Hierarchical Models for Activity Recognition

Amarnag Subramanya
Dept. of Electrical Engineering
University of Washington
Seattle, WA 98195

Alvin Raj
Dept. of Computer Science
University of Washington
Seattle, WA 98195

Jeff Bilmes
Dept. of Electrical Engineering
University of Washington
Seattle, WA 98195

Dieter Fox
Dept. of Computer Science
University of Washington
Seattle, WA 98195

***Abstract*— In this paper we propose a hierarchical dynamic Bayesian network to jointly recognize the activity and environment of a person. The hierarchical nature of the model allows us to implicitly learn data driven decompositions of complex activities into simpler sub-activities. We show by means of our experiments that the hierarchical nature of the model is able to better explain the observed data thus leading to better performance. We also show that joint estimation of both activity and environment of a person outperforms systems in which they are estimated alone. The proposed model yields about 10% absolute improvement in accuracy over existing systems.**

I. Introduction

In the recent past, advances in wearable sensing and computing devices have made possible the fine-grained estimation of a person's activities over extended periods of time [1]. The interest in human activity recognition stems from a number of applications that rely on accurate inference of activities that a person is performing. These include, context aware computing [2] to support for cognitively impaired people [3], long-term health and fitness, monitoring and automatic after action review of military missions.

Bao and Intille [4] used multiple accelerometers placed on a person's body to estimate activities such as standing, walking, or running. Kern *et al* [5], [6] and Lukowicz *et al* [7] added a microphone to a similar set of accelerometers in order to extract additional context information. One of the drawbacks of the system in [5], [7] is that they utilize multiple sensors and measurements taken all over the body. This can often lead to unwieldy systems with large battery packs. To overcome this, Lester *et al* [1] developed a small low-power sensor board that is mounted on a single location on the body.

Once a wearable sensor system is in place, the next logical step is to design algorithms to extract pertinent features from the sensor streams, and then classifiers that make use of these features to infer the activities being performed. [1] also showed how to apply boosting in order to learn activity classifiers based on the sensor data. However, a common drawback in all previously proposed approaches is that they feed the sensor data or features into static classifiers [4], [2], or a bank of temporally independent HMMs [1]. Further, most of the previously proposed algorithms [1], [4] do not make a distinction between 'complex' and 'simple' activities. In practice, it might be advantageous to decompose complex activities into simpler activities that might be easier to learn.

A number of 'complex' activities that we perform in our daily lives can be broken into smaller, simpler activities. For example, the process of driving a car involves, getting into the car, turning on the engine, driving, etc. Or getting onto an elevator, could comprise calling for the elevator, waiting for the elevator, etc. In this paper we refer to these simpler activities as *sub-activities*. Intuitively, it might be easier for the model to learn the simpler sub-activities rather than the complex ones. In practice though, it is not entirely clear how a given activity can be split into its constituent sub-activities, i.e., consider the car example above, we could say that when a person is in the process of turning on the engine, his motion state is *stationary*, on other hand since he is really sitting inside the car, his motion state could also be classified as *vehicle*. Thus, a statistically ideal approach would be to let the model learn the best constituent sub-activities for a given activity from the data during training. In this paper, we propose a hierarchical dynamic Bayesian network that implicitly learns these sub-activities during training.

Yet another novelty of our work here is the joint estimation of both the motion state and the environment. In many situations, the type of activity that we perform is constrained by our surroundings (environment). For example, if a person were inside a building, he is very unlikely to be driving a car. Similarly, it is more likely that a person is going up/down stairs when indoors rather than when he is outdoors. In this paper, we propose a model that in addition to estimating the motion state (activity) of a person, jointly estimates his environment, i.e., whether a person is indoors, outdoors or in a vehicle. We also show how jointly estimating both the state and environment outperforms systems that estimate them independently.

In addition to the above, this paper describes the models used in the first NIST evaluations for the DARPA ASSIST project. While the models proposed here can be applied to any activity recognition task, we use automatic after-action-review (AAR) of military missions to explain the models. An AAR is essentially a summary of a military mission and is created from memory by the mission leader. It reports on various activities/incidents that took place during the mission. As the duration of the mission increases, it becomes difficult for the leader to remember all the incidents to a significant degree of detail. The proposed system is supposed to aid the leader towards creating better summaries of the mission.

In our previous work on the same problem [8], [9], we have proposed algorithms to jointly infer the activity and location of a person. These systems make use of information from a GPS unit in addition to the sensors streams used in this paper.

[9] makes use of a Rao-Blackwellised particle filter, while [8] makes use of a dynamic Bayesian network in order to track the activity and location of a person. [8] also shows how virtual evidence can be used to train an activity recognition system in a semi-supervised manner (i.e. when labels are missing). However, the part of the system that infers the users activities in our previous work is a subset of the models proposed here.

In Section II, we give a brief overview of our sensor board. Section III describes the proposed hierarchical model and the feature extraction process. Experiments are described in Section V, followed by conclusions and future work in Section VII.

II. Wearable Sensor System

We make use of the sensor board developed by [1]. It consists of a multi-sensor board and a Holux GPS unit with SIRF-III chipset which are connected (using Bluetooth) to an iPAQ PDA for data storage.

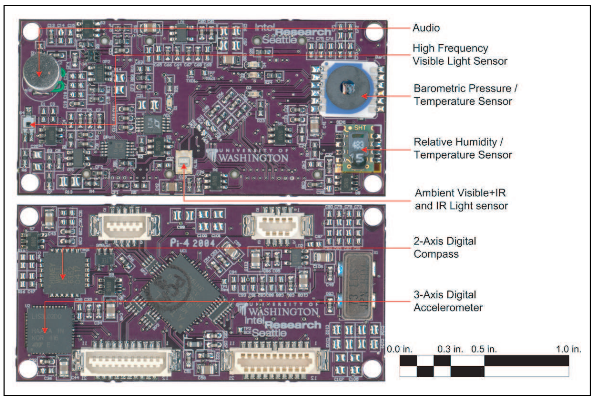
Audio
High Frequency Visible Light Sensor
Barometric Pressure / Temperature Sensor
Relative Humidity / Temperature Sensor
Ambient Visible+IR and IR Light sensor
2-Axis Digital Compass
3-Axis Digital Accelerometer
0.0 in.
0.3 in.
0.5 in.
1.0 in.

Fig. 1. Multi-sensor board (MSB).

Our customized multi-sensor board shown in Fig. 1 is extremely compact, low-cost, and uses standard electronic components. It weighs only 121g *including battery and processing hardware*. Sensors include a 3-axis accelerometer, several microphones for recording speech and ambient sound, phototransistors for measuring light conditions, and temperature and barometric pressure sensors. The overall system is able to operate for more than 8 hours with a single battery charge.

III. Model Description

Fig. 2 shows two consecutive time slices of the hierarchical model that is used to jointly infer the activity and environment of a person. Note that all observed variables are shaded, deterministic dependences are depicted using solid black lines, value specific dependences (see equation 1) are shown using a dot-dash lines and random dependencies are represented using dotted lines. In this model, A_t represents the current activity (motion state), E_t the environment, A_t^P models the 'sub-activity', E_t^P models 'sub-environment', A_t^T, E_t^T, A_t^{ST} and E_t^{ST} are random variables that turn on when there is an activity, environment, sub-activity and sub-environment transition respectively. Note that all variables in the model are discrete. Also note that in the following, given any random variable (rv) X, we use D_X to denote the domain of X. For example, D_{A_t} represents the set of values that the rv A_t can take.

The proposed model exhibits a synchronous hierarchy at two levels. The first at the activity level, where the activity observations are explained by the simpler sub-activity variables, which in turn depend on the activity variables. A similar hierarchy is also seen at the environment level. In the joint model, these two levels are synchronized at the activity and environment levels. The proposed model is in essence a multisteam asynchronous dynamic Bayesian network.

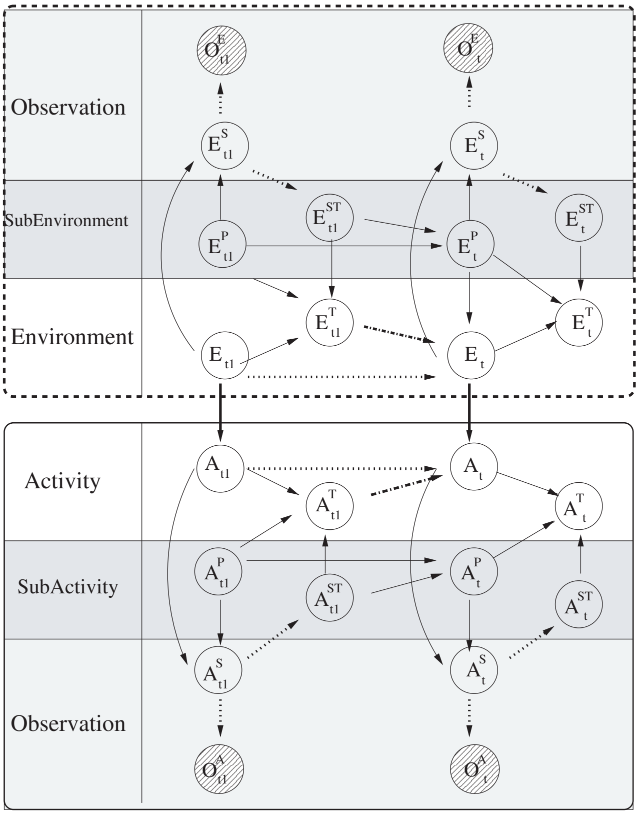

Fig. 2. Graphical model representation of the joint activity and environment recognizer

As described in section I, we assume that there are a finite set of states that a person can be in at any given instant of time. In our current system, these states include $A_t \in$ {`stationary`, `walking`, `running`, `driving vehicle`, `going up-stairs`, `going down-stairs`, `situation assessment from cover`, `incapacitated`}. The above activities were chosen by NIST/DARPA as relevant to a soldier for AARs. The interaction between A_t and the other variables in the model is defined by

$$P(A_t = i | A_{t-1} = j, E_t = l, A^T_{t-1})$$
$$= \begin{cases} P(A_t = i | A_{t-1} = j, E_t = l) & \text{if } \; A^T_{t-1} = 1, \\ 1 & \text{if } \; i = j, A^T_{t-1} = 0, \\ 0 & \text{if } \; i \neq j, A^T_{t-1} = 0. \end{cases} \tag{1}$$

where $P(A_t = i|A_{t-1} = j, E_t = l)$ is a dense conditional probability density that is learnt during training. In comparison to some of the previous work, here, we model the temporal dependencies of the both A_t and E_t. These allow the system to capture information such as “it is very unlikely to get into the driving state right after going upstairs”.

The conditional probability of the transition variable A_t^T given its parents is assumed to be

$$P(A_t^T = 1|A_t^P = j, A_t = l, A_t^{ST} = m) \\ = \begin{cases} 1 & \text{if } \ g_A(m) + j > f_A(l), \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

where, f_A is a mapping from the set of motion states to the number of sub-activities for each motion state and g_A is a mapping from A_t^{ST} to the increment it causes in A_t^P. The sub-activity variable A_t^P is modeled using

$$P(A_t^P = i|A_{t-1}^P = j, A_{t-1}^T, A_{t-1}^{ST} = m) \\ = \begin{cases} 1 & \text{if } \ i = 0, \ A_{t-1}^T = 1, \\ 1 & \text{if } \ i = g_A(m) + j, \ A_{t-1}^T = 0 \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

The conditional distributions of the sub-activity and sub-environment transition variables A_t^{ST} and E_t^{ST} are dense CPTs and are learnt from data. Once again, we do not specify a particular division of an activity into sub-activities, but let the model learn them implicitly. These ‘sub-transition’ variables, could for example, enforce the constraint that for someone to have performed the activity drive, they must have performed the sub-activities which might include, getting into a vehicle, turning on the engine, driving, and so on.

The variable E_t, captures the person’s spatial context, where we assume $E_t \in \{\texttt{indoors}, \texttt{outdoors}, \texttt{vehicle}\}$. Note that due to the edge between E_t and A_t, there can be (both soft and hard) constraints imposed between the motion state and the environment. For example, whenever it is the case that the environment is in the `indoors` or `outdoors` state, we *a priori* preclude `driving` from being a possible value of the motion type (i.e., it has zero probability). Whenever the environment is in the `vehicle` state, the motion type may not be `up/down stairs` (but it may be `stationary`, for example). Like the motion state variable, the environment variable is observed during training. The variables E_t^T and E_t^P are modeled in a manner similar to A_t^T and A_t^P explained above. The only difference though, was that we used a different set of mapping functions f_E and g_E in place of f_A and g_A.

In the remaining part of this section, we discuss the feature extraction algorithm. The sensor board produces a variety of signals at different rates. We employ a feature extraction process developed by our colleagues [1]. Briefly, the sensor sample rates are first normalized by low-pass filtering and/or up/down-sampling to an appropriate rate so that information is not lost. Next, each signal is windowed, and in each signal a feature vector is extracted, giving us, for each underlying time window, a feature vector of very high dimensionality. Such high-dimensionality feature vectors are not possible to utilize directly in a model, and typical approaches either require dimensionality reducing linear transforms (e.g., principle component analysis (PCA), or linear-discriminant analysis(LDA)) or alternatively feature selection. We utilize the approach taken in [1] to select pertinent features for classification. Essentially, for each activity we learn boosted one-level decision-tree classifiers. In other words, for each activity, we learn a collection of decision trees that each have a depth of unity, and where the next decision tree is obtained via boosting. Since the trees are of depth one, we can also view each tree as a simple threshold detector. Each decision tree essentially acts as a weak-learner, alone incapable of making an accurate detection decision, but when combined with kindred classifiers, capable of making highly accurate decisions. This collection of decision trees for each activity is then used to produce a final event detection probability $0 \leq p_i \leq 1$ for activity i. The detection probability p_i is obtained by viewing the decision tree threshold as a decision boundary, the distance to which constitutes a margin. Considering these margins together, we can obtain an average distance to decision boundary, which is then passed through a sigmoid function to produce a $[0, 1]$-valued probability. It is these probabilities that are then uniformly quantized into 10 bins to produce the integer observations — e.g., $O_t = 3$ if $0.2 \leq p_i < 0.3$.

Thus for each activity and environment we learn a tree (depth one) of boosted classifiers, which implies that the dimensionality of O_t^A is $|D_{A_t}|$, and that of O_t^E is $|D_{E_t}|$. In our current system, the dimensionality of O_t^A and O_t^E are 8 and 3 respectively. The observation model make use of a naive Bayes like implementation, i.e.,

$$p(O_t^A|A_t^S) = \prod_{i=1}^{|A_t|} p(O_t^A(i)|A_t^S) \tag{4}$$

where, $O_t^A(i)$ is the i^{th} dimension of the observation at time k. The distribution for $p(O_t^E|E_t^S)$ is defined in a similar fashion.

IV. Data Collection

In order to collect data, users were asked to perform a variety of activities on the University of Washington, Seattle campus. These activities included walking, running, going up/down a flight of stairs, driving around in a vehicle, etc. Users were instructed to perform the above activities in a natural manner and neither the sequence of activities nor their durations was choreographed. The users were instructed to label the different activities that they performed as they collected data. This meant that we had frame level labels for training our models. In all, there were 8 participants in the data collection effort, resulting in about 25 data traces. Each trace had an average duration of 30 minutes.

V. Experiments

In order to evaluate the proposed hierarchical model in different settings, we did the following: In the first experiment, we set $f_A(l) = f_E(l) = 1 \ \ \forall \ l$. Thus in this case, $|D_{A_t^P}| =$

$|D_{E_t^P}| = 1$. Also $|D_{A_t^{ST}}| = |D_{E_t^{ST}}| = 2$, which implies that at each time instant you are making a probabilistic choice between staying the same activity or making a transition (to either the same or different activity). In this model each activity has an geometric duration model and is referred to as the 'single state model' in the rest of the paper. With the above parameterization, we are making the assumption that each activity contains only a single sub-activity, which is itself. Another parameterization that we tried was to set

$$f_A(l) = \begin{cases} 4 & \text{if } \ l \in \{upstairs,\ downstairs\} \\ 2 & \text{if } \ l \in \{situation\ assessment\ from\ cover\} \\ 8 & \text{otherwise,} \end{cases} \tag{5}$$

and $f_E(l) = 8 \quad \forall\ l$. In this case it can seen that $|D_{A_t^P}| = |D_{E_t^P}| = 8$ which implies that each activity/environment can have at most 8 components (sub-activities/sub-environments). The above choice of f_A and f_E was motivated by a number of factors including the expected minimum duration of an activity and the amount of training data available. The above choice was verified to yield the best performance on a held-out set. Further, in this experiment, we set $|D_{A_t^{ST}}| = |D_{E_t^{ST}}| = 3$, where $A_t^{ST} = 0$ meant no sub-activity transition, $A_t^{ST} = 1$ would force the model to transition to the next sub-activity and $A_t^{ST} = 2$ gave the model the freedom to skip the next sub-activity. It important to clearly understand why a model needs to be given the freedom to skip sub-activities: consider, for example, the process of traveling in a car as a passenger, then if one of the sub-activities (for driving) is 'turn on engine', the model trying to explain the passenger data must be capable of skipping the 'turn on engine' sub-activity. In the following we refer to the above parameterization as the 'multi state model'. In all of the above, we make use of an identity mapping in the case of g, i.e., $g_A(m) = g_E(m) = m$.

In the proposed hierarchical model (figure 2), disconnecting the link between E_t and A_t yields two (sub)-graphical models, one that attempts to infer the context (environment) of the user and the other which attempts to infer the motion state of the user. In figure 2, the part of the model that infers motion state alone (Ω_A) is shown using a solid bounding box and the part of the model that infers context alone (Ω_E) is depicted using a dashed bounding box. On the other hand keeping the link intact, leads to a model in which motion state and environment are jointly inferred (Ω_{EA}).

All of the above models in different settings were implemented using the Graphical Models Toolkit (GMTK) [10]. In each case, we performed leave-one-out cross validation on our data set. We trained the binary adaboost classifiers and discretized the margins of the weak learners, as explained in section III. These discrete features were then used to jointly learn all the parameters of the graphical model. The models were then evaluated based on the Viterbi output on the test trace.

VI. Results

For each trace, accuracy was determined by counting the number of correctly labeled frames divided by the total number of frames. We separately determined accuracy in estimating the person's motion state and accuracy in estimating the environment. The mean and 95% confidence intervals of the motion state and environment accuracies achieved for different sets on the 25 test traces are summarized in tables I and II

Task	Adaboost	Single State	Multi State
State Only (Ω_A)	77.0 ±2.5	82.0 ±2.1	84.58 ±1.02
Environment Only (Ω_E)	82.1 ±3.7	88.7 ±3.7	90.90 ±1.78

TABLE I

Comparison of accuracies for different models when motions state and environment are estimated independently.

Accuracy	Single State	Multi State
Motion State	82.2 ±2.1	86.10 ±0.98
Environment	89.4 ±3.3	92.83 ±1.35

TABLE II

Results for model Ω_{EA}. Both environment and motion state were jointly inferred.

Table I shows the results for independent inference of activity and environment using various techniques. The column corresponding to Adaboost gives the results of using the boosted tree of classifiers to classify each frame, i.e., makes use of no temporal information [1]. The 'single state' model, in some sense may be considered as a first step towards incorporating temporal constraints (and is closest to the current state of the art). As it can be seen the single state model improves system performance by about 5% for both activity and environment when compared to the system that makes use of only Adaboost. This suggests that temporal information can help improve performance. The third column shows the results of the 'multi state model'. It can be seen that giving the model the freedom to choose sub-states (activities/environments) yields about 2.5% improvement in system performance over the single state model. In addition, it can be seen that the multi state model is able to achieve a smaller 95% confidence interval in comparison to other models.

Table II shows the results of the joint inference. Note that in the case of Adaboost, as the classifiers for each individual activity/environment are learnt independently, they cannot be jointly estimated. It can be seen here that the multi-state hierarchical model outperforms the single state model by about 4% for both tasks.

VII. Conclusions and Future Work

In this paper we have proposed a hierarchical model to jointly estimate both the context and motion state of a person. We have shown that modeling temporal dependencies can help improve system performance. Further, we have also shown that it is advantageous to break a complex activity into simpler/smaller sub-activities and then build models for these sub-activities. Finally we have shown that jointly estimating both the motion state and context of a person performs better than individual estimation.

In future we plan on using the proposed hierarchical models to jointly recognize a persons activities, environment and his location. We also intend to investigate other approaches to feature extraction such as Neural Networks (with appropriate regularization).

Acknowledgments

The authors would like to thank Tanzeem Choudhury, Jonathan Lester, and Gaetano Borriello for many useful discussions and for making their feature extraction code available. This work was supported by DARPA's ASSIST Program, Contract No-N000140510388.

References

[1] J. Lester, T. Choudhury, N. Kern, G. Borriello, and B. Hannaford, "A hybrid discriminative-generative approach for modeling human activities," in *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2005.

[2] J. Ho and S. Intille, "Using context-aware computing to reduce the perceived burden of interruptions from mobile devices," in *Proc. of the Conference on Human Factors in Computing Systems (CHI)*, 2005.

[3] D. Patterson, L. Liao, K. Gajos, M. Collier, N. Livic, K. Olson, S. Wang, D. Fox, and H. Kautz, "Opportunity Knocks: a system to provide cognitive assistance with transportation services," in *International Conference on Ubiquitous Computing(UbiComp)*, 2004.

[4] L. Bao and S. Intille, "Activity recognition from user-annotated acceleration data," in *Proc. of the Int. Conference on Pervasive Computing and Communications*, 2004.

[5] N. Kern, B. Schiele, and A. Schmidt, "Recognizing context for annotating a live life recording."

[6] N. Kern, B. Schiele, H. Junker, P. Lukowicz, and G. Trster, "Wearable sensing to annotate meeting recordings," in *The International Symposium on Wearable Computers*, 2002.

[7] P. Lukowicz, J. Ward, H. Junker, M. Stager, G. Troster, A. Atrash, and T. Starner.

[8] A. Subramanya, A. Raj, J. Bilmes, and D. Fox, "Recognizing activity and spatial context using wearable sensors," in *Uncertainity in Artificial Intelligence*, 2006.

[9] A. Raj, A. Subramanya, J. Bilmes, and D. Fox, "Rao-blackwellized particle filters for recognizing activities and spatial context from wearable sensors," in *Experimental Robotics: The 10th International Symposium, Springer Tracts in Advanced Robotics (STAR), Springer-Verlag*, 2006.

[10] J. Bilmes and G. Zweig, "The graphical models toolkit: An open source software system for speech and time-series processing," in *Proc of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2002.