A Unified Model for Human Behavior Modeling Using a Hierarchy with a Variable Number of States

Conference Paper · August 2014

DOI: 10.1109/ICPR.2014.653

5 authors, including:

Tim van Kasteren
Schibsted Media Group
36 PUBLICATIONS **767** CITATIONS
SEE PROFILE

Maria Niessen
Volkswagen AG
21 PUBLICATIONS **234** CITATIONS
SEE PROFILE

Andreas Merentitis
AGT International
31 PUBLICATIONS **171** CITATIONS
SEE PROFILE

Cem Ersoy
Bogazici University
187 PUBLICATIONS **2,931** CITATIONS
SEE PROFILE

A Unified Model for Human Behavior Modeling using a Hierarchy with a Variable Number of States

Hande Alemdar*, T. L. M van Kasteren†, M. E. Niessen†, A. Merentitis† and Cem Ersoy*
*NETLAB, Department of Computer Engineering
Boğaziçi University, Istanbul/Turkey
Email: {hande.alemdar,ersoy}@boun.edu.tr
†AGT International
Hilpertstrasse 35 64295 Darmstadt, Germany,
Email: {tkasteren,mniessen,amerentitis}@agtinternational.com

***Abstract*—Human behavior modeling enables many applications for smart cities, smart homes, mobile phones and other domains. We present a hierarchical hidden Markov model for human activity recognition that uses semi-supervised learning to automatically learn the model parameters using only labeled data of the top-layer of the hierarchy. This significantly reduces the annotation requirements for such a model and simplifies the design of such a model, since the inherent structure of the activity is automatically learned from data. The design consideration that remains is the number of states used for representing the actions that an activity consists of. Using multiple real world datasets we show that the same model works both for the recognition of activities of daily living in a smart home and for recognizing office activities from audio data. We show how a variable number of action states per activity can result in a significant increase in performance over using a fixed number per activity. Finally, we show how the use of Bayesian and Akaike information criterion results in models using a sub-optimal set of action states, since a model using intuitively chosen set states is able to outperform them.**

I. Introduction

In an increasingly sensor equipped world, human behavior modeling enables many applications in various domains: Smart cities can support authorities and citizens in managing a safer and more secure environment; Smart homes allow independent living for elderly and provide added comfort to our daily lives [1]; Mobile phones provide useful utilities that help us in our daily operations [2].

Human behavior contains rich hierarchical structure and previous work has shown that modeling this structure can benefit the recognition of human activities from sensor data [3]. However, the added complexity that a hierarchy brings can make the construction of an accurately fitting hierarchical model challenging, while the additional layers of representation can require additional annotation efforts for supervised learning methods. This makes it makes it more difficult to deploy such models in different configurations and environments, which limits their applicability.

In this work, we assume that a human activity can be broken into a set of actions that represent more atomic events of the behavioral routine. For example, an activity like cooking might consist of an action 'cutting vegetables' and an action 'frying them in a pan'. Our proposed hierarchical model learns the model parameters using a semi-supervised learning method that requires labeled data for the activities, but not for actions. The actions in the model are only used for recognition purposes, so we can remain agnostic about the interpretation of the actions that the learning method allocates. The only design consideration is the number of states used to represent the actions that make up each activity.

In our previous work, we have shown how a fixed number of actions for each activity results in an increase in recognition performance [4], [5]. In this paper, we show that a variable number of actions per activity can further improve the recognition performance. We evaluate the model performance on multiple real world datasets and show that the same model works both for the recognition of activities of daily living in a smart home and for recognizing office activities from audio data. Our results demonstrate that the same hierarchical model can be applied on multiple sensing modalities and can serve as a unified model for behavior modeling. The use of a unified model can be very beneficial for the large scale deployment of a solution. Techniques such as transfer learning can help obtain accurate parameters with limited annotation efforts, while a single code base makes maintenance easier and cost-effective.

The rest of the paper is organized as follows. In Section II, we discuss related work in the area of human behavior modeling and hierarchical modeling. In Section III, we describe our proposed hierarchical model and two methods for selecting the number of action states. Section IV introduces the smart-home and audio activity recognition scenarios and Section V discusses the results on multiple real-world datasets. Finally, in Section VI, we conclude.

II. Related Work

Human behavior modeling using different modalities of sensing has been an active research topic recently. The data were obtained from either ambient sensors deployed in the environment such as video [6], [7], audio [3], [8], and binary sensors [9], [10] or wearable sensors deployed on the body such as accelerometers and gyroscopes [11], [2]. Although there are different modalities of sensing, in terms of modelling of human activities, temporal probabilistic models such as hidden Markov models (HMMs) and conditional random fields (CRFs) have been shown to give better results with their ability of modeling the temporal dependencies and sequential nature of human activities.

Despite the powerful temporal modeling abilities, the flat versions of these models often fail to accurately model the complex nature of human activities with a variety of possible ways of performing the activity and with different interactions with the environment. Therefore, hierarchical models were used to obtain a more grained model for complex human activities.

The Hierarchical HMM (HHMM) is a generalization of the HMM that can have a hierarchical structure and is introduced by [12] for modeling complex multi-scale structure in sequential data. The original inference algorithm provided by Fine et al. has cubic time complexity in terms of the sequence length which prevented it to be applied to domains where the sequences are long. Murphy el al. [13] showed that the HHMM can be represented as a dynamic Bayesian network (DBN) with a linear time inference complexity with respect to the sequence length. This much simpler and more efficient inference algorithm has made the hierarchical models good candidates for modeling the data in many different domains, such as natural language processing, handwriting recognition and human activity modeling.

There are several studies that use hierarchical models in human activity recognition. van Kasteren et al. [4] proposed a two layer hierarchy where the top layer represents the human activities of daily living and the second layer are the several actions made during the course of the actual activity. The experiments on three real world smart home datasets reveal that the use of two or three action clusters per activity gives the best performance.

Karaman et al. [14] use two level hierarchical model with multimodal audio and video data in order to classify human activities. The semantic activities are encoded in the top-level followed by a bottom level HHM that models an activity with a number of non-semantic states. They experimented with 3, 5 or 7 sub-states and reported that using 3 non-semantic sub-states yields better performance.

While the previous studies already showed the improvement over the flat HMM models, they use an equal and fixed number of states in the second layer of the hierarchy. Therefore, they assume the same level of complexity for every activity at the top layer. However, it is very likely that the complexity of different activities varies. For sleeping activity, 1 or 2 states may be sufficient whereas preparing a meal requires much more complicated interactions with the environment and therefore it requires more states to be accurately modelled. Therefore, the ideal number of states for each top layer activity should be decided separately.

Celeux and Durand [15] proposed using penalised cross-validated likelihood criteria to determine the number of hidden states. They compare the performance of several information criteria such as AIC (Akaike's Information Criteria), BIC (Bayesian Information Criteria), PML (Penalised Marginal Likelihood) and ICL (Integrated Complete Likelihood) using simulated data. According to the results, AIC, BIC and ICL were observed having similar behaviour. They also state that AIC has a tendency to underpenalise the complexity of a model, ICL favours models that give rise to partitioning the data with the greatest evidence from the hidden states, and BIC performs well only if a HMM gives a representation of the observed process. PML converges very slowly to the optimal solution. Moreover, in practical situations, it seems to have a high tendency to overpenalise the complexity of a HMM model when the sequence length is not very large.

III. Hierarchical HMM with Variable Number of States

In this section, we first describe the hierarchical model we use for behavior modelling followed by our proposed method for selecting the sub-states within an activity.

A. Hierarchical HMM

Our model for activity recognition is a two-layer hierarchical hidden Markov model (Fig. 1). The top layer state variables y_t represent the activities and the bottom layer variables z_t represent the action clusters. Each activity consists of a sequence of action clusters and the temporal ordering of these action clusters can vary between different executions of an activity. The last action cluster of the sequence signifies the end of an activity and indicates the start of a new sequence of action clusters. This information is captured by the finished state variable f_t, which is used as a binary indicator to indicate that the bottom layer has finished its sequence.

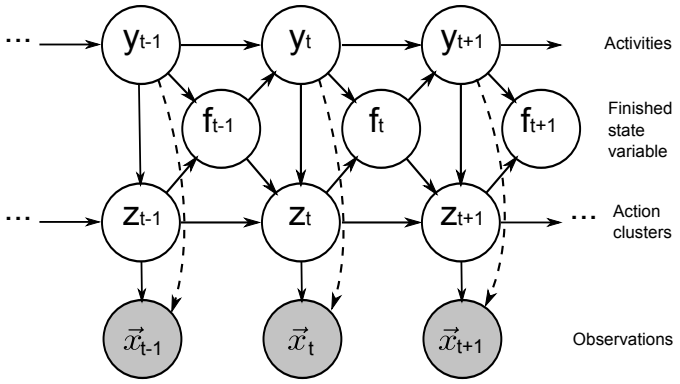

Fig. 1. The graphical representation of a two-layer HHMM. Shaded nodes represent observable variables, the white nodes represent hidden states. The dashed line is an optional dependency relation; we can choose to model the observation probability as $p(\vec{x}_t \mid y_t, z_t)$ or as $p(\vec{x}_t \mid z_t)$.

The joint probability distribution of the model factorizes as follows:

$$p(\mathbf{y}_{1:T}, \mathbf{z}_{1:T}, \mathbf{f}_{1:T}, \mathbf{x}_{1:T}) = \prod_{t=1}^{T} p(\vec{x}_t \mid y_t, z_t) p(y_t \mid y_{t-1}, f_{t-1}) \\ p(z_t \mid z_{t-1}, y_t, f_{t-1}) p(f_t \mid z_t, y_t)$$

where we have defined $p(y_1 \mid y_0, f_0) = p(y_1)$ and $p(z_1 \mid z_0, y_1, f_0) = p(z_1 \mid y_1)$ for the sake of notational simplicity. The entire model consists of a set of parameters $\theta = \{\pi_0, \pi_{1:Q}, A_0, A_{1:Q}, B, \phi\}$. These parameters are learned in a semi-supervised way by using the expectation-maximization (EM) algorithm. The initial state parameters π and transition parameters A exist for both the top layer and bottom layer states. To distinguish between these two types of parameters, we include a 0 in the subscript to indicate that a parameter is of the top layer and an index of 1 to Q for each of the bottom layer parameters. The distributions of the bottom layer

states depend on which top layer state the model is in and so there is a separate set of bottom layer state parameters for each possible top layer state, with Q being the number of top layer states. For example, if the model at one point is in the top state $y_t = k$, then the transition parameter A_k is used for the bottom layer state transitions. We now provide a detailed explanation of each of the factors that make up the joint probability and how they are parameterized.

At the first timeslice, the initial state distribution of the top layer states is represented by a multinomial distribution which is parameterized as $p(y_1 = j) = \pi_0(j)$. This top layer state generates a bottom layer state, also represented by a multinomial distribution and parameterized as $p(z_1 = j \mid y_1 = k) = \pi_k(j)$.

The factor $p(z_t = j \mid z_{t-1} = i, y_t = k, f_{t-1} = f)$ represents the transition probabilities of the bottom layer state variable. These transitions allow us to incorporate the probability of a particular temporal order of action clusters with respect to a given activity. A transition into a new state z_t, depends on the previous bottom layer variable z_{t-1}, the current top layer state variable y_t and the finished state variable f_{t-1}. Two distributions make up this factor, depending on the value of the finished state variable f_{t-1}. If in the previous timeslice the bottom layer state sequence ended ($f_{t-1} = 1$), a new sequence of bottom layer states starts at this timeslice and therefore the top layer state generates a bottom layer state using the same distribution as we saw at the first timeslice, parameterized by the set of parameters $p(z_t = j \mid z_{t-1} = i, y_t = k, f_{t-1} = f) = \pi_k(j)$. In case the bottom layer state sequence did not end ($f_{t-1} = 0$), a transition to a new bottom layer state is made using the transition matrix parameterized as $p(z_t = j \mid z_{t-1} = i, y_t = k, f_{t-1} = f) = A_k(i, j)$. These two cases can be compactly formulated as:

$$p(z_t = j \mid z_{t-1} = i, y_t = k, f_{t-1} = f) = \begin{cases} A_k(i,j) & \text{if } f = 0 \\ \pi_k(j) & \text{if } f = 1 \end{cases}$$

Transitions of the top layer state variables are represented by the factor $p(y_t = j \mid y_{t-1} = i, f_{t-1} = f)$. This factor is similar to the transition distribution of an HMM, except that it also depends on the finished state variable f_{t-1}. This dependency is important because it restricts the model in transitioning to a different top layer state as long as the bottom layer state sequence has not finished. When a bottom layer state sequence did not finish, the top layer state variable continues into the next timeslice with the same state value ($y_t = y_{t-1}$). Once the bottom layer state sequence has ended, a transition of the top layer state is made according to a transition matrix parameterized as $p(y_t = j \mid y_{t-1} = i, f_{t-1} = f) = A_0(i, j)$.

These two cases can be compactly formulated as:

$$p(y_t = j \mid y_{t-1} = i, f_{t-1} = f) = \begin{cases} \delta(i,j) & \text{if } f = 0 \\ A_0(i,j) & \text{if } f = 1 \end{cases}$$

where $\delta(i, j)$ is the Kronecker delta function, giving 1 if $i = j$ and 0 otherwise.

The probability of a bottom layer state sequence finishing is represented by the factor $p(f_t = f \mid y_t = j, z_t = l)$. This factor depends on both the bottom layer state z_t and the top layer state y_t. Even though the variable f_t indicates whether z_t is a finishing state, it is important that the distribution is also conditioned on the top layer state y_t. This is because the probability of a particular action cluster being the last action cluster for that activity can differ among activities. The factor is represented using a binomial distribution, parameterized as $p(f_t = f \mid y_t = j, z_t = l) = \phi_f(j, l)$.

Different observation models can be plugged into the model, to allow the model to process different sensing modalities. We present a Gaussian observation model and a Bernoulli observation model that are used in our audio and smart home experiments, respectively.

1) Gaussian observation Model: Using a multidimensional Gaussian distribution, each sub-event cluster is associated with a single Gaussian distribution $p(\vec{x}_t \mid y_t = k,\ z_t = l) = \mathcal{N}(x_t \mid \mu_{kl},\ \Sigma_k)$. Note that the covariance matrix Σ_k only has a subscript k, meaning that we have a different covariance matrix for each sound event, but that the covariance matrix for different sub-event clusters is shared among the sub-event clusters for a particular sound event k.

2) Bernoulli observation Model: Using independent Bernoulli distributions, each sensor corresponding to one Bernoulli distributions. This factorizes as $p(\vec{x}_t \mid y_t, z_t) = \prod_{n=1}^{N} p(x_n \mid y_t, z_t)$, with $p(x_n \mid y_t = j, z_t = k) = \mu_{jkn}^{x_n}(1 - \mu_{jkn})^{(1-x_n)}$. The observation parameters are collectively represented by a variable $B = \{\mu_{jkn}\}$.

B. Model Selection for Sub-States

In order to estimate the number of hidden states in an HMM, many criteria have been proposed that use a penalty term together with the data likelihood. Since it is possible to increase the likelihood by adding more parameters, using only the model likelihood may result in overfitting. Therefore, many of the proposed criteria trade off the data likelihood L with model complexity m in order to find the optimum number of states. We experiment with the two mostly used criteria: Bayesian Information Criterion (BIC) and Akaike Information Criterion (AIC). BIC and AIC resolve the overfitting problem by introducing a penalty term for the number of parameters in the model. BIC further uses the sample size in penalty term, thus the penalty term is larger in BIC than in AIC.

More formally, given a set of models, the model that has the minimum value of Eqn. 1 is the one to be preferred when using AIC. Similarly, when using BIC, the model that has the minimum value of Eqn. 2 is preferred.

$$AIC = -2logp(x \mid \theta) + 2m \tag{1}$$

$$BIC = -2logp(x \mid \theta) + mlog(n) \tag{2}$$

where $L = logp(x \mid \theta)$ is the data likelihood, m is the number of free parameters and n is the length of the sequence.

We find the optimum number of sub-states as follows. For each activity, we take all occurrences as different data sequences. We denote the total number of sequences as K. We then experiment with different models having different number of states starting from 1 up to 10 using leave one out cross

validation. We first learn the model parameters for a given model using the $K-1$ sequences, then we calculate the data likelihood L on the remaining test sequence and use the test likelihood while calculating AIC and BIC values. Then we select the model with the minimum AIC or BIC.

IV. Experiments

Our experiments aim to answer two questions 1) Can hierarchical models be used with data from different modalities using the same unified model for human behavior modelling? 2) Does allowing different levels of complexity for different activities increase the recognition performance? We first experiment with a flat HMM and with hierarchical HMMs having a variety of fixed number of sub-states. Then, we experiment with three different sub-state selection methods: using two of the widely used information criteria for model selection, i.e, AIC and BIC together with a manually selected configuration based on human intuition about the activities to be recognized.

In the remainder of this section we present the details of our experimental setup, we describe the datasets used in the experiments and provide the details of our configuration selection methods.

A. Experimental Setup

We used two different kinds of publicly available datasets for the recognition of activities of daily living in a smart home and for recognizing office activities from audio data. We developed an experiment framework codebase using Matlab that allows a varying number of sub-states per activity.

Recognition performance is measured on a timeslice level, using the F-measure, which is the harmonic mean of precision and recall values. This metric considers the recognition of each activity as equally important and provides a reliable way for evaluating activity recognition methods. We repeat the experiments 10 times and present the average over those runs. This is done because the EM algorithm requires a random initialization of the parameters.

B. Audio Data

We use the event detection dataset (Office Live (OL) dataset) that was created for the IEEE challenge "Detection and classification of acoustic scenes and events" [16]. The dataset consists of every-day audio events in a number of office environments. The recorded activities include door knock, door slam, speech, laughter, keyboard clicks, objects hitting table, keys clinging, phone ringing, turning page, cough, printer, short alert-beeping, clearing throat, mouse click, drawer, and switches. The training data consists of around 20 training examples for each of the 16 different classes that can occur in an office environment. The official test data set for the challenge was never released to the public, our experiments are therefore performed on the development set, which consists of three recordings that are each roughly two minutes long (referred to as 'script01', 'script02' and 'script03'), recorded in a various office environments ('Office Live') and annotated by two people.

The audio data is discretized into frames using a window size of 80 ms with an overlap of 50% and a rectangular window. Features extracted include MFCCs and zero-crossing rate (ZCR), short-term energy (STE) and linear prediction coefficients. All features combined, we obtain a 35-dimensional feature vector for each frame. Additional details can be found in [5].

C. Smart Home Data

We use five real-world datasets collected each in a different smart home, three of the datasets are part of the 'van Kasteren' datasets, the other two are part of the ARAS datasets. All smart homes are equipped with binary sensors such as reed switches, pressure mats, mercury contacts, passive infrared (PIR) sensors, float sensors. The activities include leaving the house, toilet use, showering, brushing teeth, sleeping, having breakfast, dinner, snacking, and other. Further details about the data sets can be found in [10] and [17] respectively.

Data obtained from the sensors is transformed to the changepoint representation and discretized in timeslices of length $\Delta t = 60$ seconds. We split the data into a test and training set using a 'leave one day out' approach. In this approach, one full day of sensor readings are used for testing and the remaining days are used for training. We cycle over all the days in the dataset, so that each day is used once for testing.

D. Model Selection for Activity Complexity Determination

In order to find a suitable number of sub-states for each activity, we use AIC and BIC measures described in the previous section. We use all the occurrences of a given activity as a separate dataset. In order to obtain the optimum complexity level for the given activity, we start experimenting with the minimum possible model having a single cluster and try up to 10 clusters. By using cross validation, we obtain the data log likelihood on each sequence of a specific activity separately and apply the penalization criteria on each sequence separately. Finally, we average over all the folds and use the model with the minimum AIC or BIC value according to the metric used. Since the optimum number of sub activities are learnt independently for each activity, the transitions in the upper layer are not taken into account.

TABLE I. Manually Selected Configurations on Audio Data

	Number of Sub-States				
Activity	Set 1	Set 2	Set 3	Set 4	Set 5
Alert	1	5	1	1	1
Clear Throat	1	3	2	2	2
Cough	1	3	1	1	1
Door Slam	1	1	1	1	1
Drawer	2	2	2	2	2
Keyboard	1	5	2	2	2
Keys	1	4	2	2	1
Knock	1	1	1	1	1
Laughter	2	5	2	2	2
Mouse	1	2	1	1	1
Page Turn	1	3	1	1	1
Pen Drop	1	2	2	2	1
Phone	1	3	2	2	1
Printer	3	3	2	2	3
Speech	3	8	3	3	3
Switch	1	2	1	1	1
Other	5	3	3	5	5

As an alternative, we also use our intuition about the activity complexity levels and manually form a combination

for each dataset. For this reason, we chose a few number of candidate combinations for each dataset and experimented with them. We reported the combination with the maximum performance in the results section. We provide the experimented sets for the audio data in Table I. We obtained the best performance with the configuration in Set 5.

For the smart home datasets, each dataset belongs to a different house with different type and number of activities. Due to space considerations, we provide only the best performing configurations for the most common activities in Table II. The remaining activities include shaving, dressing, having a drink, playing the piano and they were annotated in only few of the houses. Those activities generally consisted of two sub-states in our intuitively selected combinations.

TABLE II. MANUALLY SELECTED CONFIGURATIONS ON SMART HOME DATA

	ARAS		Kasteren		
Activity	House A	House B	House A	House B	House C
Leave house	3	2	2	3	2
Breakfast	6	3	2	2	4
Lunch	6	7	-	-	2
Dinner	4	2	2	2	4
Eating (Snack)	4	-	2	2	5
Toileting	3	-	1	2	2
Showering	3	-	2	2	2
Brushing Teeth	2	-	2	2	3
Sleeping	4	1	2	3	2
Other	10	3	3	4	6

V. RESULTS

In this section, we present the detailed results of experiments on audio data and smart home data in the context of human behaviour modelling followed by a discussion on the results.

A. Audio Data

For the audio case, according to the results summarized in Table III, HHMM outperforms HMM for all scripts in all configurations. For Script 1, a single sub-state gives the best performance whereas for Script 2, using two sub-states gives a higher performance. For Script 3, using either number of sub-states yields in nearly equal performance. Increasing the number of sub-states to 5 does not result in an increase in performance. Yet, using a manually selected configuration of sub activities based on intuition. Script 3, which is the most challenging script in terms of performance, benefits most from the intuitively selected configuration.

TABLE III. RESULTS ON AUDIO DATA

		HHMM					
	HMM	All 1	All 2	All 5	Intuitive	BIC	AIC
Script 1	46.7	66.1	62.0	64.2	69.2	66.1	66.1
Script 2	39.3	54.4	56.8	55.2	60.7	54.4	54.4
Script 3	28.1	34.0	34.3	32.2	43.4	34.0	34.0
Average	38.0	51.5	51.0	50.5	57.8	51.5	51.5

B. Smart Home Data

For the smart home data, which are summarized in Table IV, HHMM consistently outperforms HMM for all five houses. For ARAS datasets, using 5 states for all activities gives the best performance and for van Kasteren datasets, using all two sub-states results in the maximum performance. In smart home datasets, the configurations selected using BIC and AIC resulted in configurations very similar to all two sub-states cases, thus yielding in nearly equal performances. With intuitively selected combinations, the performance on ARAS datasets is slightly improved, whereas for Kasteren datasets the improvement diminishes.

TABLE IV. RESULTS ON SMART HOME DATA

		HHMM					
	HMM	All 2	All 5	All 10	Intuitive	BIC	AIC
ARAS A	58.1	56.1	62.6	61.4	63.4	61.0	61.3
ARAS B	62.6	64.0	64.7	60.4	66.5	65.8	65.9
Kasteren A	64.7	69.9	70.2	67.4	69.3	65.8	70.7
Kasteren B	46.3	54.0	48.9	45.3	52.2	53.7	55.2
Kasteren C	42.8	50.3	48.6	47.8	50.5	50.0	49.5

C. Discussion

The results of our experiments demonstrate a significant increase in recognition performance in terms of F-measure when a hierarchical model is used. The performance gain is obtained with both continuous audio data and binary sensor data in the context of human behaviour modeling with the exact same model.

We also show that allowing different number of sub-states for different activities can result in significant increase in the performance. When we have a fixed number of sub-states, we assume that all activities have the same complexity level. While this assumption may hold for some data modality as in the case of smart homes, we cannot always make that assumption. With the audio data, we obtain a significant performance increase with different number of states. For example, the activities of daily living like having a shower or shaving can share the same level of complexity depending on the sensor types and deployment places. In that case, allowing different number of sub activities do not help. On the other hand, it is more likely that audio data for a human speech when compared to knocking on the door requires different level of complexity. The significant increase in the performance in the audio case supports this statement.

In terms of model complexity selection strategies, we obtained the best results with intuitively selected combinations. Selection using AIC and BIC measures resulted in less complex models. For the audio case, both BIC and AIC measures selected the least complex model that consists of a single state for every activity. Similarly, for the smart home cases, both BIC and AIC tend to select less complex models. For nearly all of the houses, they selected a configuration with two sub-states for each activity with some exceptions. For instance, for ARAS House A dataset, both AIC and BIC resulted in 5 sub-states for the 'Other' activity, which contains several activities that are not annotated separately. For Kasteren House B, BIC selection resulted in 3 sub-states for Leaving the house and sleeping activities, for House A, AIC selected 3 sub-states for leaving the house activity. Based on the experiment results, we conclude that AIC and BIC measures generally underestimates the complexity of the models for several activities both for continuous audio data and binary smart home data. This sub-optimal assignment leads to a degradation in recognition

performance. However, it is possible to find a better assignment methodology in order to fully make use of the power of hierarchical models.

VI. Conclusion

We have presented a hierarchical model for the recognition of human activities from sensor data. The proposed model uses a semi-supervised learning approach to automatically cluster the inherent structure of activities into actions. Our evaluation on both audio and smart-home data shows how two very different sensor modalities can rely on the same model for the recognition of human behavior. Such a unified approach to modeling can be very beneficial for the large scale deployment of a solution.

Our results on multiple real world datasets show that the use of a hierarchical model consistently outperforms its non-hierarchical counterpart in terms of recognition performance, given that an adequate number of states is used for modeling the actions in the hierarchy. In the case of the audio data, we have seen that the use a hierarchy with a variable number of action states can result in a significant increase in performance compared to a hierarchy that uses a fixed number of states. On the smart home data, the use of a variable number of action states did not result in a significant increase, demonstrating that the importance of using a variable number of states depends on the varying complexity of the activities that are being modeled. For those problems that require a variable number of action states, we identified the need for an automatic method for determining the optimal number of action states to use. Classic methods for model selection such as the Bayesian information criterion and the Akaike information criterion proved unsuccessful in determining the ideal number of action states. An intuitively chosen set of action states, based on the estimated complexity of the activities, was able to outperform the models that were selected by the Bayesian and Akaike information criterion.

Acknowledgment

This work is supported by the Boğaziçi University Research Fund under the grant number 11A01P7.

References

[1] T. van Kasteren, A. Noulas, G. Englebienne, and B. Kröse, "Accurate Activity Recognition In A Home Setting," in *Proceedings of the 10th international conference on Ubiquitous computing*, ser. UbiComp '08. New York, NY, USA: ACM, 2008, pp. 1–9.

[2] Y.-S. Lee and S.-B. Cho, "Activity Recognition Using Hierarchical Hidden Markov Models on a Smartphone with 3D Accelerometer," in *Proceedings of the 6th International Conference on Hybrid Artificial Intelligent Systems - Volume Part I*, ser. HAIS'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 460–467.

[3] N. Oliver, E. Horvitz, and A. Garg, "Layered Representations for Human Activity Recognition," in *In Fourth IEEE Int. Conf. on Multimodal Interfaces*, 2002, pp. 3–8.

[4] T. L. M. van Kasteren, G. Englebienne, and B. J. Kröse, "Hierarchical Activity Recognition Using Automatically Clustered Actions," in *Ambient Intelligence*. Springer, 2011, pp. 82–91.

[5] M. E. Niessen, T. L. M. Van Kasteren, and A. Merentitis, "Hierarchical Sound Event Detection," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2013.

[6] M. Ghazvininejad, H. R. Rabiee, N. Pourdamghani, and P. Khanipour, "HMM Based Semi-supervised Learning for Activity Recognition," in *Proceedings of the 2011 International Workshop on Situation Activity & Goal Awareness*, ser. SAGAware '11. New York, NY, USA: ACM, 2011, pp. 95–100.

[7] L. Piyathilaka and S. Kodagoda, "Gaussian Mixture Based HMM for Human Daily Activity Recognition Using 3D Skeleton Features," in *8th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, 2013, pp. 567–572.

[8] M. A. M. Shaikh, K. Hirose, and M. Ishizuka, *The Systemic Dimension of Globalization*. InTech, 2011, ch. Recognition of Real-World Activities from Environmental Sound Cues to Create Life-Log.

[9] G. Singla, D. J. Cook, and M. Schmitter-Edgecombe, "Recognizing Independent and Joint Activities Among Multiple Residents In Smart Environments," *Journal of Ambient Intelligence and Humanized Computing*, vol. 1, no. 1, pp. 57–63, 2010.

[10] T. van Kasteren, G. Englebienne, and B. Kröse, "Human Activity Recognition from Wireless Sensor Network Data: Benchmark and Software," in *Activity Recognition in Pervasive Intelligent Environments*. Springer, 2011, pp. 165–186.

[11] A. Mannini and A. M. Sabatini, "Machine Learning Methods for Classifying Human Physical Activity from On-Body Accelerometers," *Sensors*, vol. 10, no. 2, pp. 1154–1175, 2010.

[12] S. Fine, Y. Singer, and N. Tishby, "The Hierarchical Hidden Markov Model: Analysis and Applications," *Machine Learning*, vol. 32, no. 1, pp. 41–62, Jul. 1998.

[13] K. Murphy and M. A. Paskin, "Linear Time Inference In Hierarchical HMMs," in *Advances in Neural Information Processing Systems (NIPS)*, 2001.

[14] S. Karaman, J. Benois-Pineau, R. Mgret, J. Pinquier, Y. Gaestel, and J.-F. Dartigues, "Activities of Daily Living Indexing by Hierarchical HMM for Dementia Diagnostics," in *9th International Workshop on Content-Based Multimedia Indexing (CBMI)*, 2011, pp. 79–84.

[15] G. Celeux and J.-B. Durand, "Selecting Hidden Markov Model State Number With Cross-Validated Likelihood," *Computational Statistics*, vol. 23, no. 4, pp. 541–564, 2008.

[16] D. Giannoulis, E. Benetos, D. Stowell, and M. D. Plumbley, "IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events - Training Dataset for Event Detection Task, subtasks 1 - OL and 2 - OS," Queen Mary University of London, Tech. Rep., 2012.

[17] H. Alemdar, H. Ertan, O. D. Incel, and C. Ersoy, "ARAS Human Activity Datasets in Multiple Homes with Multiple Residents," in *7th International Conference on Pervasive Computing Technologies for Healthcare*, 2013.