented a new approach to discover group mobility behaviors of mobile users in wireless networks, where the users belong to the same mobility group exhibit more similar movement characteristics. First, we have introduced a model of similarity measure to estimate the similarity between mobility patterns, which is discovered from WLAN logs. Our computational model is a weighted combination of spatial and temporal similarity measures in mobility. Second, we have applied the similarity model to algorithm SMPC for classifying mobility patterns into different mobility groups. Our clustering algorithm is an extension of the k-means paradigm to the categorical domain in wireless network. Our algorithm SMPC is composed of two novel procedures: (i) initialization procedure for starting values of k seeds in the clustering algorithm instead of traditional random initialization, and (ii) update procedure for updating the cluster centers in the clustering process. In order to evaluate the necessity and effectiveness of the model STPS and algorithm SMPC, we have conducted experiments with various simulated conditions and then evaluated results with various measures of clustering quality. The experimental results demonstrate that it is necessary to combine both spatial similarity and temporal similarity. The experiments also indicate that the initialization and update procedures contribute significantly to stability of clustering quality. Furthermore, the effectiveness and accuracy of our proposed clustering method are also verified in comparison with the conventional approach. We are currently utilizing the proposed clustering algorithm SMPC to develop a prediction technique for future movement of mobile users with the aim to improve the precision for predicting user mobility behaviors in wireless network. These research results will be presented in our future work.

REFERENCES

[1] W. Hsu, D. Dutta and A. Helmy, Mining behavioral groups in large wireless LANs, *Proc. of the 13th Annual ACM International Conference on Mobile Computing and Networking*, pp.338-341, 2007.

[2] W. Hsu, D. Dutta and A. Helmy, Mining behavioral groups based on usage data in large wireless LANs, *Technical Report*, http://nile.cise.ufl.edu/w̃eijenhs/publication/trace_mining_tech_report.pdf, 2011.

[3] E. H. Lu, V. S. Tseng and P. S. Yu, Mining cluster-based temporal mobile sequential patterns in location-based service environments, *IEEE Transactions on Knowledge and Data Engineering*, vol.23, no.6, pp.914-927, 2011.

[4] N. Pelekis, G. Andrienko, N. Andrienko, L. Kopanakis, G. Marketos and Y. Theodoridis, Visually exploring movement data via similarity-based analysis, *Journal of Intelligent Information Systems*, vol.38, no.2, pp.343-391, 2012.

[5] J. Kang and H. Yong, Mining spatio-temporal patterns in trajectory data, *Journal of Information Processing Systems*, vol.6, no.4, pp.521-536, 2010.

[6] S. Elnekave, M. Last and O. Maimon, Predicting future locations using clusters' centroids, *Proc. of the 15th Annual ACM International Symposium on Advances in Geographic Information Systems*, USA, 2007.

[7] K. Laasonen, Clustering and prediction of mobile user routes from cellular data, *Knowledge Discovery in Database, Lecture Notes in Computer Science*, vol.3721, pp.569-576, 2005.

[8] T. Anagnostopoulos, C. Anagnostopoulos, S. Hadjiefthymiades, M. Kyriakakos and A. Kalousis, Predicting the location of mobile users: A machine learning approach, *Proc. of the 2009 International Conference on Pervasive Services*, pp.65-72, 2009.

[9] L. Song, D. Kotz, R. Jain and X. He, Evaluating location predictors with extensive Wi-Fi mobility data, *The 23rd Annual Joint Conference of the IEEE Computer and Communications Societies*, vol.2, pp.1414-1424, 2004.

[10] J. Park, Y. Park, S. Kim and G. Cho, An efficient mobile object tracking method based on dynamic clustering in sensor network, *Proc. of the International Conference on Wireless Networks*, pp.36-40, 2008.

[11] I. Nietic, *Analysing Behaviour of Moving Objects*, Ph.D. Thesis, University of Zagreb, 2008.

[12] T. V. T. Duong and D. Q. Tran, An effective approach for mobility prediction in wireless network based on temporal weighted mobility rule, *International Journal of Computer Science and Telecommunications*, vol.3, no.2, pp.29-36, 2012.

[13] S. Ma, S. Tang, D. Yang, T. Wang and J. Han, Combining clustering with moving sequential patterns mining: A novel and efficient technique, *Proc. of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining, LNCS*, vol.3056, pp.419-423, 2004.

[14] K. Wang and B. Li, Group mobility and partition prediction in wireless ad-hoc networks, *IEEE International Conference on Communications*, vol.2, pp.1017-1021, 2002.

[15] F. A. Mazarbhuiya and M. Abulaish, Clustering periodic frequent patterns using fuzzy statistical parameters, *International Journal of Innovative Computing, Information and Control*, vol.8, no.3(B), pp.2113-2124, 2012.

[16] S. Oh and J. Kim, A sequence-element-based hierarchical clustering algorithm for categorical sequence data, *International Journal of Information Technology & Decision Making*, vol.4, no.1, pp.81-96, 2005.

[17] Z. Huang, Extensions to the k-means algorithm for clustering large data sets with categorical values, *Journal of Data Mining and Knowledge Discovery*, vol.2, no.3, pp.283-304, 1998.

[18] O. M. San, V. N. Huynh and Y. Nakamori, An alternative extension of the k-means algorithm for clustering categorical data, *International Journal Applied Mathematics Computer Science*, vol.14, no.2, pp.241-247, 2004.

[19] B. Bahmani, B. Moseley, A. Vattani, R. Kumar and S. Vassilvitskii, Scalable k-means++, *Proc. of the VLDB Endowment*, vol.5, no.7, pp.622-633, 2012.

[20] M. Ranjan, D. P. Anna and P. G. Arka, A systematic evaluation of different methods for initializing the k-means clustering algorithm, *IEEE Transactions on Knowledge and Data Engineering*, http://www.public.iastate.edu/ apghosh/files/IEEEclust2.pdf, 2010.

[21] D. Arthur and S. Vassilvitskii, k-means++: The advantages of careful seeding, *Proc. of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp.1027-1035, 2007.

[22] T. V. T. Duong and D. Q. Tran, Modeling mobility in wireless network with spatiotemporal state, *Southeast-Asian J. of Sciences*, vol.1, no.1, pp.113-125, 2012.

[23] T. V. T. Duong, D. Q. Tran and C. H. Tran, A weighted combination similarity measure for mobility patterns in wireless networks, *International Journal of Computer Networks and Communications*, vol.4, no.3, pp.21-35, 2012.

[24] H.-J. Do and J. Y. Kim, Clustering categorical data based on combinations of attribute values, *International Journal of Innovative Computing, Information and Control*, vol.5, no.12(A), pp.4393-4405, 2009.

[25] J. He, A. H. Tan, C. L. Tan and S. Y. Sung, On quantitative evaluation of clustering algorithms, *Clustering and Information Retrieval*, pp.105-133, 2003.

[26] M. Meila, Compare clusterings – An information based distance, *Journal of Multivariate Analysis*, pp.873-895, 2007.

[27] Z.-Y. Niu, D.-H. Ji and C.-L Tan, Using cluster validation criterion to identify optimal feature subset and cluster number for document clustering, *Information Processing and Management*, vol.43, pp.730-739, 2007.

[28] C. Gomez-Alonso and A. Valls, A similarity measure for sequences of categorical data based on the ordering of common elements, *Proc. of the 5th International Conference on Modeling Decisions for Artificial Intelligence*, pp.134-145, 2008.

[29] J. A. Hartigan and M. A. Wong, A k-means clustering algorithm, *Applied Statistics*, vol.28, pp.100-108, 1979.

[30] P. Berkhin, A survey of clustering data mining techniques, *Grouping Multidimensional Data in Grouping Multidimensional Data*, pp.25-71, 2006.

[31] K.-H. Huang, T. H.-K. Yu and T.-T. Kao, Analyzing structural changes using clustering techniques, *International Journal of Innovative Computing, Information and Control*, vol.4, no.5, pp.1195-1202, 2008.

[32] K. Arunprabha and V. Bhuvaneswari, Comparing K-value estimation for categorical and numeric data clustering, *International Journal of Computer Applications*, vol.11, no.3, pp.4-7, 2010.

[33] N. Lee and J. Kim, Conversion of categorical variables into numerical variables via Bayesian network classifiers for binary classifications, *Computational Statistics & Data Analysis*, vol.54, no.5, pp.1247-1265, 2010.

[34] Q. Yang, J. Kou, F. Chen and M. Li, A novel two-stage scheme built-upon clustering for sequential pattern mining, *International Journal of Innovative Computing, Information and Control*, vol.7, no.5(B), pp.2809-2819, 2011.

[35] R. Krishnapuram, A. Joshi, O. Nasraoui et al., Low-complexity fuzzy relational clustering algorithms for web mining, *IEEE Transactions on Fuzzy Systems*, vol.9, no.4, pp.595-607, 2001.

[36] M. Al-Sanabani, S. Subramaniam, M. Othman and Z. Zukarnain, Mobility prediction based resource reservation for handoff in multimedia wireless cellular networks, *International Arab Journal of Information Technology*, vol.5, no.2, pp.162-169, 2008.