A Fully Unsupervised Approach to Activity Discovery

Umut Avci and Andrea Passerini

Dipartimento di Ingegneria e Scienza dell'Informazione
Università degli Studi di Trento
Trento, Italy
{avci,passerini}@disi.unitn.it

Abstract. We propose an activity discovery framework that aims at identifying activities within data streams in the absence of data annotation. The process starts with dividing the full sensor stream into segments by identifying differences in sensor activations characterizing potential activity changes. Then, extracted segments are clustered in order to find groups of similar segments each representing a candidate activity. Lastly, parameters of a sequential labeling algorithm are estimated using segment clusters found in the previous step and the learned model is used to smooth the initial segmentation. We present experimental evaluation for two real world datasets. The results obtained show that our segmentation approaches perform almost as good as the true segmentation and that activities are discovered with a high accuracy in most of the cases. We demonstrate the effectiveness of our model by comparing it with a technique using substantial domain knowledge.

Keywords: Activity discovery, sequence segmentation, sequential labeling

1 Introduction

Activity recognition has long been studied by the machine learning community. Most of the work in the field has focused on supervised approaches in order to train activity models. However, these require the availability of labelled sequences, an expensive and time consuming process. Furthermore, training data are specific to the setting involving the activities to be recognized and the persons involved, as the daily living habits change from an individual to another. Activity discovery aims at identifying activities within data streams in the absence of data annotation. Therefore, it can be used in any possible daily-life scenario. In health monitoring applications, for instance, one of the task is continuously checking the behaviour of a patient in order to determine whether his/her routines are maintained, regardless of the type of activities being performed. Inconsistencies in daily routines, i.e. changes in the structure of performed activities, can suggest problems in patient's health. As many unsupervised learning tasks, activity discovery is a challenging problem: many activities tend to share a similar set of

signals (e.g. kitchen sensors for food-related activities), short periods lacking any signal at all can occur during an activity, to be distinguished from truly "idle" periods where no activity is being performed. Finally the discovery needs to be robust enough to account for variations in the way activities can be performed.

There are few works on activity discovery in the literature. Using sequential patterns in order to represent activities was proposed in [4] and [2]. Both approaches are based on the idea that similar patterns can be used for representing the activities. Following this idea, patterns are first mined from data and then clustered by k-means and LDA respectively. However, these techniques are bound to the quality and coverage of the extracted patterns. Hamid et al. [5] suggests clustering segments instead of patterns extracted from them. For this purpose, segments are represented by histograms each of which then corresponds to a node in an edge-weighted graph. Maximal cliques in the graph are identified as activity candidates. However, the method cannot be used for real activity discovery as segments are assumed to be known in advance. Hong et al. [8] developed an activity discovery approach based on conceptual definitions of activities in terms of Evidential Ontology Networks (EON). A candidate segment that fits an EON best is recognized as the corresponding activity. Since the method works on previously segmented data, the authors later introduced three segmentation approaches [9]. Extracted segments are fed into the EON for determining activities. The method is based on a deep knowledge of the activities being searched, needed to compile the set of rules which are used for the activity discovery phase.

In this paper, we present an activity discovery approach that addresses the abovementioned limitations. The rationale behind the approach is that distinct activities should correspond to separate sets of sensors, e.g. activations of pairs or triplets of sensors, possibly repeated over time, jointly indicating a certain activity. With this reasoning, we assume that transition from one set to another indicates the possible time of an activity change. Following this rationale, we propose two segmentation algorithms looking for the change points in a sensor stream. Extracted segments are then clustered in order to find groups of similar segments each representing a candidate activity. Finally we use segments labeled with cluster identifiers to train the parameters of a sequential labeling algorithm which is then used for smoothing the initial segmentation.

We evaluate our technique in two freely available smart home datasets. The results show that our segmentation approaches perform almost as good as the true segmentation and that activities are discovered with a high accuracy in most of the cases. Comparisons with [8] show that our clustering achieves better results than an ontology network relying on expert-based activity descriptions.

The paper is organized as follows. Section 2 introduces the data representation format and the notion of activity segment. We describe our activity discovery framework in Section 3. An extensive experimental evaluation is reported in Section 4. Finally, conclusions are drawn in Section 5.

2 Data Representation

A dataset $\mathcal{D} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(d)}\}$ is a collection of input sequences for a number of days d. An input example $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$ consists of a consecutive sequence of observations, each covering a certain time instant t. An observation $\mathbf{x}_t$ is represented by the set of sensors which are active at that time instant (i.e. within its time interval). When feeding input sequences to labeling algorithms (see Section 3.3), observations will be represented as binary vectors rather than sets. Given N sensors, an observation $\mathbf{x}_t$ will thus be encoded as a binary feature vector $\mathbf{x}_t = (x_t^1, \ldots, x_t^N)$, each feature being 1 if the corresponding sensor is active and 0 otherwise.

The labeling task consists of predicting a sequence of activity labels $\mathbf{y} = \{y_1, \ldots, y_T\}$, one for each time instant. Each label $y_t \in [1, L]$ is one of L possible activities, with one indicating no activity. We assume here that activities are not simultaneous, i.e. only a single activity is performed at each time instant. We define an activity segment as a sequence of consecutive time instants labeled with the same activity. A segment $s_u = (b_u, e_u, y_u)$ is represented by its starting and ending time instants $b_u, e_u \in [1, T]$, with $e_u \geq b_u$, and the segment label y_u. A label sequence $\mathbf{y}$ can be split into a sequence $\mathbf{s} = \{s_1, \ldots, s_U\}$ of activity segments such that $b_1 = 1$, $e_U = T$, $b_u = e_{u-1} + 1$ and $y_u \neq y_{u-1}$ for all u. We define as $\mathbf{x}_{b_u:e_u}$ the segment of $\mathbf{x}$ ranging from b_u to e_u included. A collection of $\mathbf{s}$ over all the days forms $\mathcal{S} = \{\mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(d)}\}$ with d being the number of days. Note that in the activity discovery framework we assume no knowledge is available concerning labels, including the number of activities L.

3 Activity Discovery Framework

Our activity discovery process consists of three steps:

1. **Sequence segmentation**: in this step, the full sensor stream is partitioned into segments which represent candidate activities, i.e. each segment should approximately span the whole time horizon in which an activity is continuously conducted. The segmentation procedure scans the stream searching for *changepoints* suggesting a change in the activity being performed.
2. **Segment clustering**: once segment have been identified, a clustering algorithm is employed in order to group together similar segments, each group representing a distinct candidate activity. Designing an appropriate segment similarity measure is crucial here in order to boost performance.
3. **Sequential labeling**: the final step employs the segment clusters produced by the previous step to train the parameters of a sequential labeling algorithm. The learned model is then used to run inference on the full sensor stream obtaining the final segmentation output. The rationale of this component is that the learned probabilistic model should allow to smooth segment borders with respect to the segmentation and clustering output, possibly improving recognition accuracy.

In the following we detail each step of the process.

3.1 Sequence Segmentation

The aim of the segmentation phase is to partition the sensor stream into fragments so that each fragment characterizes the occurrence of an activity. As will be seen in Section 4, activity datasets used for the evaluation differ from each other by the number and the type of the sensors. The fact that the quality of the segmentation is highly depend on the dataset properties necessities a specific handling in algorithm design. For this purpose, we propose two novel approaches, distance-based and context-based segmentation, for each experimental setting.

Distance-based segmentation is based on the idea that an activity is related to the sensor events occurring within a specific range. More specifically, the consecutive activation of two sensors whose distance to each other is less than a threshold (ϕ) is likely to indicate the persistence of the same activity. For example, preparing dinner is typically characterized by activation of kitchen sensors. Any sensor event occurring in the bedroom, however, is probably unrelated to the dinner activity. Selection of the threshold can be done in a number of ways depending on the dataset. One can assume that every activity is bounded with a certain room in the apartment. In this case, the threshold is computed as the distance between the two closest rooms. We used this type of distance in Van Kasteren dataset (see Section 4) as activities are known to be performed in separate rooms.

```
Algorithm 1: Distance-based Segmentation
Input:
  Sequence of observations (D)
  Separation threshold (φ)
  Sensor coordinates (C)
Output:
  Candidate activity borders
begin
  Initialize border candidates (B) to the empty set
  Calculate Manhattan distance matrix (M) of sensor pairings with (C)
  Find active time instants T = (t_1, t_2, ..., t_m) in D
  for i from 1 to m-1
    Initialize pairwise distances (P) to the empty set
    for all sensor pairing (j,k) ∈ (D(t_i), D(t_{i+1}))
      P ← P ∪ M(j,k)
    if count(P > φ)/|P| > 0.5
        B ← B ∪ t_i
        if t_{i+1} − 1 ∉ T
            B ← B ∪ t_{i+1} − 1
  B ← B ∪ length(D)
  return B
```

Algorithm 1 shows the pseudocode of our distance-based segmentation technique. The algorithm takes as inputs a sequence of observations (D) as sensor activations and the spatial coordinates of all sensors (C), plus a threshold ϕ controlling when to introduce a breakpoint in the sequence. It first computes a

matrix M of pairwise distances between sensors using the Manhattan metric, as it provides a natural measure of walking path length. The algorithm then identifies all time instants having at least one sensor activation and iteratively processes each of them. In order to decide whether to introduce a breakpoint at active time instant t_i, the algorithm compares its active sensors with those of the next active time instant t_{i+1}, using the previously computed distance matrix. If more than half of the comparisons have a distance greater than the threshold ϕ, i.e. sensors from the two time instants tend to be far apart, a breakpoint is added at time instant t_i. Note that t_{i+1} is not necessarily the time instant immediately following t_i, as they can be separated by a sequence of time instants lacking any sensor activation, likely indicating an idle "activity". In this case the algorithm introduces an additional breakpoint at time instant $t_{i+1} - 1$, isolating the segment with no activations. Note that conversely, null segments separating two active time instants with spatially close active sensors are merged in the segment containing t_i and t_{i+1}. This can be reasonable as activities often include short periods with no activations, but can miss longer null segments potentially representing idle cases. At the end of this section we introduce a post-processing procedure addressing this problem. Our algorithm resembles the one in [9], and indeed the two produce a very similar segmentation, but the latter requires much more information, concerning the location where specific activities are performed and ad-hoc rules extracted via profound investigation of the sensor stream.

The distance-based segmentation approach is suitable for datasets in which the location information is closely related to the activity being performed. In many cases, on the other hand, information regarding the locations of the sensors is not available, which makes the proposed method inapplicable. Therefore a general approach that does not depend on any kind of knowledge is required. We thus propose the context-based segmentation in which change points in sensor activation patterns are extracted. The rationale behind the approach is that two activities should be related to two distinct patterns of sensor activations, e.g. pairs or triplets of sensors jointly activated in the same time instant. This is implemented by representing each time instant by the set of its up to n-grams, where an n-gram is a set of n sensors jointly active in the time instant. The similarity between two time instants with sets A and B is then computed using the Jaccard index:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

We extend this similarity to include the *context* of a time instant by defining a *frame* as a sequence of time instants of a certain length (τ) and considering similarity between frames. The similarity between time instants t_i and t_{i+1} is then computed considering the frames $[t_{i-\tau+1} : t_i]$ and $[t_{i+1} : t_{i+1+\tau}]$, representing each frame by the union of the sets of (up to) n-grams of its time instants, and computing the Jaccard index between the two frames. Algorithm 2 outlines the context-based segmentation approach, where frames exceeding the borders of the sequence are appropriately trimmed.

```
Algorithm 2: Context-based Segmentation
Input:
  Sequence of observations (D)
  Frame size (τ)
  Gram size (n)
Output:
  Candidate activity borders
begin
  Initialize border candidates (B) to the empty set
  L ← length(D)
  for i from 1 to L-1
    if Jaccard(n, D[t_{i−τ+1} : t_i], D[t_{i+1} : t_{i+τ}]) = 0
      B ← B ∪ i
  B ← B ∪ L
  return B
```

The border generation process may result with a number of segments larger than the true one. In order to fix the most obvious cases, we applied a pruning procedure based on a number of simple reasonings. (1) There are a few occasions in which the distance-based algorithm extracts segments of size one when two consecutive time instants have exactly the same active sensors, and these come from two different locations. For instance, toilet activity is interleaved with sleeping, and is characterized by sensor activations from both the bedroom (bedroom door) and the toilet (e.g. toilet flush). If this occurs in a row for a small number of time instants, we merge them together in a single segment. (2) Let a segment without any sensor activation (zero segment) be preceded and followed by two segments whose active sensors either occur in the same location or are similar. These three consecutive segments should be merged into one to represent a single activity or be kept separate as two distinct segments of the same activity separated by an idle segment, depending on the length of the zero segment in the middle. We choose the former option if the resulting segment is smaller than a threshold representing the typical duration of activities, and the latter otherwise. The threshold is computed as the average length of the segments obtained by the segmentation algorithms, after excluding very long segments which likely represent peculiar activities like sleeping or idle. As an example of the merge operation, consider a dinner activity as taking food from the fridge followed by heating food in the microwave and then eating. The time we waited for heating is a zero segment, yet belongs to the same dinner activity. A three minute toilet activity performed two hours after another occurrence of toilet activity, on the other hand, should not be merged with the previous one since it is clear that they are distinct activity occurrences separated by another activity (e.g. sleeping).

3.2 Segment Clustering

The purpose of the clustering step is to determine the intrinsic grouping of the segments extracted in the previous phase so that each group represents an activity. We represent segments in terms of histograms of time instant-based

features collected over each segment. Time instant-based features are extracted in the same manner as we did in the context-based segmentation, i.e. up to n-grams of sensors are extracted for each time instant in the input sequence. Found features are then collected over the segments and used for creating histograms as the counts of each feature.

Segment representations are then fed to a clustering algorithm. This has to deal with high-dimensional data, as coming from the up to n-gram feature representation, and automatically identify the number of clusters, which is not known in advance. We rely on the HDDC method [1] which satisfies both requirements. It is based on a modified Gaussian Mixture Model with a dimensionality reduction technique which determines the specific subspace in which each class is located by using eigenvectors of the covariance matrix. Models representing the subspaces are used to choose the number of clusters. To this end, clustering results are computed for different number of clusters and different models and the one maximizing Bayesian Information Criterion is selected. Further details can be found in the original paper [1].

3.3 Sequential labeling

In principle, our algorithm could end up with the groupings returned by the clustering algorithm, each group representing a candidate activity. However, both segmentation and clustering steps are prone to errors and only provide approximations of actual segments and true groups. We use these approximations to train a sequential labeling algorithm, which assigns a label to each time instant in the sequence. The learned model is then used to run inference on the full sensor stream, providing the final sequential labeling. Each cluster in our setting corresponds to a different label in the sequential model.

We employ a Hidden Semi-Markov model (HSMM) [10] as sequential labeling approach, which is appropriate to label sequences where consecutive time instants tend to share the same label. An HSMM is a variant of HMM which allows for explicit duration distributions for different states, making it especially useful for segmenting sequences into fragments, each characterized by the same label. The joint probability modelled by the HSMM is represented as:

$$p(\mathbf{x}, \mathbf{s}) = \prod_{u=1}^{U} p(y_u|y_{u-1}) p(d_u|y_u) p(\mathbf{x}_{b_u:e_u}|y_u) \tag{1}$$

where $d_u = e_u - b_u + 1$ is the duration of segment s_u. The transition probability $p(y_u|y_{u-1})$ models the probability that activity y_u directly follows y_{u-1}. The duration probability $p(d_u|y_u)$ can be modelled as a histogram distribution (see e.g. [7]), where candidate activity durations are grouped into a certain number of bins (5 in our experiments). The probability of a certain segment of sensor activations given its label is commonly computed as the product of the probabilities of its time instants:

$$p(\mathbf{x}_{b_u:e_u}|y_u) = \prod_{t=b_u}^{e_u} p(\mathbf{x}_t|y_u) \tag{2}$$

where $p(\mathbf{x}_t|y_u)$ is further decomposed as a product of Bernoulli probabilities (active vs inactive) over sensors:

$$p(\mathbf{x}_t|y_u) = \prod_{i=1}^{N} p(x_t^i|y_u) \tag{3}$$

where N is the number of sensors. Given that each label is associated with a cluster from the previous step, parameters of all probabilities can be readily estimated from counts over the cluster segments. The transition probability between labels y_u and y_v, for instance, can be computed as the fraction of times in which a segment from cluster y_v follows one from cluster y_u in the original sequence, with respect to the overall number of segments in cluster y_u.

Once parameters are learned from the clustering results, inference is run on the whole sequence providing the labelled segmentation with maximal probability:

$$\mathbf{s}^* = \text{argmax}_{\mathbf{s}}\ p(\mathbf{x}, \mathbf{s}) \tag{4}$$

which can be efficiently computed by the well-known Viterbi algorithm [10].

4 Experiments

In this section we first present the experimental setup used for evaluating the proposed approach and then provide results of the experiments.

4.1 Setting

The performance of the proposed framework was evaluated on a collection of freely available[1][2] benchmark datasets. Van Kasteren's dataset was collected over 28 days in a three room apartment occupied by a single resident [6]. The dataset consists of 14 state-change sensors, e.g reed switches and passive infrared. Activities were annotated by recording the start and end time of the corresponding activity either via handwritten diary or bluetooth headset. CASAS dataset differs from the previous one as two residents are simultaneously monitored in an apartment for 46 days, with a sensor network (of 71 nodes) mainly composed of motion and utility usage sensors [3]. Annotators labeled the data using a 3D visualization tool and residents' diaries.

We used the Changepoint feature representation that considers a sensor active (value 1) only in the time instants in which it alters its state [6]. It is robust

[1] https://sites.google.com/site/tim0306/kasterenDataset.zip
[2] http://ailab.wsu.edu/casas/datasets/twor.2009.zip

against noises and is capable of tolerating the dataset specific sensor failures. For example, a door that is left open after completion of an activity causes sensors to be active for longer periods than the actual activity duration, which eventually damages the segment information of other activities. Changepoint representation eliminates such activations by only considering activation and deactivation times of the sensors.

The performance of the system was evaluated by using the class accuracy metric proposed in [6]. The measure represents the average percentage of correctly classified timeslices per class and provides a proper model performance when a dataset contains unbalanced classes in terms of appearance frequency. The class accuracy is computed as follows:

$$\text{Class: } \frac{1}{C}\sum_{c=1}^{C}\frac{\sum_{n=1}^{N_c}[inferred_c(n) = true_c(n)]}{N_c} \tag{5}$$

where $[a = b]$ is a binary indicator returning 1 when true and 0 otherwise. C is the number of classes and N_c is the total number of time slices for class c. Precision, recall, and F measure are omitted for space limitations.

4.2 Results

Tables 1, and 2 show confusion matrices computed from the class accuracies for the van Kasteren and the CASAS dataset respectively. As discussed in Section 3.1, we applied the distance-based segmentation to the former and the context-based one to the latter. In all experiments the maximum gram size was set to two as higher values provided similar results while increasing computational complexity. Clustering does not assign names to the detected groups. However, it is clear that if a cluster contains mostly segments corresponding to a certain activity, it can be considered as an approximation of that activity. In order to identify the most likely activity for each cluster, we try all possible distinct activity assignments to clusters and choose the one maximizing class accuracy. If the algorithm identifies more clusters than the true number of activities, the best assignment will assign the clusters in excess to a dummy wrong activity. We do not explicitly report dummy clusters in the Tables, but include their assignments when computing the percentage of correct predictions (i.e. predicted rows do not always sum to one). Note that this best assignment measure is a fair evaluation procedure, as we simply identify for each cluster which is the activity it is most likely representing, forcing each cluster to represent a distinct activity.

The first set of experiments aims at comparing our approach with the evidential ontology network (EON) model proposed in [8] and evaluated on the Van Kasteren dataset. Table 1 shows confusion matrices for the different activities, where rows indicate true activities and column predicted ones. `EON` rows report results for the EON model [8], while `CLU` shows results of our clustering step applied to the true segmentation, the same setting used in [8]. Our approach

Table 1: Detailed results of van Kastaren Dataset (values as percentages)

		I.	L.	T.	Sh.	Sl.	B.	Din.	Dr.
Idle	EON	72	4	13	3	5	0	3	0
	CLU	100	0	0	0	0	0	0	0
	SEG+CLU+HSMM	74(74)	12(1)	3(4)	0(2)	8(3)	2(2)	1(2)	0(0)
Leaving	EON	0	74	11	0	14	1	0	0
	CLU	0	80	20	0	0	0	0	0
	SEG+CLU+HSMM	4(41)	96(59)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)
Toileting	EON	0	56	27	5	1	11	0	0
	CLU	9	0	76	6	9	0	0	0
	SEG+CLU+HSMM	4(13)	1(1)	78(32)	5(4)	4(4)	0(1)	0(0)	0(10)
Showering	EON	0	0	0	100	0	0	0	0
	CLU	0	0	0	100	0	0	0	0
	SEG+CLU+HSMM	9(15)	8(0)	1(0)	76(85)	0(0)	0(0)	0(0)	0(0)
Sleeping	EON	0	0	0	0	100	0	0	0
	CLU	62	0	3	7	28	0	0	0
	SEG+CLU+HSMM	35(27)	21(0)	0(1)	0(0)	44(44)	0(0)	0(0)	0(24)
Breakfast	EON	0	21	14	0	4	44	14	3
	CLU	0	0	0	0	0	100	0	0
	SEG+CLU+HSMM	18(31)	0(0)	5(0)	0(0)	9(1)	68(68)	0(0)	0(0)
Dinner	EON	0	0	59	0	13	0	28	0
	CLU	7	0	0	0	0	57	36	0
	SEG+CLU+HSMM	23(12)	2(0)	0(0)	0(0)	0(0)	36(35)	38(53)	0(0)
Drink	EON	37	0	0	0	0	0	0	63
	CLU	14	0	0	0	0	0	0	86
	SEG+CLU+HSMM	9(16)	1(0)	1(5)	0(0)	0(0)	80(64)	8(15)	0(0)

outperforms the competitor[3] in six out of eight activities and is on par on one. The only case where we get worse results is on Sleeping, which is characterized by a single sensor activation of bedroom door and a long period of no sensor activation. Sleeping is divided into many parts as it is interleaved by the Toilet activity. Segments separating two consecutive Toilet activities do not have any sensor activation, and are thus wrongly clustered in the Idle group[4].

`SEG+CLU+HSMM` rows report results of our complete approach, while results in brackets show the performance of the segmentation and clustering steps only (`SEG+CLU`). Clustering generates nine clusters for both true segmentation and predicted one. Activities performed in the kitchen are clustered together in predicted segmentation as they share basically the same sensor activations, explaining the overprediction of Breakfast. Drink, however, is assigned to a distinct cluster in true segmentation which decreased the confusion in prediction of kitchen-oriented activities. Incorporating sequential labeling (`SEG+CLU+HSMM` rows) provides overall better results, by improving recognition of Leaving and Toileting (while performance for Showering and Dinner are slightly degraded).

[3] Note that by substantially extending the knowledge concerning activities being searched it is possible to achieve much higher recognition accuracy [8]. However, our aim here is to perform activity detection without any specific knowledge on the activities being performed.

[4] [9] Hong et al., Segmenting sensor data for activity monitoring in smart environments

Table 2: Detailed results of CASAS Dataset (values as percentages)

		I.	B.to.T.	B.	G.	S.	W.at.C.	W.at.D.R.
Idle	CLU	100	0	0	0	0	0	0
	SEG+CLU+HSMM	100(83)	0(7)	0(6)	0(0)	0(0)	0(0)	0(0)
Bed to Toilet	CLU	68	0	0	31	1	0	0
	SEG+CLU+HSMM	28(9)	0(38)	0(1)	70(33)	1(9)	1(3)	0(0)
Breakfast	CLU	7	0	93	0	0	0	0
	SEG+CLU+HSMM	6(29)	0(3)	92(35)	2(7)	0(0)	0(0)	0(0)
Grooming	CLU	12	0	0	88	0	0	0
	SEG+CLU+HSMM	9(6)	0(30)	0(5)	91(41)	0(8)	0(1)	0(0)
Sleeping	CLU	2	0	0	40	58	0	0
	SEG+CLU+HSMM	6(3)	0(3)	0(2)	0(1)	94(51)	0(4)	0(0)
Working at computer	CLU	3	13	0	0	0	82	2
	SEG+CLU+HSMM	12(10)	0(1)	0(6)	0(1)	0(2)	86(36)	2(3)
Working at dining room	CLU	0	0	0	0	0	38	62
	SEG+CLU+HSMM	7(34)	0(0)	0(0)	0(2)	0(0)	8(8)	45(23)

For Leaving, the improvement is achieved by recovering from incorrect segmentations introducing spurious segments predicted as Idle within a Leaving activity. For Toilet, the clustering algorithm actually spreads segments containing Toilet activities in two different clusters, together to some segments from other activities, while the HSMM manages to identify most of them as belonging to the same class.

Table 2 reports results of `CLU`, `SEG+CLU+HSMM` and (in brackets) `SEG+CLU` for resident one of the CASAS dataset. A similar behaviour is observed for resident two (results omitted for space limitations). The clustering algorithm generates 10 clusters. The complete model provides significant improvements over `SEG+CLU` in almost all cases. This is mostly due to recovering portions of segments which were assigned to spurious clusters (clustering detects more clusters than the true number of activities), thanks to the smoothing effect of HSMM and its capacity of correctly modeling duration of activities. This allows the complete model to even slightly improve over the clustering applied to the true segmentation, as shown by comparing rows `CLU` and `SEG+CLU+HSMM`. As for the Van Kasteren dataset, a current limitation of the approach is that similar activities tend to be merged into the same cluster. Indeed, both `CLU` and `SEG+CLU+HSMM` fail to identify the Bed to Toilet activity, as it is very similar to Grooming in terms of sensor activations involved.

5 Conclusion

In this paper, we presented an activity discovery framework that identifies activities in sensor streams without requiring data annotation. The proposed approach first extracts segments from sequences by identifying candidate activity changepoints, clusters extracted segments into groups representing candidate activities, and trains a sequential labelling model with segment clusters, which is then used to provide the final refined segmentation.

The effectiveness and suitability of our approach was evaluated in two smart home datasets. Initial results shows that proposed approach succeeds in discovering activities in many situations. Although our technique does not depend on any assumptions on dataset, e.g. type of activities, number of clusters, it outperformed a method using activity definitions as domain knowledge. We observed that our segmentation algorithm produces segments which are quite close to the true ones. The final sequential labeling model succeeds in further refining the results, by smoothing segment borders and recovering part of the segments assigned to spurious clusters.

The proposed framework, however, suffers from a number of limitations. Similar activities tend to be clustered together and are hard to distinguish. In order to prevent this, a better way to represent segments or additional features (e.g. time of the day, duration of the activity etc.) can be defined. Interleaved activities also decrease performance, as when repeatedly going to toilet during the night. Relationships between neighbouring segments could be included in the clustering phase in order to address this problem.

Acknowledgments. This research was partially supported by grant PRIN 2009LNP494 (Statistical Relational Learning: Algorithms and Applications) from Italian Ministry of University and Research.

References

1. Berg, L., Bouveyron, C., Girard, S.: HDclassif: An R Package for Model-Based Clustering and Discriminant Analysis of High-Dimensional Data. Journal of Statistical Software. 46-6, 1–29 (2012)
2. Chikhaoui, B., Wang, S., Pigot,H.: ADR-SPLDA: Activity discovery and recognition by combining sequential patterns and latent Dirichlet allocation. Pervasive and Mobile Computing. 8-6, 845-862 (2012)
3. Cook, D.J., Schmitter-Edgecombe, M.: Assessing the quality of activities in a smart environment. Methods of Information in Medicine. 48-5, 480-485 (2009)
4. Rashidi, P., Cook, D.J., Holder L.B., Schmitter-Edgecombe, M.: Discovering Activities to Recognize and Track in a Smart Environment. IEEE Transactions on Knowledge and Data Engineering. 23, 527-539 (2011)
5. Hamid, R., Maddi, S., Johnson, A., Bobick, A., Essa, I., Isbell, C.: A novel sequence representation for unsupervised analysis of human activities. Artificial Intelligence. 173-14, 1221–1244 (2009)
6. van Kasteren, T., Noulas, A., Englebienne, G., Kröse, B.: Accurate activity recognition in a home setting. In: 10th international conference on Ubiquitous computing, pp 1–9 (2008)
7. van Kasteren, T., Englebienne, G., Kröse, B.: Activity Recognition Using Semi-Markov Models on Real World Smart Home Datasets. Ambient Intelligence and Smart Environments thematic issue on Smart Homes. 2-3, 311-325 (2010)
8. Hong, X., Nugent, C.D.: Implementing evidential activity recognition in sensorised homes. Technology and Health Care. 19-1, 37-52 (2011)
9. Hong, X., Nugent, C.D.: Segmenting sensor data for activity monitoring in smart environments. Personal Ubiquitous Computing. 17-3, 545-559 (2013)
10. Yu, S.Z.: Hidden semi-Markov models. Artificial Intelligence. 174-2, 215-243 (2010)