Article

A Novel Approach Based on Time Cluster for Activity Recognition of Daily Living in Smart Homes

Yaqing Liu 1,2,3,4, Dantong Ouyang 2, Yong Liu 3,* and Rong Chen 1

1 School of Information Science & Technology, Dalian Maritime University, Dalian 116026, China; liuyaqing@dlmu.edu.cn (Y.L.); rchen@dlmu.edu.cn (R.C.)
2 Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China; ouyd@jlu.edu.cn
3 Artificial Intelligence Key Laboratory of Sichuan Province, Sichuan University of Science and Engineering, Zigong 643000, China
4 School of Computer and Software, Nanjing University of Information Science & Technology, Nanjing 210044, China
* Correspondence: scliuyong@163.com; Tel.: +86-813-5505925

Received: 23 May 2017; Accepted: 14 September 2017; Published: 1 October 2017

Abstract: With the trend of the increasing ageing population, more elderly people often encounter some problems in their daily lives. To enable these people to have more carefree lives, smart homes are designed to assist elderly people by recognizing their daily activities. Although different models and algorithms that use temporal and spatial features for activity recognition have been proposed, the rigid representations of these features damage the accuracy of activity recognition. In this paper, a two-stage approach is proposed to recognize the activities of a single resident. Firstly, in terms of temporal features, the approximate duration, start and end time are extracted from the activity records. Secondly, a set of activity records is clustered according to the records’ temporal features. Then, the classifiers are used to recognize the daily activities in each cluster according to the spatial features. Finally, two experiments are done to validate the recognition of daily activities in order to compare the proposed approach with a one-dimensional model. The results demonstrate that the proposed approach favorably outperforms the one-dimensional model. Two public datasets are used to evaluate the proposed approach. The experiment results show that the proposed approach achieves average accuracies of 80% and 89%, respectively.

Keywords: smart homes; activity recognition; sensors

1. Introduction

With the trend of the ageing population, more elderly people must live alone and cannot receive care from their children or spouses. It is well known that elderly people are prone to accidents in their daily lives. In traditional homes, it is difficult to recognize in a timely manner that an accident has occurred. To help single elderly people live healthy lives, smart homes are being developed to detect the daily activities of elderly people. Naturally, the activity recognition is the key function in the smart home development.

Over the last decade, there has been considerable research on activity recognition in smart homes. This research can be divided into five categories, depending on the monitoring technology [1]. The first monitoring technology is based on video cameras that are fixed in rooms [2,3]. Activities are recorded by the cameras and recognized by analyzing the video records. Although the camera-based approaches were criticized for privacy exposure, video-processing techniques were recently introduced to anonymize and record only the situations where the user is in potential danger. At present, the main problem of camera-based approaches is the required infrastructure for a pervasive
and everywhere-monitoring coverage. The second monitoring technology is based on sound recognition [4]. Sound recognition uses micro-phones to detect different classes of daily activity, e.g., the sound of doing the dishes or the fall of an object or a person. Obviously, the activity recognition is able to be interfered by noise. The third monitoring technology is based on body-worn sensors [5,6]. The residents are required to wear monitoring devices, and their activities are recorded by the devices in real time. Although this method protects the resident’s privacy, the wearing devices is an extra burden. The fourth monitoring technology is based on pressure sensors [7]. The pressure sensors are used to detect the presence of residents on chairs or in bed, sit-to-stand transfers and stand-to-sit transfers. The monitoring technology can only detect a few simple activities. The fifth monitoring technology is based on ambient sensors [8–14] that are placed in various rooms. Generally, the ambient sensors include light sensors, temperature sensors, magnetic door sensors, etc. When the resident moves or does something in the room, some ambient sensors are activated. The current activity of the resident is inferred from the sensor events. The resident’s activities activate a sequence of ambient sensors. For example, the “washing” activates the ambient sensors that are fixed in the taps. In this manner, the resident has privacy protection and is freed from wearing extra devices. The drawback of the ambient sensor-based solutions is the required disbursement to cover the entire residence. Furthermore, even if the sensors are placed in different rooms, some blind spots can remain. Finally, the activity recognition model cannot address the case when the user modifies his/her behavior patterns at certain instants. To use the ambient sensors and body-worn sensors to obtain better performance, the ambient sensors and body-worn sensors are jointly utilized. For example, Atallah et al. presented a Bayesian classification framework for activity recognition [11]. To alleviate the user’s load, a light-weight e-AR sensor is used to generate prior information. Ambient sensors are harnessed to produce more detailed activity profiling.

Substantial effort has been expended on activity recognition [15,16]. Most researchers have attempted to introduce machine learning into their approaches. Some have used time series models to recognize activities, e.g., the hidden Markov model (HMM) or conditional random fields (CRF). For example, Kasteren et al. used hidden Markov models and the hierarchical hidden Markov model to recognize residents’ activities [17–19]. Tong et al. used the latent-dynamic conditional random fields and hidden state conditional random field for abnormal activities and the activities of single residents and multiple residents [20–22]. The commonality of these approaches is that the order of activities and the order of sensor events are emphasized. However, time series models usually have poor robustness [23]. For example, the daily order of a resident’s activities is not always identical. For a single activity, the order of sensor events often changes. Furthermore, the order of one resident’s activities is different from that of another. To improve the robustness, researchers exploit static classifiers for activity recognition, e.g., naive Bayesian (NB), k-nearest neighbor (kNN) or support vector machine (SVM) [24]. For example, Cook et al. used NB to recognize daily activities [25]. Yin et al. used the one-class SVM to recognize abnormal daily activities [26–28].

Temporal features have been interlaced with spatial features in earlier publications [29,30]. Krishnan et al. [31] incorporated time decay, mutual information-based weighting of sensor events and contextual information to perform activity recognition on streaming sensor data. Lotfi et al. [32] used the echo state network to predict abnormal behaviors for the elderly dementia sufferer. In their approach, the start time and duration were considered. In addition, temporal relations between sensor events were exploited as features to infer activities [33–36]. Generally, the temporal relations were built firstly. Then, the logical rules for activity recognition were generated. In this paper, a two-stage approach is proposed for activity recognition. Firstly, the temporal features of an activity record, e.g., the approximate start time, approximate end time and approximate duration, are used to divide a dataset of activity records into two clusters. Then, the spatial features of the sensor events are used to recognize the activities in each cluster. In the proposed approach and previous approaches, the temporal features are used.

In this paper, a data-driven approach is presented. Compared to the one-dimensional recognition model, the two-stage approach is new. For the first stage, the temporal features are used to cluster the training set. The activity records among different clusters are temporally sensitive, whereas the activity
records within the same cluster are temporally insensitive, and the temporal features are noise data. For the second stage, the activity records within the same cluster are spatially sensitive. Activity records are recognized only regarding spatial features. The contribution of this paper is as follows. Compared to the one-dimensional recognition model, the activity recognition of the second stage is not interfered by noise data and has a high performance, which is shown by the experiments.

The remainder of this paper is organized as follows: firstly, we introduce the related work. Then, the approach to activity recognition is described. Next, we validate the proposed approach and discuss the results. Finally, the obtained findings are summarized, and the future work is declared.

2. Related Work

Activity recognition approaches are commonly divided into the data-driven approaches and knowledge-driven approaches. The knowledge-driven approaches emphasize the recognition rules. In most situations, the recognition rules are generated by following a heuristic strategy and are represented in some logical language, e.g., temporal logic or description logic. Rugrone et al. [37] used temporal logic to represent recognition rules to recognize abnormal activity. Yin et al. and Chen et al. [38,39] represented recognition rules as an ontology, and Chen et al. [40] proposed an improved ontology-based approach. The core of these approaches is an iterative process that begins from the so-called “seed” activity models, which are created by ontological engineering, deployed and subsequently evolved through incremental activity discovery and model updates. The knowledge-driven approaches have strong robustness because the recognition rules can be used in different environments. However, the raw data commonly include substantial noise and uncertain information, which are difficult to identify and consequently affect the accuracy of activity recognition.

The data-driven approaches are classified as supervised approaches, semi-supervised approaches and unsupervised approaches. Early research focused on supervised approaches, where the activity records were first labelled in advance. Then, the labelled activity records were learned by a classifier, e.g., NB [25], DT [41], kNN [42], SVM [43,44], HMM [17–19] or CRF [20–22]. Although supervised approaches achieve high accuracy, labelling the activity records is expensive and time consuming. To avoid data-labelling efforts, unsupervised approaches were proposed [45,46]. Maekawa et al. [47] used information about the end user’s physical characteristics to recognize similar activities of other users according to similarities between activities. Although the labelling efforts can be avoided, the unsupervised approaches are typically criticized for poor accuracy. To reach a trade-off among the accuracy, overhead and labelling efforts of the supervised and unsupervised methods, a growing amount of recent studies has focused on the semi-supervised approaches. For semi-supervised approaches, training data are generally separated into labelled and unlabeled. Four tasks must be orderly fulfilled to generate the recognition model. Firstly, the features are extracted from the sensor events of both labelled data and unlabeled data. Secondly, for the labelled data, sensor events are segmented regarding the start and end boundaries of an entire activity record. The activity features are generated by regarding activity records. Thirdly, for the unlabeled data, the sensor events are segmented by regarding a predefined strategy, e.g., the sliding window strategy. The activity features are generalized by regarding segmented sensor events. The activity labels are assigned to segment the sensor events by calculating the feature similarity between the labelled and unlabeled activity records. Finally, the activity features are extended from both unlabeled activity records and harnessed to generate the recognition model. Some documented semi-supervised studies are as follows. Nef et al. [48] performed a hybrid approach of an unsupervised part and a supervised part to recognize eight activities of daily living. Cluster algorithms were first employed to cluster the same activities according to the spatial locations of the firing sensors. Secondly, three classifiers (NB, SVM and RF) were used for activity recognition. The experiment showed that the random forest classifier was superior to the naive Bayesian and support vector machine in terms of specificity, sensitivity, precision and F-measure. Bourobou and Yoo [23] proposed a similar pattern for activity recognition. Firstly, a cluster algorithm K-pattern was used to find the frequent patterns of activities [49], which were viewed as features of activities. Secondly, a device based on an artificial neural network (ANN) was used to recognize the activities according to their features. Wen and Zhong [50,51] divided
trained sensor events into labelled and unlabelled. The labelled sensor events were used to find the initial activity patterns. The unlabelled events were used to enrich the initial activity patterns if the similarities between the unlabelled events and the initial activity patterns satisfied a minimum similarity. Otherwise, they were clustered to find new activity patterns.

3. Terminologies

To represent the proposed approach, some terminologies are defined in advance. For clarity, a segment of activity records is shown in Table 1.

Definition 1. For a sensor s, sr = (d, h, m, sn, sv, al) is a sensor event iff s is run. d is the date when s was run; h is the hour; and m is the minute. sn is the name of s; sv is the value of s; and al is an explanatory activity label.

Throughout this manuscript, sr.d, sr.h, sr.m, sr.sn, sr.sv and sr.al are used to represent the tuples d, h, m, sn, sv and al of a sensor event sr, respectively. Ω is used to represent the set of sensor events.

For example, “15 June 2011 00:25:01.892474 LS013 7 Sleep” denotes that sensor LS013 is activated at 00:25:01.892474 on 15 June 2011 with the measured value of seven. At that time, the resident was sleeping.

Table 1. A segment of activity records.

<table>
<thead>
<tr>
<th>Number</th>
<th>Date</th>
<th>Time</th>
<th>Sensor</th>
<th>Value</th>
<th>Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15 June 2011</td>
<td>00:06:32.834414</td>
<td>M021</td>
<td>ON</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>15 June 2011</td>
<td>00:12:32.670631</td>
<td>BAT012</td>
<td>9540</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>15 June 2011</td>
<td>00:15:01.957718</td>
<td>LS013</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>15 June 2011</td>
<td>00:25:01.892474</td>
<td>LS013</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>15 June 2011</td>
<td>01:05:01.622637</td>
<td>BAT013</td>
<td>9460</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>15 June 2011</td>
<td>03:38:28.21206</td>
<td>M021</td>
<td>ON</td>
<td>Sleep</td>
</tr>
<tr>
<td>7</td>
<td>15 June 2011</td>
<td>03:38:44.482092</td>
<td>MA013</td>
<td>ON</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>15 June 2011</td>
<td>03:38:45.135517</td>
<td>M018</td>
<td>OFF</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>15 June 2011</td>
<td>03:38:47.645521</td>
<td>MA013</td>
<td>OFF</td>
<td>Bed_Toilet_Transition</td>
</tr>
</tbody>
</table>

Definition 2. Given two sensor events sr1 and sr2, sr1 is said to be the precursor of sr2 iff sr1.d < sr2.d holds or (sr1.d == sr2.d AND sr1.h < sr2.h) or (sr1.d == sr2.d AND sr1.h == sr2.h AND sr1.m < sr2.m) holds. sr2 is said to be the successor of sr1 if sr1 is the precursor of sr2.

Throughout this manuscript, sr1 < sr2 indicates that sr1 is the precursor of sr2.

For example, [15 June 2011 00:25:01.892474 LS013 7 Sleep] is the precursor of [15 June 2011 01:05:01.622637 BAT013 9460 Sleep]. [15 June 2011 01:05:01.622637 BAT013 9460 Sleep] is the successor of [15 June 2011 00:25:01.892474 LS013 7 Sleep].

Definition 3. Given two sensor events sr1 and sr2 for which sr1 < sr2 holds, sr1 is said to be the direct precursor of sr2 iff ¬∃sr ∈ Ω, sr1 < sr AND sr < sr2 holds. sr2 is said to be the direct successor of sr1 if sr1 is the direct precursor of sr2.

Throughout this manuscript, sr1 → sr2 indicates that sr1 is the direct precursor of sr2.

For example, [15 June 2011 00:25:01.892474 LS013 7 Sleep] is the direct precursor of [15 June 2011 01:05:01.622637 BAT013 9460 Sleep]. [15 June 2011 01:05:01.622637 BAT013 9460 Sleep] is the direct successor of [15 June 2011 00:25:01.892474 LS013 7 Sleep].

Definition 4. Given an activity a and n sensor events sr0, sr1, sr2, ..., sn, sn+1, SRs is said to be a sensor sequence of a iff ∀1 ≤ i ≤ n, sr.al == a AND sr0 ≠ a AND srn+1 ≠ a AND ∀2 ≤ i ≤ n − 1 sr. → sr+1 holds.

Definition 5. For an activity a and a sequence of ambient sensors sr0, sr1, ..., sn, of a, ar = (sr0,h, sr1,h, u, SNT, a) is an activity record. u is the approximate duration of a. SNT is the spatial feature and is defined as a set {srn, T)}, where sn ∈ {srn, sn+1 i ≤ i ≤ n} is the name of some sensor, and T = |{srn, sn+1 i ≤ i ≤ n} is the frequency that the sensors named srn occur. u and SNT are solved using Algorithm 1.

Algorithm 1. GenerateActivityRecord
Input: \( sr_1, sr_2, \ldots, sr_n \), a sequence of sensor events

Output: \( u, SNT \)

1. \( u \leftarrow (sr_n.d - sr_1.d) \times 24 \times 60 + (sr_n.h - sr_1.h) \times 60 + (sr_n.m - sr_1.m) \)
2. \( SNT \leftarrow \emptyset \)
3. for each \( sr \) in \( \{sr_1, sr_2, \ldots, sr_n\} \)
4. if \( (sr.sn, T) \in SNT \) then // \( T \) is the number of times that sensor \( sr.sn \) is run.
5. delete \( (sr.sn, T) \) // delete \( (sr.sn, T) \) from \( SNT \)
6. \( SNT \leftarrow SNT \cup \{(sr.sn, getT(sr.sn)+1)\} \) //getT\( (sr.sn) \) is employed to get tuple \( T \) of \( (sr.sn, T) \)
7. else
8. \( SNT \leftarrow SNT \cup \{(sr.sn, 1)\} \)
9. end if
10. return \( u, SNT \)

For the example in Table 1, “(0, 3, 212, \{(MA021, 2), (BATV012, 1), (BATV013, 1), (LS013, 2)\}, Sleep)” is an activity record of “Sleep”. The duration \( u \) is “212” min because the approximate duration between the approximate start and approximate end times of “Sleep” is “212” min. The ambient sensors “MA021” and “LS013” are each run twice; the ambient sensors “BATV012” and “BATV013” are each run once.

4. Methodology

The entire process is composed of five tasks: Proposing, Clustering, GeneratingModel, Aligning and AssigningActivityLabel, which are shown in Figure 1. The first task, Proposing, generates a set of activity records based on the set of sensor events, which is shown in Algorithm 2. Then, the activity record set is divided into a training dataset and a test dataset in some proportion. In the experiment, the proportion is set as 9:1. The second task, Clustering, clusters the training data and test data according to the tuples \( sr_1.h, sr_n.h \) and \( u \) of each activity record using K-means. The third task, GeneratingModel, generates a recognition model for each training cluster. Note that both training data and test data are involved in the Clustering task regarding temporal tuples, which include \( sr_1.h, sr_n.h \) and \( u \). Tuples \( al \) and \( SNT \) are not involved in the Clustering task. The training data are only input into the classifier to generate a recognition model regarding tuples \( SNT \) and \( al \), where \( al \) is the class label of the activity record. In contrast, the activity records in the test set are assigned to class labels by the recognition model regarding the tuple \( SNT \). The fourth task Aligning is to find an optimal alignment between recognition models and the test cluster set, which is shown in Algorithm 3. For each potential alignment, an outlier detection algorithm (LOF) is used to decide whether each test activity record is an outlier against the training set. The alignment with the minimum outliers is optimal. The last task, AssigningActivityLabel, assigns an activity label to each activity record of the test set, which is shown in Algorithm 4.

Note that both real-time activity recognition and off-line activity recognition are discussed in the smart home context [8]. In this paper, the proposed approach focuses on the recognition across pre-segmented sensor data instead of real-time sensor data.
Algorithm 2. GenerateActivityRecordsSet

**Input:** $sr_1, sr_2, ..., sr_n$, a sequence of sensor events

**Output:** $AR$, activity records set

1. $i \leftarrow 1$
2. $k \leftarrow 1$
3. $AR \leftarrow \emptyset$
4. **while** ($i < n-1$) **do**
5.   **if** $sr_{i \cdot al} != sr_{i+1\cdot al}$ **then**
6.     $ar \leftarrow$ generateActivityRecord($sr_i, sr_{i+1}, ..., sr_l$)
7.     $AR \leftarrow AR \cup ar$
8.     $k \leftarrow l + 1$
9.   **else**

---

**Figure 1.** The entire process of activity recognition based on the proposed approach.
Algorithm 3. AlignTrainSet & TestSet

**Input:** $M$, set of recognition models.
$P_{TS}$, test set of activity records.

**Output:** $MP$, a subset of $M \times P_{TS}$.

1. Let $M$ be $\{m_1, m_2, \ldots, m_n\}$
2. Let $P_{TS}$ be $\{Tes_1, Tes_2, \ldots, Tes_n\}$
3. $sm \leftarrow \infty$
4. for any $n$ elements $(m_{xi}, Tes_{xi})$, $(m_{x2}, Tes_{x2})$, ..., $(m_{xn}, Tes_{xn})$ in $M \times P_{TS}$
5. $oc \leftarrow 0$ // $oc$ is used to count outliers of $Tes_{xi}$ against $Tr_{Tra}$
6. $i \leftarrow 1$
7. while ($i \leq n$)
8. for each $ar$ in $Tes_{xi}$
9. if $isOutlier(m_{xi}, ar)$ then // $isOutlier(m_{xi}, ar)$ is used to decide if $ar$ is an outlier or not.
10. $oc \leftarrow oc + 1$
11. end if
12. end for
13. end while
14. if $oc < sm$ then
15. $sm \leftarrow oc$
16. $MP \leftarrow \{(m_{x1}, Tes_{x1}), (m_{x2}, Tes_{x2}), \ldots, (m_{xn}, Tes_{xn})\}$
17. end if
18. end for
19. return $MP$

Algorithm 4. RecognizeActivity

**Input:** $\{m_1, m_2, \ldots, m_n\}$, set of recognition models
$\{Tes_1, Tes_2, \ldots, Tes_n\}$, test set
$C$, classifiers including NB, kNN, C4.5 and RF

**Output:** $\{AL_1, AL_2, \ldots, AL_n\}$, activity class set

1. $i \leftarrow 1$
2. $j \leftarrow 1$
3. while ($i \leq n$)
4. Let $Tes_i$ be $\{ar_0, ar_1, \ldots, ar_m\}$ // $ar_0$ is an activity record.
5. $j \leftarrow 1$
6. while ($j \leq m$)
7. $al_0 \leftarrow assignLabel(C, m_0, ar_0)$ // $m_0$ is fed to C and C outputs an activity label.
8. $AL_i \leftarrow \bigcup_{j=1}^{m} \{al_j\}$
9. $j \leftarrow j + 1$
10. end while
11. $i \leftarrow i + 1$
12. end while
13. return $\{AL_1, AL_2, \ldots, AL_n\}$
5. Results and Evaluation

5.1. Datasets

The two used datasets were selected from the “single-resident apartment data” provided by Washington State University, U.S. [52,53]. Datasets “HH102” and “HH104” were used to validate the recognition of daily activities by comparing the two-stage approach ("TS") and one-dimensional ("SS") model.

The details of dataset “HH102” are displayed in Table 2. These data are obtained from healthy elderly people. The measurement time was 64 days. A total of ninety-seven sensors was installed in the apartment. The sensors included the following categories.

(1) Identifiers starting with "BA" indicate the sensor battery levels: BATP013, BATP019, BATV001–BATV023, BATV102–BATV105.
(2) Identifiers starting with "D" indicate magnetic door sensors: D001, D002, D005 and D006.
(3) Identifiers starting with "L" and "LL" indicate light switches: L001–L005, LL001 and LL005.
(4) Identifiers starting with "LS" indicate light sensors: LS001–LS023.
(5) Identifiers starting with "M" indicate infrared motion sensors: M001–M022.
(6) Identifiers starting with “MA” indicate wide-area infrared motion sensors: MA003, MA009, MA010, MA013, MA014, MA020 and MA023.
(7) Identifiers starting with “T” indicate temperature sensors: T101–T105. Thirty activities were considered in the dataset. The raw dataset included 413,142 sensor events. The activity dataset included 2087 activity records. In our experiment, 12 activities were selected to test the proposed approach: "Sleep" ("S"), "Bathe" ("B"), "Dress" ("D"), "Eat_Breakfast" ("E_B"), "Eat_Dinner" ("E_D"), "Groom" ("G"), "Take_Medicine" ("T_M"), "Toilet" ("T"), "Wash_Breakfast_Dishes" ("W_B_D"), "Wash_Dinner_Dishes" ("W_D_D"), "Watch_TV" ("W_T") and "Work_At_Table" ("W_A_T"). In total, 951 activity records were used.

Table 2. Description of dataset “HH102”.

<table>
<thead>
<tr>
<th>Sensors</th>
<th>Activity</th>
<th>Raw Sensor Events</th>
<th>Raw Activity Records</th>
<th>Selected Activity</th>
<th>Selected Activity Records</th>
</tr>
</thead>
<tbody>
<tr>
<td>97</td>
<td>30</td>
<td>413,142</td>
<td>2087</td>
<td>12</td>
<td>951</td>
</tr>
</tbody>
</table>

The details of dataset “HH104” are displayed in Table 3. The measurement time was 61 days. A total of one hundred and thirty six sensors was installed in the apartment. The sensors included the following categories.

(1) Identifiers starting with “BA” indicate the sensor battery levels: BATP001–06, BATP101–106, BATV001–026, and BATV101–106.
(2) Identifiers starting with “D” indicate magnetic door sensors: D001–D006.
(3) Identifiers starting with “L” and “LL” indicate light switches: L001–L006.
(4) Identifiers starting with “LS” indicate light sensors: LS001–LS026.
(5) Identifiers starting with “M” indicate infrared motion sensors: M001–M013, M016 and M020–M026.
(6) Identifiers starting with “MA” indicate wide-area infrared motion sensors: MA014, MA015, MA017–MA019 and MA022.

Twenty-eight activities were analyzed in the dataset, which included 347,102 sensor events. The activity dataset included 3139 activity records. In our experiment, 12 activities were selected to test the proposed approach: “Sleep_Out_Of_Bed” ("S_O_O_B"), “Evening_Meds” ("E_M"), “Dress” ("D"), "Cook_Breakfast" ("C_B"), "Cook_Dinner" ("C_B"), “Phone” ("P"), “Take_Medicine” ("T_M"), “Toilet” ("T"), “Wash_Breakfast_Dishes” ("W_B_D"), “Wash_Dinner_Dishes” ("W_D_D"), “Morning_Meds” ("M_M") and “Work_On_Computer” ("W_O_C"). There were 1121 activity records.
Table 3. Description of dataset “HH104”.

<table>
<thead>
<tr>
<th>Sensors</th>
<th>Activity</th>
<th>Raw Sensor Events</th>
<th>Raw Activity Records</th>
<th>Selected Activity</th>
<th>Selected Activity Records</th>
</tr>
</thead>
<tbody>
<tr>
<td>136</td>
<td>28</td>
<td>347,102</td>
<td>3139</td>
<td>12</td>
<td>1121</td>
</tr>
</tbody>
</table>

5.2. Classifiers and Evaluation Metrics

In our experiment, the recognition model is based on one of five classifiers: naive Bayesian (NB), k-nearest neighbor (kNN), C4.5, random forest (RF) and the hidden Markov model (HMM) [17]. The details of the classifiers are shown in Table 4.

Table 4. Parameters of the classifiers.

<table>
<thead>
<tr>
<th>Classifiers</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB</td>
<td>numDecimalPlaces: 2</td>
</tr>
<tr>
<td>kNN</td>
<td>the number of neighbors to use: 1, distanceWeighting: No distance weight, nearestNeighbourSearchAlgorithm: LinearNNSearch</td>
</tr>
<tr>
<td>C4.5</td>
<td>confidenceFactor: 0.25, minNumObj: 2, numDecimalPlaces: 2</td>
</tr>
<tr>
<td>RF</td>
<td>numIterations: 100, numDecimalPlaces: 2</td>
</tr>
</tbody>
</table>
| HMM         | The set of sensor events is divided into a number of windows regarding a duration of 60 s. $S = \{S_1, S_2, \ldots, S_N\}$ is the set of states. Activities map to states. $V = \{V_1, V_2, \ldots, V_M\}$ is the set of observations. Sensors map to observations. $A = \{a_{ij} = P(q_{t+1} = S_j | q_t = S_i), 1 \leq i, j \leq N\}$ is a set of transition probabilities. Each $a_{ij}$ represents the probability of transition from state $S_i$ to $S_j$. $a_{ij}$ is solved by Formula (1). $C_{ij}$ of Formula (1) is the frequency of transition from state $S_i$ to $S_j$. States of two continuous windows map to a transformation. $B = \{b_{jk} = P(o_t = V_k | q_t = S_j), 1 \leq j \leq N, 1 \leq k \leq M\}$ are emission probabilities. Each $b_{jk}$ represents the probability of observation $V_i$ being emitted by $S_j$. $b_{jk}$ is solved by Formula (2). $E_j(V_k)$ of Formula (2) is the frequency that $V_i$ are fired for $S_j$. $\pi = \{\pi_i = P(q_1 = S_i), 1 \leq i \leq N\}$ is an initial state distribution. Each $\pi_i$ represents the probability that $S_i$ is a start state. $\pi_i$ is solved by Formula (3). $\text{Init}(i)$ of Formula (3) is the frequency of $S_i$ occurring as a start state. The first state $S_i$ of each window maps to an initial state.

\[
a_{ij} = \frac{c_{ij}}{\sum_{k=1}^{N} c_{ik}}, 1 \leq i, j \leq N \tag{1}
\]

\[
b_{jk} = \frac{E_j(V_k)}{\sum_{i=1}^{M} E_i(V_k)}, 1 \leq i \leq N, 1 \leq k \leq M \tag{2}
\]

\[
\pi_i = \frac{\text{Init}(i)}{\sum_{j=1}^{N} \text{Init}(j)}, 1 \leq i \leq N \tag{3}
\]
The evaluation metrics are accuracy, precision and F-measure, which are shown in Formulas (4)–(6), respectively. In Formulas (4)–(6), $Q$ is the number of activity labels; $TP_i$ is the number of true positives; $FP_i$ is the number of false positives; $FN_i$ is the number of false negatives; $TN_i$ is the number of true negatives. Each validation was taken as a 10-fold cross-validation.

$$\text{Accuracy} = \frac{\sum_{i=1}^{Q} TP_i}{Q}$$

(4)

$$\text{Precision} = \frac{\sum_{i=1}^{Q} TP_i}{TP_i + FP_i}$$

(5)

$$\text{F-Measure} = \frac{2 \times \text{Precision} \times \text{Accuracy}}{\text{Precision} + \text{Accuracy}}$$

(6)

5.3. Results from HH102

The accuracy, precision and F-measure of the activity recognition of classifiers NB, kNN, C4.5 and RF are shown in Figures 2–4, respectively.

“SS” had an average accuracy of 0.6 (NB), 0.66 (kNN), 0.7 (C4.5) and 0.75 (RF); an average precision of 0.53 (NB), 0.53 (kNN), 0.57 (C4.5) and 0.58 (RF); and an average F-measure of 0.51 (NB), 0.53 (kNN), 0.56 (C4.5) and 0.57 (RF). “TS” had an average accuracy of 0.74 (NB), 0.73 (kNN), 0.74 (C4.5) and 0.8 (RF); an average precision of 0.67 (NB), 0.66 (kNN), 0.68 (C4.5) and 0.78 (RF); and an average F-measure of 0.66 (NB), 0.65 (kNN), 0.67 (C4.5) and 0.7 (RF). “TS” had higher average accuracies, precisions and F-measures than “SS”. For individual activity, there were 9 (NB), 7 (kNN), 9 (C4.5) and 8 (RF) activities, so “TS” had higher accuracies than “SS”. There were 9 (NB), 7 (kNN), 8 (C4.5) and 11 (RF) activities, so “TS” had higher precisions than “SS”.

In addition, we compare the proposed approach with the HMM-based approach. The HMM is defined as five tuples $(S, V, A, B, \pi)$. The activities map to the states in HMM. Two continuous activity records map to a transformation from one state to another. For an activity record, $(s, f)$ maps to an observation, where $s$ is the sensor name, and $f$ is the frequency that $s$ occurs in the activity record. As shown in Figure 5, the HMM had the accuracy, precision and F-measure of 0.68, 0.64 and 0.66. “TS” had the average accuracy, precision and F-measure of 0.78, 0.76 and 0.77 for NB, kNN, C4.5 and RF. “TS” had a higher accuracy, precision and F-measure than HMM. For individual activities, there were nine activities for which “TS” had higher accuracies than HMM. There were seven activities for which “TS” had higher precisions than HMM.

Figure 2. Accuracies on dataset “HH102” by using classifiers NB, kNN, C4.5 and RF.
In addition, the Pearson correlation coefficient method was used to examine the correlation between the average duration of activities and the average accuracies of activity recognition. As shown in Figure 6, the correlations are 0.45 ("SS") and 0.36 ("TS") for dataset HH102. For activities longer than 25 min, their accuracies are highly correlated to activity durations. For activities shorter than 25 min, the duration of the activity is weakly related to the accuracy of activity recognition.
5.4. Results from HH104

The accuracies, precisions and F-measures of the activity recognition of classifiers NB, kNN, C4.5 and RF are shown in Figures 7–9, respectively.

“SS” had an average accuracy of 0.7 (NB), 0.78 (kNN), 0.79 (C4.5) and 0.83 (RF); an average precision of 0.5 (NB), 0.56 (kNN), 0.67 (C4.5) and 0.77 (RF); and an average F-measure of 0.5 (NB), 0.6 (kNN), 0.57 (C4.5) and 0.63 (RF). “TS” had an average accuracy of 0.79 (NB), 0.81 (kNN), 0.87 (C4.5) and 0.89 (RF); an average precision of 0.6 (NB), 0.62 (kNN), 0.78 (C4.5) and 0.75 (RF); and an average F-measure of 0.55 (NB), 0.6 (kNN), 0.7 (C4.5) and 0.72 (RF). “TS” had higher average accuracies, precisions and F-measures than “SS”. For individual activities, there were 9 (NB), 6 (kNN), 9 (C4.5) and 8 (RF) activities for which “TS” had higher accuracies than “SS”. There were 8 (NB), 8 (kNN), 9 (C4.5) and 9 (RF) activities for which “TS” had higher precisions than “SS”.

In addition, we compare the proposed approach with the HMM-based one. As shown in Figure 10, HMM had an accuracy, precision and F-measure of 0.57, 0.59 and 0.62, respectively. “TS” had an average accuracy, precision and F-measure of 0.63, 0.72 and 0.68, respectively, for NB, kNN, C4.5 and RF. “TS” had a higher accuracy, precision and F-measure than HMM. For individual activities, there were eight activities, so “TS” had higher accuracies than HMM. Additionally, there were nine activities, so “TS” had higher precisions than HMM.
As shown in Figure 11, the correlations are 0.24 (“SS”) and 0.1 (“TS”) for dataset HH104. For activities longer than 25 min, their accuracies are highly correlated with the activity durations. For activities shorter than 25 min, the duration of activity is weakly related to the accuracy of activity recognition.
5.5. Discussion

In this paper, we presented a new two-stage approach for activity recognition. In the first stage, the activity records are clustered by regarding temporal features. In the second stage, the activity records of each cluster are recognized by the classifier. It would be interesting to combine the proposed approach with current studies. The datasets were public and used in many current studies, and the used classifiers are common in the studies. Generally, the studies focused on feature selection and classifier optimization. Comparatively, the proposed approach focused on the recognition process. Hence, the idea of two stages can easily extend the studies to improve the feature selection, which was also shown with the experiment results.

The experiments show that “TS” performs better than “SS” and the Hidden Markov Model. The conclusion can also be drawn by analyzing the proposed approach. For the first stage, the temporal features are used to cluster the training set. The activity records among different clusters are temporal-sensitive, whereas the activity records within the same cluster are temporal-insensitive, and temporal features are noise data. For the second stage, the activity records within the same cluster are spatially sensitive and only recognized based on spatial features. Compared to the one-dimensional recognition model, the activity recognition of the second stage is not interfered by noise data and has high performance.

However, different individual activities had a large variation in performance. The activities with sufficient samples obtain high performance. In contrast, activities with a few samples obtain low performance because it is difficult for the classifiers to learn the valid features of activity from the samples. In addition, the activities with a notably small number of samples are probably clustered into different clusters, which makes each cluster have fewer samples. For dataset “HH102”, the activities “Sleep”, “Dress”, “Eat_Dinner”, “Toilet”, “Wash_Dinner_Dishes” and “Watch_TV” had high performance. For dataset “HH104”, the activities “Sleep_Out_Of_Bed”, “Dress”, “Cook_Breakfast”, “Cook_Dinner”, “Toilet”, “Morning_Meds” and “Work_On_Computer” had high performance. In contrast, few activities had poor performance. For dataset “HH102”, the activities “Bathe” and “Take_Medicine” had a notably small sample. “Bathe” had only 36 activity records. “Take_Medicine” had only 24 activity records. These are far fewer than the average number of 80. For dataset “HH104”, the activity “Take_Medicine” had a notably small sample. “Take_Medicine” had only 12 activity records. Fewer activity records imply that the features of the activities were difficult to be adequately obtained.

Although the same sensors were used in detecting some activities, the accuracies had also a large variation in performance. For instance, the activity “Eat_Breakfast” (“Wash_Breakfast_Dishes”) is harder to detect than “Eat_Dinner” (“Wash_Dinner_Dishes”). Compared to the activity “Eat_Dinner” (“Wash_Dinner_Dishes”), the activity “Eat_Breakfast” (“Wash_Breakfast_Dishes”) is temporally irregular. For instance, the activity “Wash_Breakfast_Dishes” sometimes happened at noon or in the evening.
6. Conclusions and Future Work

To better aid elderly people by using context-aware services, this paper proposes a two-stage approach for the activity recognition of a single resident. To validate the proposed model, the naive models were used for comparison in two experiments, each of which was performed to recognize twelve daily activities. The results show that the proposed approach outperforms the naive spatial model in terms of both accuracy of individual activities and average accuracy.

The two-stage approach demonstrates the superiority in recognizing single-resident activities. In the next stage, we will attempt to apply the proposed approach to more complex data situations, e.g., unlabeled data and multiple-resident situations. In this paper, the cluster algorithm is used to partition the set of activity records into two clusters. The number of clusters is vital to the performance of activity recognition. A supplementary algorithm for the cluster algorithm will be conceived. The supplementary algorithm is expected to output a reasonable number of clusters. In addition, we will study deeply the topic of how the sample size affects the calculation of the accuracy of an activity.

Acknowledgments: We thank all reviewers for their useful comments to improve the paper. This work was supported by the National Natural Science Foundation of China (No. 61672122, No. 61402070, No. 61602077, No. 61672261), the Natural Science Foundation of Liaoning Province of China (No. 2015020203), the Educational Commission of Liaoning Province of China (No. L2015060), the Fundamental Research Funds for the Central Universities (No. 3132016348) and the Open Project Program of Artificial Intelligence Key Laboratory of Sichuan Province (2016RY01).

Author Contributions: Yaqing Liu and Dantong Ouyang conceived of the research subject and contributed critical suggestions. Yong Liu conducted the experiments. Rong Chen drafted the paper. All authors have read and approved the final manuscript.

Conflicts of Interest: The authors declare no conflicts of interest.

References


© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0).