Fine-Grained Activity Recognition by Aggregating Abstract Object Usage

Donald J. Patterson, Dieter Fox, Henry Kautz
University of Washington
Department of Computer Science and Engineering
Seattle, Washington, USA
{djp3,fox,kautz}@cs.washington.edu

Matthai Philipose
Intel Research Seattle
Seattle, Washington, USA
matthai.philipose@intel.com

Abstract

In this paper we present results related to achieving fine-grained activity recognition for context-aware computing applications. We examine the advantages and challenges of reasoning with globally unique object instances detected by an RFID glove. We present a sequence of increasingly powerful probabilistic graphical models for activity recognition. We show the advantages of adding additional complexity and conclude with a model that can reason tractably about aggregated object instances *and gracefully generalizes from object instances to their classes by using* abstraction smoothing. *We apply these models to data collected from a morning household routine.*

1. Introduction

The convergence of advances in wearable sensing technology and representations and algorithms for probabilistic reasoning is pushing the boundaries of the types of context with which a computer may reason. In the realm of wearable computers context is primarily used to refer to the state of the user: his or her location, current activity, and social environment.

In this paper we provide evidence that reasoning about objects in an environment, both in general and specific terms, is fruitful for inferring aspects of context related to activity performance. We obtain information about object use by utilizing a glove outfitted with a Radio Frequency Identification (RFID) antenna. Notably, RFID antennae are able to discriminate among specific instances of objects that are otherwise the same (e.g., two spoons), with a 0% false positive rate [22].

Previous work has approached activity recognition from several directions. Early work on plan recognition [11, 23] had the insight that much behavior follows stereotypical patterns, but lacked well-founded and efficient algorithms for learning and inference, and did not ground their theories in directly sensed experience. More recent work on plan-based behavior recognition [4] uses more robust probabilistic inference algorithms, but still does not directly connect to sensor data.

In the computer vision community there is much work on behavior recognition using probabilistic models, but it usually focuses on recognizing simple low-level behaviors in controlled environments [9]. Recognizing complex, high-level activities using machine vision has only been achieved by carefully engineering domain-specific solutions, such as for hand-washing [13] or operating a home medical appliance [17].

There has been much recent work in the wearable computing community, which, like this paper, quantifies the value of various sensing modalities and inference techniques for interpreting context. This includes unimodal evaluations of activity and social context from audio [3, 19, 18], video [5], and accelerometers [10], multi-modal sensor evaluations [12] and work which optimizes sensor selection for arbitrary activity recognition [2]. A theme of much of this work is that “heavyweight” sensors such as machine vision can be replaced by large numbers of tiny, robust, easily worn sensors [7, 21].

Object-based activity recognition relies on the “invisible human hypothesis” of [15] which states that activities are well characterized by the objects that are manipulated during their performance, and therefore can be recognized from sensor data about object touches. Although little work has explored this hypothesis, Philipose *et al.* [15] looked at the problem of detecting 65 activities of daily living (ADLs), from 14 classes such as cooking, cleaning, and doing laundry, and used data from a wearable RFID tag reader deployed in a home with tagged objects. This work was able to categorize activities that were disambiguated by key objects (for example, using a toothbrush versus using a vacuum cleaner), used hand-made models, and did not model how activities interact and flow between each other.

In this paper we begin to explore extending object-interaction based activity recognition in a more realistic set-

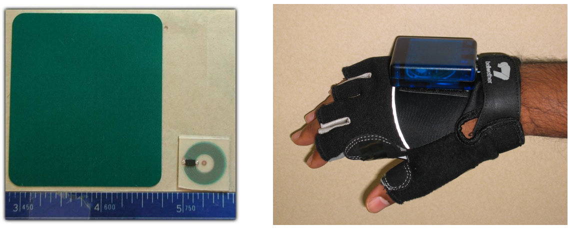

Figure 1. RFID tag types and the RFID glove used in this experiment.

ting. We address fine-grained activity recognition — for example, not just recognizing *that* a person is cooking, but determining *what* they are cooking — in the presence of interleaved and interrupted activities collected during the performance of a morning household routine. The following are our novel contributions:

- We demonstrate accurate activity recognition in the presence of interleaved and interrupted activities.
- We accomplish this using automatically learned models.
- We demonstrate the ability to disambiguate activities when models share common objects.
- We show that object-based activity recognition can be improved by distinguishing object instances from object classes, and learning dependencies on aggregate features.
- We implement abstraction smoothing, a form of relational learning, that can provide robustness in the face of expected variation in activity performance.

2. Problem Domain

Our experiments focused on routine morning activities which used common objects and are normally interleaved. Table 1 lists the 11 activities which were observed. To create our data set, one of the authors performed each activity 12 times in two contexts: by itself twice, and then on 10 mornings all of the activities were performed together in a variety of patterns.

In order to capture the identity of the objects being manipulated, the kitchen was outfitted with 60 RFID tags placed on every object touched by the user during a practice trial. The list of tagged objects is shown in table 2.

RFID tags have a form factor comparable to a postage stamp. The two tag technologies that we used are shown in figure 1. Unlike barcodes they do not require line-of-sight in order to be read and they identify specific globally unique instances of objects, rather than merely classes of objects.

1	Using the bathroom
2	Making oatmeal
3	Making soft-boiled eggs
4	Preparing orange juice
5	Making coffee
6	Making tea
7	Making or answering a phone call
8	Taking out the trash
9	Setting the table
10	Eating breakfast
11	Clearing the table

Table 1. Activities performed during data collection

For example, a bar code might identify a can of orange juice of a specific brand and size, but an RFID tag will identify which of the millions of those cans are being used.

The user simultaneously wore two gloves (see figure 1), built by Intel Research Seattle [8], outfitted with antennae that were able to detect when an RFID tag was within 2 inches of the palm. The time and id of every object touched was sent wirelessly to a database for analysis. These gloves function as a prototype for a more practical bracelet antenna which is under development.

The mean length of the 10 interleaved runs was 27.1 minutes ($\sigma = 1.7$) and object touches were captured at 10 per second. The mean length of each uninterrupted portion of the interleaved tasks was 74 seconds. Most tasks were interleaved with or interrupted by others during the 10 full data collection sessions.

This data introduces three confounding factors over previous work by [15]:

- The activities that the user performed shared objects in common. This made interleaved activity recognition much more difficult than associating a characteristic object with an activity (such as a vacuum).
- The activities were not performed sequentially or in isolation. During a pause in any given activity, progress was attempted in other parallel activities (such as when waiting for water to boil) and some activities interrupted others at uncontrolled times (such as answering the phone).
- Data was collected in the presence of four people, only one of whom was instrumented, which added uncontrolled variability to the data set.

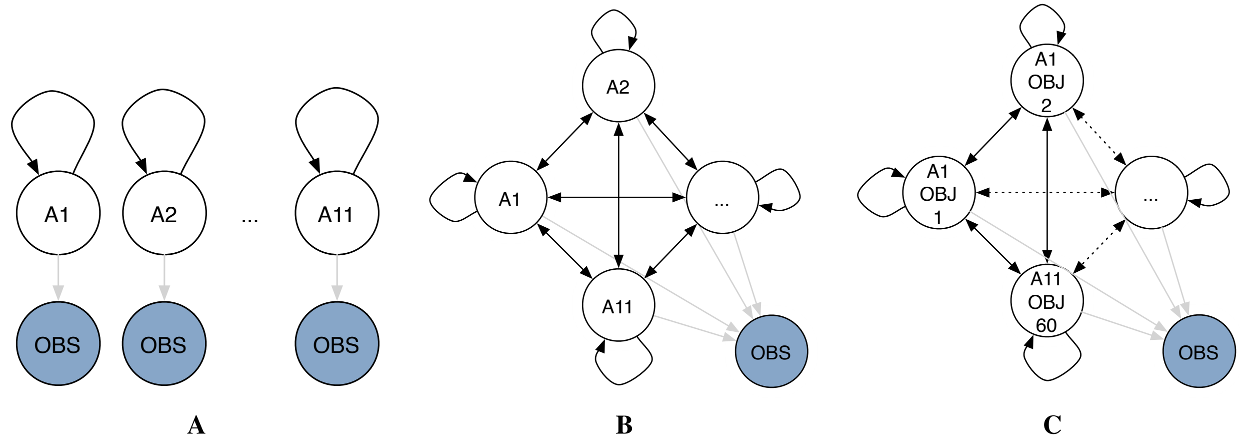

Figure 2. State diagrams for models A,B and C. A consists of 11 independent one-state HMMs. The log likelihood of each HMM was calculated on overlapping windows of data. B is a single 11-state HMM. C is a single 660-state (60 tags x 11 activities) HMM. In both B and C the most likely state sequence was recovered using the Viterbi algorithm over the entire data sequence.

bowl, coffee container, coffee grinder, coffee tamper, cupboard(6), dishwasher, door(2), drawer(2), egg carton, espresso cup(2), espresso handle, espresso steam knob, espresso switches, faucet(2), freezer, milk, hand soap, juice, juice pitcher, kettle, measuring cup-half, measuring cup-one, measuring scoop, milk steaming pitcher, mug, oatmeal, refrigerator, salt, saucepan, cooking spoon, stove control(2), sugar, table cup(4), table plate(4), table spoon(4), tea bag, tea box, telephone, toilet flush handle, toilet lid, vanilla syrup

Table 2. The 60 objects tagged for this experiment. Parentheses denote multiple instances.

3. Models Which Improve Accuracy

In order to maximize activity recognition we focused on *accuracy*, presented in this section, and then on *robustness*, presented in section 5. To improve accuracy we developed a very simple probabilistic model, evaluated its performance and then successively augmented it with features that disambiguated errors by adding representational power.

We chose to perform activity recognition using the probabilistic generative framework of Hidden Markov Models (HMMs) and their factored analog, the Dynamic Bayes Net (DBN). We chose these techniques because they are robust to sensor noise, they have well-understood algorithms for inference and parameter learning [16] and they have well-characterized temporal properties [14].

3.1. Baseline Model A: Independent Hidden Markov Models

As a baseline we modeled the activities as 11 independent one-state HMMs (see figure 2-A). Used in a generative context, each state emits an *object-X-touched* event or a *no-object-touched* event at each clock tick. Each state's emission probability was trained on the 12 recorded examples of the corresponding activity. After training, the probability of emitting a *no-object-touched* event was equalized across all HMMs so that the timing characteristic of the model was completely captured by the self-transition probability. The HMMs were trained and tested on data in which object types were equalized so that there was no distinction made between spoon #1 and spoon #2, for example, but both appeared identically as a "spoon".

To infer the activity being performed at each second, each HMM was presented with a 74 second window of data ending at the query second (the mean amount of uninterrupted time spent performing an activity in the data.) This produced a log-likelihood score for each model at each clock tick. The HMM with the highest score was the system's estimate of the current activity.

3.2. Baseline Model B: Connected HMMs

As a second baseline we connected the states from the 11 independent HMMs of model A in order to learn activity transitions. We retrained this HMM using the 10 examples of the user performing the 11 interleaved activities. The *no-object-touched* emission probability was again equalized across all states (see figure 2-B).

This HMM was evaluated over the entire data window and the Viterbi algorithm [16] was used to recover the activity at every time point given by the maximum likelihood path through the state-space. Again, this model was trained and tested on data in which object types were equalized to eliminate distinctions between instantiations of objects.

3.3. Baseline Model C: Object-Centered HMMs

As a third baseline we split the states in baseline B into a separate state for each activity and observable object. This allowed this model to capture some information about how objects were used at different temporal points in the execution of an activity at the expense of introducing more trainable parameters. We retrained this HMM using the 10 examples of the user performing the 11 interleaved activities. The *no-object-touched* emission probability was equalized across all states. The conditional probability table associated with state observations was degenerate for this model since each state could only emit an observation of one particular object or *no-object-touched* (see figure 2-C). This HMM was also evaluated over the entire data window using the Viterbi algorithm and was trained and tested on data with object types equalized.

3.4. Aggregate Model D: Dynamic Bayes Net with Aggregates

For our fourth model we examined the effect of reasoning about aggregate information by adding a feature which captured how many objects of a given type were touched during the current activity. This aggregate can only be computed if globally unique object instances can be identified. This choice was motivated by the desire to differentiate setting the table from eating breakfast. Both of these activities require touching objects in common, for example a spoon, but they differ by whether one spoon is touched four times or four spoons are touched once.

Figure 3 shows a DBN with this feature. Unlike figures 2-A,B,C, which are state diagrams, this figure is a dependency diagram in which the state of the system is factored into independent random variables indicated by nodes in the graph. It has been rolled out in time showing nodes from three different time steps and dependency diagrams for models A-C are also shown for comparison. The top row of model D has the same dependency structure as models A-C. Model D adds an additional deterministic state variable, the boolean exit node “E”, which captures dynamic information about when an activity has changed. It is true if the state has changed from one time step to another.

The gray box denotes an aggregation template which groups each class of objects with multiple instantiations. Each “Obj” node in the template is a deterministic boolean node indicating whether a given instantiation of an object has been touched since the last time the activity changed. The “Obj” nodes are aggregated by a summation node, “+”. When the DBN changes activities (“Exits”), the system explicitly reasons about the number of different instantiations of objects that were touched. This is captured by the dependence of the aggregate distribution node, “AD”, on the state node,“S”. The “AD” node is constrained to be equal to the summation node from the previous time step through the observed equality node, “=”. The equality node is always observed to be true and coupled with its conditional probability table force “AD” to equal “+”. The overall effect is that the probability of the model inferring whether or not an activity has ended is mediated by an expectation over the number of objects in each class which have been touched.

This model, including the aggregate distribution for activities, was also automatically learned from the 10 labeled training traces, but in contrast to models A, B and C, this model differentiates between instantiations of objects so that, for example, spoon #1 and spoon #2 created different observations.

3.5. Model Summary

The various features of these models are summarized in table 3. Models with “Exponential Timing” distributions implicitly expect the length of uninterrupted portions of an activity to be distributed exponentially because of their structure. The parameters of the distribution are learned from the data. “Inter-Activity Transitions” refer to the ability of the model to represent the tendency of certain activities to follow or interrupt other activities. “Intra-Activity Transitions” refer to the ability of the model to represent biases about when in the course of an activity certain objects are used. (e.g., one uses a kettle early in the process of making tea). Finally “Aggregate Information” refers to the ability of the model to represent aggregations over unique objects.

4. Accuracy Experiments

We calculate two accuracy metrics using leave-one-out cross validation across the 10 interleaved runs. The first was what percentage of the time the model correctly inferred the true activity. This metric is biased against slight inaccuracies in the start and end times and will vary based on the time granularity with which the experiments were conducted. We also evaluated our models using a string edit distance measure. We treated the output of the inference as a string over an 11 character alphabet, one character per activity, with all repeating characters merged. We calculated

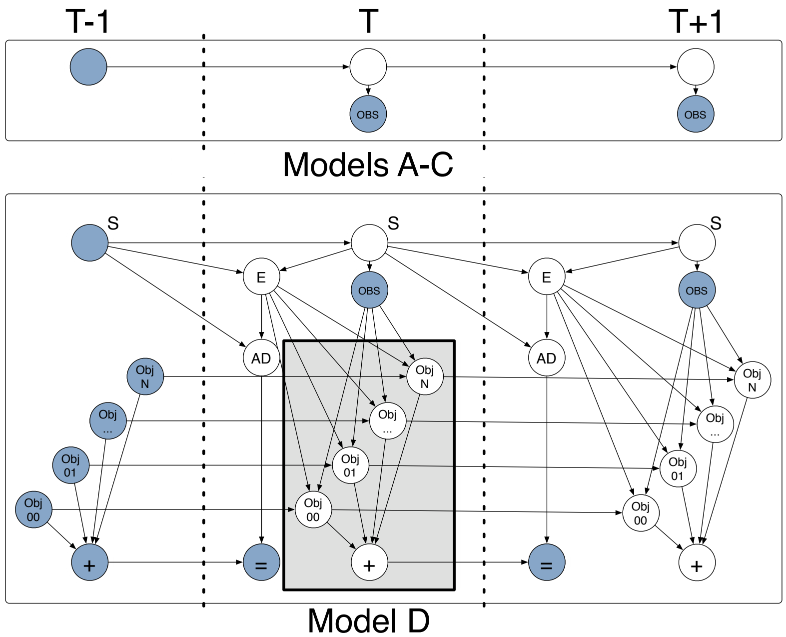

Figure 3. A dependency diagram showing the DBN used for inference in comparison to the baseline models, rolled out for three time steps. Time steps are separated by a vertical dotted line. Observed variables are shaded, hidden variables are not. All variables are discrete multinomials.

		Model			
		A	B	C	D
		Representational Power			
Exponential Timing Distribution			√	√	√
Inter-Activity Transitions			√	√	√
Intra-Activity Transitions				√	
Aggregate Information					√
		Accuracy			
Time-Slice	μ	68%	88%	87%	88%
	(σ)	(5.9)	(4.2)	(9.3)	(3.1)
Edit Distance	μ	12	9	14	7
	(σ)	(2.9)	(6.2)	(10.4)	(2.2)

Table 3. Summary of model representational power and performance.

the minimum string edit distance between the inference and the ground truth. A string edit distance of 1, means that the inferred activity sequence either added a segment of an activity that didn't occur (insertion), it missed a segment that did occur (deletion), or it inserted an activity that didn't occur into the middle of an activity (reverse splicing). A perfect inference will have a string edit distance of 0. The string edit distance is biased against rapid changes in the activity estimate and is tolerant of inaccuracies in the start and end time of activities. Table 3 summarizes the results of the experiments.

4.1. Inference Evaluation

Figure 4-A shows a short portion of the inference in which model A performed badly and demonstrates some of the key reasons why inference with this model only obtained 68% accuracy. In this figure ground truth is indicated with a thin line and the inference is indicated by a dot at each time slice. The independence assumptions of this model allow a rapid switching between activities which clearly need to be smoothed.

Figure 4-B shows a short portion of the inference performed by model B. In contrast to model A, the benefits of learning inter-activity transitions are seen in a smoother trace, and higher accuracies. There are two places in the

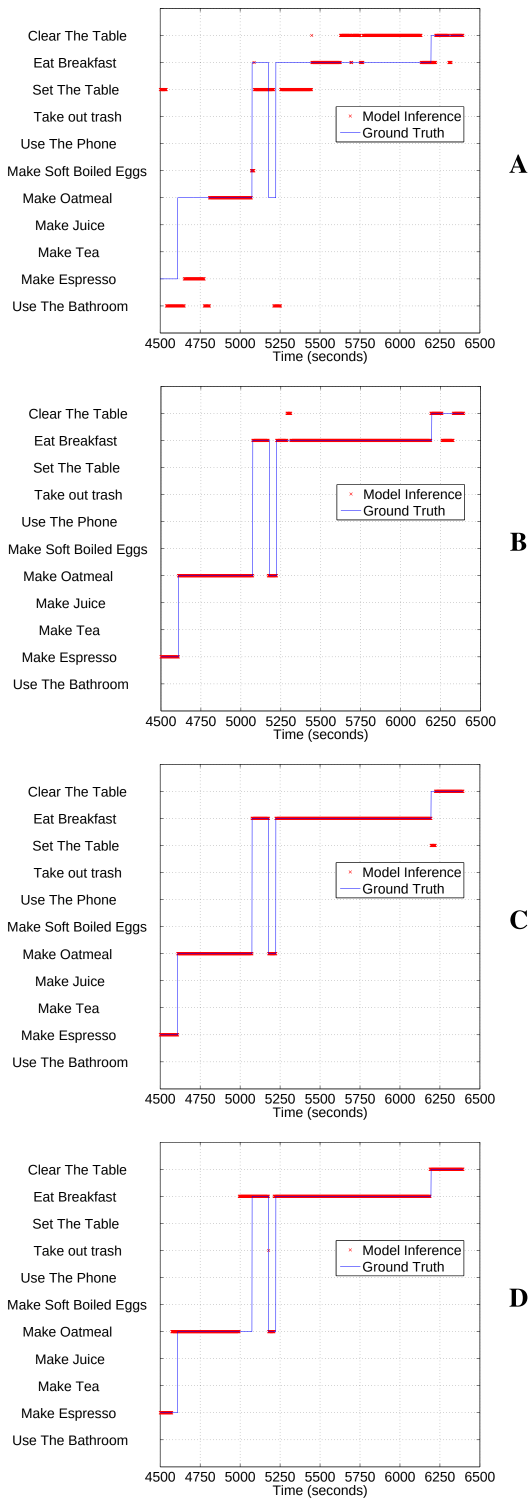

Figure 4. Small segment of inference with various models. Ground truth is indicated by the thin line. Inference is indicated by the dots.

figure where the activity “Eat Breakfast” is confused with the activity “Clear the Table”. These two activities weren’t differentiable by the objects that were used in their performance: both activities used plates, spoons and cups. Two possible techniques emerged for disambiguating these activities. The first was to allow the model to learn a tendency for objects to be used earlier or later in the activity – motivating model C. The second was to allow the model to differentiate activities based on the number of objects which were touched – motivating model D.

Figure 4-C shows a short portion of the inference performed by model C. The graph demonstrates how this model is able to correct for the errors of model B, but it clearly makes an inter-activity error by inferring a transition from eating breakfast to setting the table then to clearing the table. This caused the string edit accuracy of this model to get worse, although temporal accuracy remained constant compared to model B. Given a sufficient amount of training data, this model will not make such inferences. However a huge number of transition parameters, 660^2, are required to specify this model and our data simply did not have enough data to capture statistically significant information for every transition. Unlike model B, which captures information about transitions for activity A_i to A_j, model C represents information about transitions from $A_{i_{Obj_j}}$ to $A_{k_{Obj_m}}$.

Figure 4-D shows a short portion of the inference performed by model D. In this graph, learning a distribution over the aggregate number of object instantiations eliminates the ambiguity in figures 4-A,B,C without requiring an inordinate number of parameters to be introduced into the model. The variance in temporal accuracy is reduced and both accuracy and variance in the string edit distance is improved.

5. Models Which Improve Robustness

One of the weaknesses of the previous models is their response to the observation of an unexpected, but functionally similar object-use in an activity. For example, our data demonstrated making oatmeal with a cooking spoon. Our inference should not fail if the same task is performed with a table spoon. It should be a less likely, but still plausible alternative. To solve this problem we implemented a version of the general concept of *abstraction smoothing* introduced in [20].

5.1. Abstraction Smoothing Over Objects

Inspired by [1] we used a relational model of object similarity, but unlike the full power of that work, we used a single hierarchical object relation rather than a lattice. The hierarchy that we used was mined with supervision from an internet shopping site [6] (see figure 5). The name of each

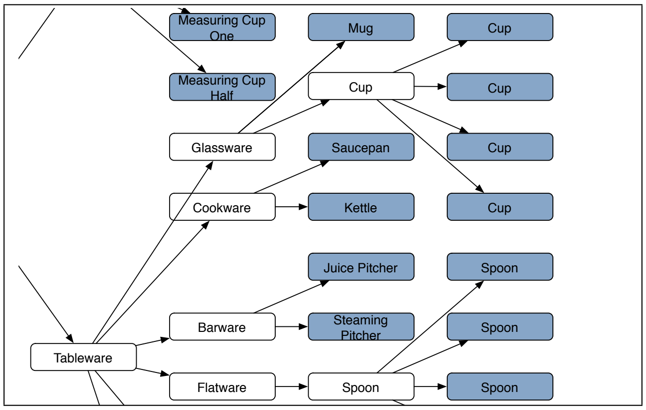

Figure 5. Object abstraction hierarchy with data set objects shaded.

object was entered into the shopping search engine and the hierarchy that was returned for that object was inserted into a global object tree. In the case of objects with multiple hierarchies, one was manually selected.

The semantics that we applied to the tree were that objects that were close to each other in the graph were functionally similar. To specify the notion of "close", we weighted all edges on the graph equally and created an all-pairs functional equivalence metric according the following formula:

$$P(O_i \rightarrow O_j) = \frac{\exp(-\frac{Dist(O_i,O_j)}{2})}{\sum_j \exp(-\frac{Dist(O_i,O_j)}{2})} \quad (1)$$

Where $Dist(O_i, O_j)$ is the shortest-path distance between O_i and O_j on the graph. When object O_i is expected in the model, it will be substituted by object O_j with probability $P(O_i \rightarrow O_j)$. The likelihood of substituting one object for another falls off exponentially with distance in the hierarchy.

Figure 6 shows how the abstraction function is inserted graphically into model D. This changes the semantics of the model. Now, node O_i represents the object that "should" be touched when what is observed is O_j. The conditional probability table associated with the dependency is captured by equation 1.

5.2. Robustness Experiments

To validate how well this technique worked when objects were substituted, we reran our experiments with abstraction smoothing added to model D. This resulted in an insignificant decrease in accuracy of 0.1% and -.1 in edit distance. Then we reran our experiments with the same data streams in which all instances of a mug were replaced with a cup.

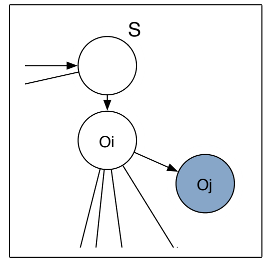

Figure 6. The O_i and O_j nodes replace the previous "OBS" node in model D.

As table 4 demonstrates, abstraction smoothing greatly increases robustness to object substitution. In the two low-parameter baseline models accuracy is dramatically lowered, but in comparison the abstraction model suffers a relatively modest decrease in accuracy.

6. Discussion and Conclusions

In this paper we have made a methodological point of applying increasingly sophisticated models to the RFID-glove activity recognition problem. Such an approach makes it clear what features are relevant for improved performance without sacrificing efficient inference and learning.

Training separate HMMs, as in model A, although popular in the literature (see [2]), performs poorly once activities share objects or interact in time. Coupling the HMMs into one system, as in model B, greatly improves accuracy by learning transitions between activities, but can be confused by activities which use the same class of object. Many activities have this property, for example, doing laundry versus getting dressed, reading a book versus organizing a bookshelf, *etc.*

An RFID-glove can distinguish members of a class of objects at the risk of losing the ability to generalize in those cases where object identity is irrelevant. The heart of the matter is that many activities are naturally non-Markovian, *i.e.*, they are best modeled in terms of the *history* of objects that were used during their execution. Model D main-

	Model		
	A	B	D
Mean Accuracy	52.5%	77.4%	81.2%
Net Change	-15.1%	-10.9%	-6.4%
Mean Edit Dist.	-12.7	-26.6	-1.1

Table 4. Accuracy metrics given object substitution.

tains such a history and allowed us to reason about the aggregate number of objects used.

Comparing model C, which allows for objects to be used at different temporal points in an activity, versus model D demonstrated that both techniques had disambiguation power. However, Model D had fewer free parameters and therefore requires less training data to achieve equal levels of performance.

Finally, we saw that the use of smoothing over an abstraction hierarchy of object types greatly enhances the robustness of the system with no significant change in accuracy, and no significant additional computational burden.

These results argue that key features for modeling activities of everyday life on the basis of object interaction include a way to capture transition probabilities between activities; a way to compute aggregate features over the history of instantiated events in the activity; and a way to handle abstract classes of events. With these features alone, it is possible to perform quite fine-grained activity recognition. The activity recognition is successful even when the activities are interleaved and interrupted, when the models are automatically learned, when the activities have many objects in common and when the user deviates from the expected way of performing an activity.

References

[1] C. Anderson, P. Domingos, and D. Weld. Relational Markov Models and their Application to Adaptive Web Navigation. In *Proc. of the Eighth Intl. Conf. on Knowledge Discovery and Data Mining*, pages 143–152. ACM Press, 2002.

[2] T. Choudhury, J. Lester, and G. Borriello. A Hybrid Discriminative/Generative Approach for Modeling Human Activities. In *Proc. of the Nineteenth Intl. Joint Conf. on A.I.*, Edinburgh, UK, 2005. Morgan-Kaufmann Publishers.

[3] T. Choudhury and A. Pentland. Sensing and Modeling Human Networks Using the Sociometer. In *Proc. of the Seventh IEEE Intl. Symposium on Wearable Computers*, pages 216–222. IEEE, October 2003.

[4] D. Colbry, B. Peintner, and M. Pollack. Execution Monitoring with Quantitative Temporal Bayesian Networks. In *6th Intl. Conf. on AI Planning and Scheduling*, April 2002.

[5] P. Fitzpatrick and C. Kemp. Shoes as a Platform for Vision. In *Proc. of the Seventh IEEE Intl. Symposium on Wearable Computers*, pages 231–234. IEEE, October 2003.

[6] Froogle Shopping Site. http://froogle.google.com/.

[7] J. L. Hill and D. E. Culler. Mica: A Wireless Platform for Deeply Embedded Networks. *IEEE Micro*, 22(6):12–24, 2002.

[8] Intel Research Seattle. Intel Research Seattle Guide Project Website, 2003. http://seattleweb.intel-research.net/projects/guide/.

[9] T. Jebara and A. Pentland. Action Reaction Learning: Automatic Visual Analysis and Synthesis of Interactive Behaviour. In *ICVS '99: Proc. of the First Intl. Conf. on Computer Vision Systems*, pages 273–292, London, UK, 1999. Springer-Verlag.

[10] K. K. Van Laerhoven and H. Gellersen. Spine Versus Porcupine: A Study in Distributed Wearable Activity Recognition. In *Proc. of the Eighth IEEE Intl. Symposium on Wearable Computers*, volume 1, pages 142–149. IEEE, October 2004.

[11] H. A. Kautz and J. F. Allen. Generalized Plan Recognition. In *AAAI*, pages 32–37, 1986.

[12] N. Kern, S. Antifakos, B. Schiele, and A. Schwaninger. A Model for Human Interruptability: Experimental Evaluation and Automatic Estimation from Wearable Sensors. In *Proc. of the Eighth IEEE Intl. Symposium on Wearable Computers*, volume 1, pages 158–165. IEEE, October 2004.

[13] A. Mihailidis, B. Carmichael, and J. Boger. The Use of Computer Vision in an Intelligent Environment to Support Aging-in-Place, Safety, and Independence in the Home. *IEEE Transactions on Information Technology in BioMedicine*, 8(3):238–247, 2004.

[14] T. Osogami and M. Harchol-Balter. A Closed-Form Solution for Mapping General Distributions to Minimal PH Distributions. In P. Kemper and W. H. Sanders, editors, *Computer Performance Evaluation / TOOLS*, volume 2794 of *Lecture Notes in Computer Science*, pages 200–217. Springer, 2003.

[15] M. Philipose, K. P. Fishkin, M. Perkowitz, D. J. Patterson, D. Hahnel, D. Fox, and H. Kautz. Inferring Activities from Interactions with Objects. *IEEE Pervasive Computing: Mobile and Ubiquitous Systems*, 3(4):50–57, 2004.

[16] L. R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In A. Waibel and K.-F. Lee, editors, *Readings in Speech Recognition*, pages 267–296. Kaufmann, San Mateo, CA, 1990.

[17] Y. Shi, Y. Huang, D. Minnen, A. Bobick, and I. Essa. Propagation Networks for Recognition of Partially Ordered Sequential Action. In *Proceedings of IEEE CVPR04*, 2004.

[18] M. Stager, P. Lukowicz, N. Perera, and G. Troster. Implementation and Evaluation of a Low-Power Sound-Based User Activity Recognition System. In *Proc. of the Eighth IEEE Intl. Symposium on Wearable Computers*, volume 1, pages 138–141. IEEE, October 2004.

[19] M. Stager, P. Lukowicz, N. Perera, T. von Buren, G. Troster, and T. Starner. SoundButton: Design of a Low Power Wearable Audio Classification System. In *Proc. of the Seventh IEEE Intl. Symposium on Wearable Computers*, pages 12–17. IEEE, October 2003.

[20] E. M. Tapia, T. Choudhury, M. Philipose, and D. Wyatt. Using Automatically Mined Object Relationships and Common-Sense for Unsupervised Activity Recognition. Technical Report IRS-TR-05-014, Intel Research Seattle, May 2005.

[21] E. M. Tapia, S. S. Intille, and K. Larson. Activity Recognition in the Home Using Simple and Ubiquitous Sensors. In *Pervasive*, pages 158–175, 2004.

[22] R. Want. A Key to Automating Everything. *Scientific American*, 290(1):56–65, 2003.

[23] R. Wilensky. *Planning and Understanding: A Computational Approach to Human Reasoning*. Addison-Wesley, Reading, MA, 1983.