Using Context with Statistical Relational Models – Object Recognition from Observing User Activity in Home Environment

Chen Wu and Hamid Aghajan
Department of Electrical Engineering
Stanford University, Stanford CA, USA
chenwu,aghajan@stanford.edu

ABSTRACT

Object recognition from images in a home environment is challenging since the object usually has low resolution in the image and the scene is usually cluttered. However, many objects have specific functions to the user and the interactions between the user and the object provides useful contextual information to recognize the object. In this paper, we use Markov logic network (MLN) to model such context information as relationship between the objects and user activities. We demonstrate that Markov logic network provides a flexible way in the syntax of first-order logic to incorporate relational context information. It is also a probabilistic graphical model which handles uncertainty in the knowledge base, observations and decisions. In our experiment, objects in the living room and kitchen in a home are recognized based on only user's activity. The user's activity is analyzed from images of cameras installed in the home. Relationship between user activity and objects is defined in a knowledge base with MLN. Experiments show that objects in the home can be recognized irrespective of their position, size and appearance in the image.

Categories and Subject Descriptors

I.4.8 [**Scene Analysis**]: Object recognition; I.2.4 [**Knowledge Representation Formalisms and Methods**]: Relation systems

General Terms

Algorithms

Keywords

Object recognition, statistical relational models, smart homes

1. INTRODUCTION

Most approaches for object recognition from images are based on image features such as local appearance models, grouping geometric primitives [3, 7] and image contexts [10]. However, many objects in the environment such as chairs and desks have varied appearance and shapes. It is difficult to recognize such objects with fixed appearance models. On the other hand, many objects are intrinsically defined by their functions to users which entails a certain human activity associated with them. In this situation, user activities and their relationship with the objects are important context information to recognize such objects. Methods on recognizing objects by observing human activities have been investigated in previous works [11, 5, 6]. In [5], Moore et al. propose to use existing objects in the scene as context information to suggest specific models to recognize activities. They also use context information of other objects in the scene and user activities to recognize unknown objects. Veloso et al. define object classes with affordance properties and recognize them through human interactions [11]. In [6] image region patches are classified into object categories in an office room also by observing human interactions.

One common feature of the above work is the use of prior knowledge, in which the objects are defined through their interaction with humans. The above methods use Bayesian networks to model such relationships. While Bayesian network is a powerful tool for relational models, design of such a model becomes increasingly difficult when the number of objects/activities becomes large.

In this paper we propose to use a statistical relational model – Markov logic network to incorporate user activity as context information for object recognition. A knowledge base of user interactions with objects is constructed. In smart environment applications, context information is often conveniently expressed as rules and is often relational. Such relationships can be effectively represented in MLN in the syntax of first-order logic. A Markov logic network combines both probability and first-order logic [8]. First-order logic intuitively and compactly represents knowledge, while the probabilistic graphical model handles uncertainty effectively. With a Markov logic network it is flexible to construct or modify our prior knowledge on relations, since they are in the form of first-order logic formulas, while working directly on the graphical model itself can be difficult especially when the size of the model is large.

The system proposed in this work consists of three parts. 1. Analyze user's activity through a camera network. 2. Model prior knowledge of object functions in Markov logic network. 3. Infer location and identity of objects from observations of user activity. Instead of segmenting the image and labeling patches, we calculate the user's position in the room with calibrated cameras, and then infer the objects' location in the room coordinate.

The rest of the paper is organized as follows. In Sec. 2, we differentiate relationships between objects and user activities into three types, and introduce the objects that are recognized in our experiments. Sec. 3 presents the MLN knowledge base used to recognize objects. In Sec. 4, user activity recognition and object recognition results in our experiments are presented.

UCVP '09, November 5, 2009 Boston, MA

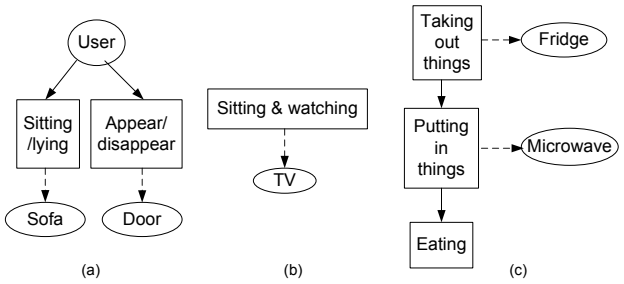

Figure 1: Examples of the three types of relations between the object and user pose/activity. (a) Direct relationship; (b) Spatial relationship; (c) Temporal relationship.

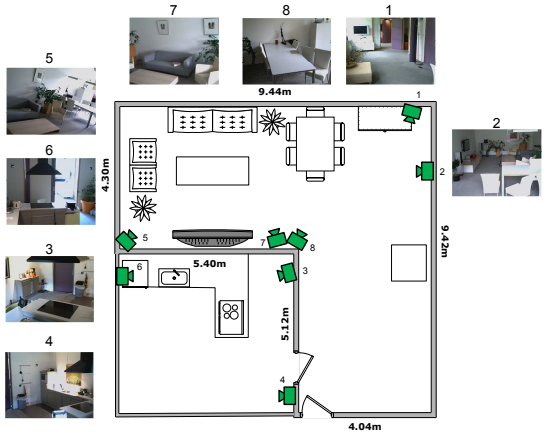

Figure 2: The Philips HomeLab floor-plan and camera views.

2. RECOGNIZING OBJECTS FROM USER ACTIVITIES

We define three types of relationships between objects and user activities (Fig. 1):

- Direct relationship of the object with user activities, in which the user's activity at time t and location l directly implies the likelihood of an object at location l. In Fig. 1(a), the sitting and lying activities of the user hint that there is probably a sofa, while the appearing and disappearing places of the user are likely to be doors.
- Spatial relationship between user activities and the object. Sometimes the user interacts with certain objects at a distance. The activity features such as the attention region of interest give us clues to infer the object's existence and location. For example, if the user is observed looking at a fixed direction for some duration, the likelihood of a TV at that location can increase (Fig. 1(b)).
- Temporal relationship between user activities may imply a single or several objects. This means that to use the object a sequence of actions are normally taken. For example, in Fig. 1(c), if the observation is that in the kitchen the user first takes out something from a place and then puts it into another place, it is likely that the first position implies a fridge while the second implies a microwave.

In this paper, we describe an embodiment of our proposed method to recognize objects in the living room and kitchen at Philips HomeLab. Cameras are installed on the walls, as shown in Fig. 2. HomeLab is part of the ExperienceLab at Philips ([1]), which is a prototyping environment to experiment with various technologies. Table. 1 lists the recognized objects in our experiment, their location (living room or kitchen), and the type of relationship with user activities used to define them.

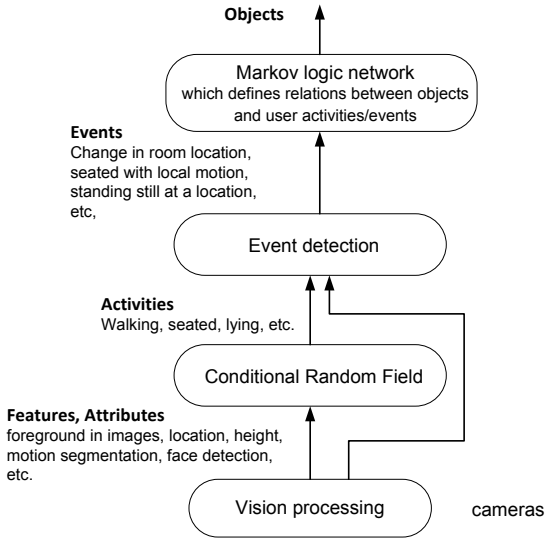

Figure 3: Flowchart of the system.

2.1 Time-driven v.s. event-driven formulation

Fig. 3 shows the flow of the operation. Vision processing in each camera extracts features from the images. The camera network is calibrated, so geometric features such as the position and height of the user can be calculated. A conditional random field (CRF) is used to infer the user's activity being walking, seated or lying, based on features composed of aspect ratio of the foreground in the image and the height of the user. The event generation block is used for the event-driven formulation, and this block is skipped for the time-driven formulation. In the highest level, observations are used to ground the MLN into a Markov random field. For each grid of the room, its probability of being an object type at each time step can be inferred.

In our knowledge base, objects with direct and spatial relationships (Table. 1) are defined using time-driven formulation, where each time step is a frame. Objects with temporal relationships are defined using event-driven formulation.

In our embodiment of the method for inferring objects in the HomeLab, the environment under observation is divided into grids of 50cm×50cm (Fig. 4). A dynamic model is used to reason object type of each grid at each time step. Therefore, it is necessary to handle sequences between time steps explicitly because MLN is a graphical model in general but not a dynamic model. So the relationships between user activity and objects need to specify how the object type *updates* based on the current knowledge of the object type given current observation at each time step, i.e., the object type

Table 1: The following objects are recognized in the system.

	living room	kitchen
Direct relationship	floor, chair, sofa	workspace
Spatial relationship	TV	
Temporal relationship	dining table	fridge, sink

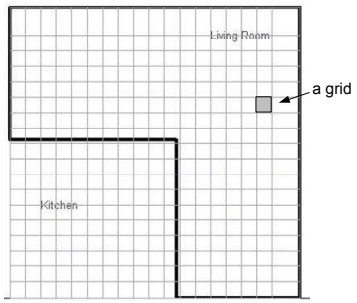

Figure 4: The environment is divided into grids. Probability of each object type is inferred for each grid.

of $grid_i$ at time t not only depends on observations at t, but also on the object type of this grid at time $t-1$. Therefore, we need to use a predicate "$Next(t_2, t_1)$" in MLN to indicate sequencing in time, which means t_2 is the next time step after t_1. If we update the object type at t based on only observations at t and the current knowledge on the object type at $t-1$, $Next(t_2, t_1)$ is sufficient since we only need to link neighboring time steps. We call this the time-driven formulation since each time step has the same length, and observations are input to MLN at each time step. Time-driven formulation applies to direct and spatial relationship. For objects with direct and spatial relationships, user activity of each frame (the smallest time step for the video sequence) is input as observation to the MLN.

However, in temporal relationship, an object may relate to a sequence of past observations at different time steps, and such observations have to be in order. In this case, $Next(t_2, t_1)$ is unable to describe observations not next to each other in time. So we introduce another predicate "$After(t_2, t_1)$" to describe that t_2 is after t_1 in time, but the interval between them can vary. Therefore, in the event-driven formulation, events related to our knowledge base are detected based on user activities and attributes, and such events are input to MLN only when they occur. The sequential order of events is indicated by the predicate $After(t_2, t_1)$. Note that direct and spatial relationships can be formulated as event-driven as well, if we treat user activity of each frame as an action event, and input it to MLN at each frame. $Next(t_2, t_1)$ can be replaced directly by $After(t_2, t_1)$. But for temporal relationship only event-driven formulation is able to handle time relations with varied intervals.

3. INFERRING OBJECTS WITH MARKOV LOGIC NETWORKS

Our knowledge base of relations between objects and user activities/events is encoded in the form of MLN [8]. An MLN has the syntax of first-order logic. It is a set of first-order logic formulas $\{F_i\}$, with a weight attached to each of them $\{w_i\}$. This weight indicates confidence of the relationship represented by the formula. With evidence an MLN is grounded into a Markov random field (MRF). Each formula F_i corresponds to a feature function $f_i(X)$ in the MRF, and the weight of the formula w_i equals the weight for the feature function, as in the log-linear probability density function of the MRF:

$$P(X) = \frac{1}{Z(w)} \exp\left(\sum_i w_i f_i(X)\right) \quad (1)$$

where $Z(w)$ is the partition function. The feature function $f_i(X)$ is an indicator function. $f_i(X) = 1$ when F_i holds given the set of variables X, and $f_i(X) = 0$ otherwise.

General inference methods on graphical models such as Markov chain Monte Carlo (MCMC) may be problematic to apply to MLNs when there are many hard constraints in the knowledge base. The hard constraints divide the state space into separate subspaces, which cannot be all traversed by a single run of MCMC. Even if initialization is at different positions, it is difficult to guarantee a proper coverage of all subspaces. [2] describes MC-SAT which combines satisfiability testing and MCMC. The satisfiability solver finds the separated subspaces by flipping the atoms (a predicate applied to a constant or variable, which is a binary random variable in the MRF) to maximize the number of satisfied formulas. This initializes MCMC and enables it to find the correct optimal quickly. The strength of MC-SAT is most evident when there are many hard constraints, which will be demonstrated in Sec. 4.2.1. The following two sections describe the two types of formulation of our knowledge base to recognize objects listed in Table. 1.

3.1 Knowledge construction for objects with direct and spatial relationships

Our first experiment recognizes objects in the living room through direct and spatial relationships. As in Table. 1, floor, chairs and sofa are defined with direct relationships, and TV is defined with a spatial relationship. The same knowledge base is applied to each grid, i.e., an MLN is constructed for each grid. The knowledge base for this experiment is listed in appendix A.

There are three predicates for this MLN:

- $Hastype(obj, t)$ means the grid has object type obj at time t. obj is a variable which can be one of $\{floor, chair, sofa, TV, unknown\}$.
- $Hasact(act, t)$ means the user has activity act at this grid at time t. act is a variable which takes one of $\{walking, seated, lying, watching, unknown\}$. Here $watching$ indicates that this grid is in the coverage of the user's gaze direction (inferred from head orientation), while the user is not necessarily in this grid. So $watching$ spatially relates this grid with the user at a distance.
- $Next(t2, t1)$ is used to specify sequencing between time steps as explained in Sec. 2.1.

3.2 Knowledge construction for objects with temporal relationship

Our second experiment recognizes the fridge, sink and workspace in the kitchen, and the dining table in the living room. These objects all relate directly or indirectly to activities in the kitchen. Fridge, sink and dining table are defined with temporal relationships, while workspace is defined with a direct relationship. A single MLN is constructed which reasons for all grids. The knowledge base for this experiment can be found in appendix B.

Events relevant to the knowledge base are defined in predicates:

- $EnterEvent(type, t)$ indicates the user changes room location at time t. $EnterEvent(type1, t)$ means the user enters the living room from outside; $EnterEvent(type2, t)$ means the user enters the kitchen from the living room; $EnterEvent(type3, t)$ means the user enters the living room from the kitchen.
- $ActEvent(act, t)$ indicates there is an event related to the user's activity at time t. $ActEvent(SS, t)$ means the user is standing still; $ActEvent(FS, t)$ means the user is standing still for the first time after changing room location; $ActEvent(SM, t)$ means the user is seated with local motion observed.

Predicates representing object types include the following: $IsFrdg(grd, t)$ means whether a grid grd is fridge at time t. Similarly,

$IsSink\ (grd, t)$, $IsTable\ (grd, t)$, $IsWS\ (grd, t)$ are predicates for sink, dining table and workspace, respectively. The predicate $IsAt\ (grd, t)$ is used to specify the position of the user (at grid grd) at time t. $After\ (t_2, t_1)$ states that t_2 is a time step after t_1, so that all the other atoms with t_2 or t_1 establish sequential orders.

4. EXPERIMENTS

In this section results from two experiments are described. In the first experiment, a time-driven formulation of the knowledge base is used to recognize floor, chair, sofa and TV in the living room. In the second experiment, an event-driven formulation of the knowledge base is applied to recognize the dining table in the living room, fridge, sink and workspace in the kitchen. Activity recognition and event generation modules are presented in Sec. 4.1. Results of object recognition are shown in Sec. 4.2.

4.1 Analyzing Activities and Events

Based on the object types of interest and our knowledge base, the user's activity is classified into walking, seated, lying and unknown with a conditional random field model (Sec. 4.1.2). Other activities such as watching and hand motion are detected separately with different image features (Sec. 4.1.1). Human activity analysis has been extensively studied with different image scenes, activity models and techniques. In [4] Moeslund et al. provide a comprehensive review for various levels of activity recognition and corresponding methods. In our sequences we face the challenge of an uncontrolled environment, where the lighting conditions may change and occlusion of the user may happen often. Besides, the image resolution is only 320×240, and the area occupied by the user may be very small since sometimes the user is far away from the camera. Considering the above conditions, robust low-level visual features need to be chosen.

4.1.1 Image features

Each camera implements background subtraction with adaptive density estimation ([12]) to retrieve the foreground. Given calibrated camera parameters, the position and height of the user can be calculated from foreground bounding boxes of individual cameras. If only one camera sees the user and the bounding box does not touch the bottom of the image, the lowest edge of the bounding box is assumed to be on the ground plane $z = 0$ to calculate the position (x, y) of the person. If multiple cameras detect the user, back-projecting the center vertical axis l_i of the bounding box from the camera center C_i gives intersecting planes. The (x, y) of the intersection line (which is vertical) is taken as the user's position. So even when the user is partially occluded by tables or chairs from the cameras, the center axes of bounding boxes from multiple cameras help resolve the user's location.

In addition to the location and pose, frontal and profile faces are detected to provide an approximate gaze direction of the user. If a face is detected, the center-line of the gaze is taken as the user-camera direction (in case of a frontal face) or its orthogonal direction (in case of a profile face). The gaze area is generated by adding an angle range δ around the center-line.

Local motion is detected as indicators for hand motion when the user is doing some tasks with hands. Motion segments with their directions are recorded for each frame. Within a window of sequential frames, if the position of such segments changes within a small region, and if the histogram of their directions is close to uniform (hand movement is likely to have random directions), then local motion is declared.

4.1.2 Activity recognition with CRF

Table 2: Confusion matrix of activity recognition with CRF.

	walking	seated	lying	NA
walking	0.900	0.028	0.018	0.054
seated	0.107	0.808	0.063	0.022
lying	0.004	0.116	0.794	0.086
NA	0.006	0	0	0.994

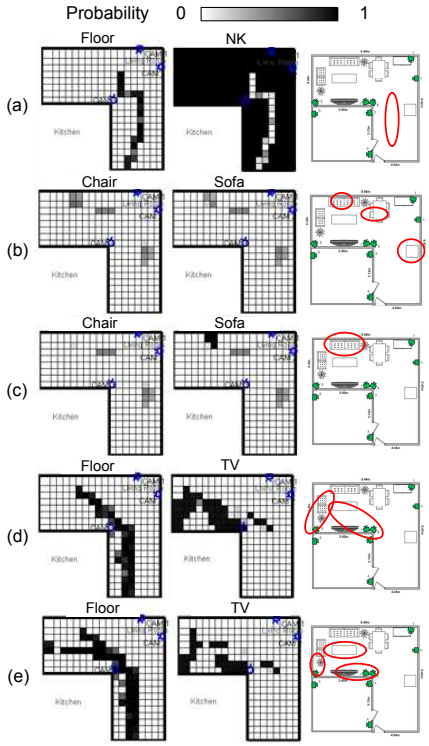

Figure 5: Probability maps of the objects at several time instances (a darker grid represents higher probability). Please refer to the text for a detailed explanation.

A temporal conditional random field is used for activity recognition in our experiment. CRF is used by Sminchisescu et al. in [9] to classify activities such as walking, jumping, running, picking or dancing based on 2D silhouettes or 3D joint positions. They demonstrate that contextual information helps to resolve ambiguities of similar gestures in the activity sequences. In our experiment, there is actually no underlying model for the user's body and his activities, since different users may have different patterns of walking, lying, etc. Thus a CRF model works well under this assumption since it does not assume a state model. The state variable can take $walking, seated, lying$ or $unknown$. The observation is a two-dimensional feature vector, the aspect ratio of the foreground bounding box and the height of the user.

The CRF is trained on 6 sets of video sequences (each with 3 views) consisting of 3 subjects. Each segment is about 90 seconds long, with a total of $6 * 3 * 90 * 10 frm/sec = 16200 frames$ of training samples. During inference, activity is inferred for each camera based on its bounding box aspect ratio and the user's height calculated collectively by the network. The test set is the same size as the training set. The confusion matrix of activity recognition on the test set is shown in Table 2.

The next task is to choose the camera with the most reliable ac-

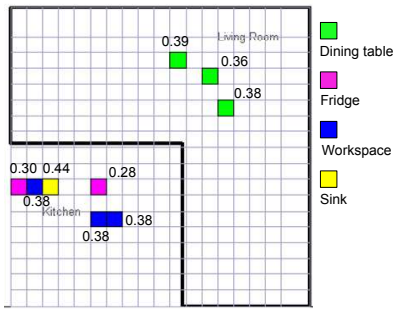

Figure 6: Four objects are recognized after observing three users. They are shown in four colors (Probability of the object is shown besides the grids).

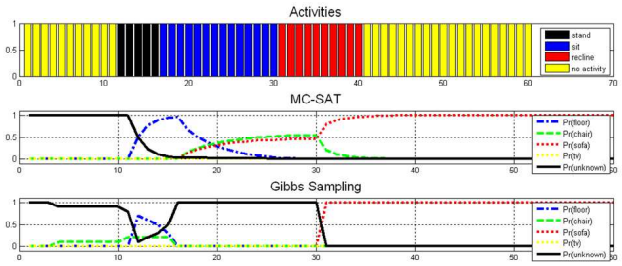

Figure 7: Activity of the user and the corresponding object probability curve. The first row shows different activities in colors (yellow: unknown; black: walking; blue: seated; red: lying; green: watching). Probability of object types from three inference methods are also shown in colors (blue: floor; green: chair; red: sofa; yellow: TV; black: unknown). The second row shows results from MC-SAT, and the third row shows results from Gibbs sampling.

tivity inference. Choosing the camera which has the highest probability of its activity inference is problematic, since we observe that $lying$ is more difficult to recognize than $seated$, and $seated$ is more difficult than $walking$. The reasons are that when the person is seated or lying on the sofa, he may be occluded from some views, and that image features can be nonrepresentative from some views. For example, when the person is lying, the frontal camera detects a more discriminative bounding box, while others may have bounding boxes confusing to other activities. Therefore we give decreasing priorities to $lying, seated$ and $walking$. That is, if one camera deducts $lying$, it is selected and the deduction is set to the state of the user.

The recognized activity is input to the MLN as an atom $Hasact$ (act, t) to each grid at each frame in the time-driven formulation.

4.1.3 Event generation

In the event-driven formulation in our experiment, event generation after activity recognition is needed to extract relevant events based on our knowledge base. The events can be easily derived from activity (Sec. 4.1.2) and other image features and user attributes (Sec. 4.1.1). The events include the following. $EnterEvent$ $(type, t)$ is detected from the location of the user, assuming that the limits of each room are defined. $ActEvent$ (SS, t) is triggered when the user is standing and his location doesn't change for a given duration. $ActEvent$ (FS, t) describes an $ActEvent$ (SS, t) that follows an $EnterEvent$ $(type, t)$. $ActEvent$ (SM, t) triggers when the user is seated with local motion observed.

4.2 Results of object recognition from MLN

In the first experiment with time-driven formulation, evidence atoms in the form of predicates in Sec. 3.1 are used to ground the MLN. Weights in the MLN are learned from 6 sequences including 3 users. Each sequence is about 1.5 mins (10 fps). The test set includes 4 subjects with 2 sequences each. Probability of each grid being a certain object is inferred at each time step. Fig. 5 illustrates object probability maps at six time instances when different objects are discovered (the darker the grid, the higher the probability). In Fig. 5(a), the user has been to only half of the living room. Most parts are still not known, which can be seen from the dark probability map of "NK". Some floor area is revealed. In Fig. 5(b), the user has been observed seated on the sofa chair on the right side of the living room, on a chair at the dining table, and on the sofa. Note that both Pr(Chair) and Pr(Sofa) get higher because of the rule that if "Seated" is observed it is likely to be either chair or sofa. In Fig. 5(c), the user has lied on the sofa. So in the sofa area, Pr(Sofa) increases while Pr(Chair) diminishes. In Fig. 5(d) when the user begins to watch TV the possible TV area is inferred from his gaze. It covers a big area because only three approximate face angles $(0^\circ, \pm 90^\circ)$ are detected from the camera which are then added with $\pm\delta$ (e.g., ± 15) degrees as the gaze range. But later on more activity clarifies some grid identities in the possible TV area (Fig. 5(e)). After the user sits at different places and walks around the coffee table to exit the living room, some more area is declared to be either chair or floor instead of TV. In the end the TV area is further confined.

In the second experiment, an event-driven formulation is used to recognize the fridge, sink, and workspace in the kitchen and the dining table. Evidence atoms of the MLN are described in Sec. 3.2. From three sequences from 3 users, the recognized objects are shown in Fig. 6. Compared to the room layout in Fig. 2, two grids are inferred as the fridge, whereas the right grid is a false positive. This is because our knowledge base doesn't agree with the user's behavior in this case (e.g., after the user comes back from grocery, he puts everything directly on the workspace instead of into the fridge). Three locations of the dining table are identified, since the three users sit at different chairs around the dining table. The sink is correctly recognized. The three grids recognized as workspace are either where the user cooks on the stove, or the bench area besides the fridge where the user puts things temporarily.

4.2.1 Comparison of inference methods

Fig. 7 illustrates an example of performance comparison between MC-SAT and Gibbs sampling on inference on MLN. The sequence of observations are $walking$, $seated$, and $lying$ with different durations. The result of MC-SAT shows that before frame 16 Pr(floor) rises, then Pr(chair) and Pr(sofa) rise with the same speed (since only the $seated$ action is observed), and finally the $lying$ activity increases Pr(sofa) and decreases all others. But Gibbs sampling does not yield correct inference as defined in the knowledge base, since the samples are initialized in the wrong subspace.

5. CONCLUSION

In this paper we consider user activities as context information to recognize objects in a home environment. This involves modeling prior knowledge of the relationship between the user's activity and objects. We use a Markov logic network to construct the knowledge base since it allows for intuitive and scalable construction of rules in first-order logic formulas. The knowledge base constructed

in our experiments is described. The whole system consists of feature extraction in the camera network, activity recognition with a CRF model, and object category inference with MLN. Experiments show that the system is able to locate objects that are defined in the knowledge base in the room coordinate.

Appendix A: Knowledge base of MLN – the time-driven formulation

1. The user's activity and the grid's object type are mutually exclusive.
   ∞ (ac!=ac')$\wedge$Hasact(ac,t)$\Rightarrow\neg$Hasact(ac',t)
   ∞ $\forall$t $\exists$ac Hasact(ac,t)
   ∞ (ob!=ob')$\wedge$Hastype(ob,t)$\Rightarrow\neg$Hastype(ob',t)
   ∞ $\forall$t $\exists$ob Hastype(t,ob)
2. If the activity is $walking$, it indicates that the grid is floor. w_i Hasact(walking,t) $\wedge$ Hastype(obj$_i$,t') $\wedge$ Next(t,t') $\Rightarrow$ Hastype (floor,t). w_i corresponds to different obj$_i$, which can be walking, seated, lying, unknown.
3. If the activity is $seated$, it increases likelihood of the grid being chair and sofa, given that current knowledge of the grid is unknown. However, if the grid is already sofa, it doesn't change its object type. w_i Hasact(seated,t) $\wedge$ Hastype (unknown,t') $\wedge$ Next(t,t') $\Rightarrow$ Hastype(sofa,t) $\vee$ Hastype(chair,t). ∞ Hasact(seated,t) $\wedge$ Hastype(sofa,t') $\wedge$ Next(t,t') $\Rightarrow$ Hastype (sofa,t).
4. If the activity is $lying$, it increases the probability of the grid being sofa. w_i Hasact(lying,t) $\wedge$ Hastype(obj$_i$,t') $\wedge$ Next(t,t') $\Rightarrow$ Hastype(sofa,t).
5. If the grid is under the gaze range of the user, and there haven't been other activities on this grid so far, then it is likely to be TV. w_i Hasact(watching,t) $\wedge$ (Hasact(unknown,t') $\vee$ Hasact(TV,t')) $\wedge$ Next(t,t') $\Rightarrow$ Hastype(TV,t).
6. If there is no activity, do not change any inference. ∞ Hasact(unknown,t) $\wedge$ Hastype(ob,t') $\wedge$ Next(t,t') $\Rightarrow$ Hastype(ob,t).

Appendix B: Knowledge base of MLN – the event-driven formulation

1. Consider the situation when the user comes home from grocery. If he first enters the living room from outside, then goes directly to the kitchen, and then stays at a location, it is possibly the fridge where he puts the grocery in. There is a pair of formulas. The first is w_1 EnterEvent(type1,t1) $\wedge$ EnterEvent(type2,t2) $\wedge$ After(t2,t1) $\wedge$ ActEvent(FS,t3) $\wedge$ IsAt(g,t3) $\wedge$ After(t3,t2) $\wedge$ IsFrdg(g,t2) $\Rightarrow$ IsFrdg(g,t3). The second formula is very similar to the first, but it changes the last atom on the left hand side IsFrdg(g,t2) into !IsFrdg(g,t2), and the weight to w_2. The difference between the two formulas is that it updates our belief on fridge differently given the current knowledge of it being or not being a fridge. We assign a bigger value to w_1 than w_2, since in the first formula, the events confirm our previous knowledge that this grid is a fridge.
2. If the user goes from the kitchen to the living room, and then sits at a location, with local motion observed, he is assumed to be eating at a dining table. There is a pair of formulas as well. The first one is w_3 EnterEvent(type3,t1) $\wedge$ ActEvent(SM,t2) $\wedge$ IsAt(g,t2) $\wedge$ After(t2,t1) $\wedge$ IsTable(g,t1) $\Rightarrow$ IsTable(g,t2). The second one changes IsTable(g,t1) to !IsTable(g,t1) and w_3 to w_4.
3. If the user is first observed seated with local motion, then he goes to the kitchen and stops at a location, he is assumed to be eating at the beginning and then puts plates into the sink in the kitchen. w_5 ActEvent(SM,t1) $\wedge$ EnterEvent(type2,t2) $\wedge$ After(t2,t1) $\wedge$ ActEvent(FS,t3) $\wedge$ IsAt(g,t3) $\wedge$ After(t3,t2) $\wedge$ IsSink(g,t2) $\Rightarrow$ IsSink(g,t3). The second formula changes IsSink(g,t2) to !IsSink(g,t2), and w_5 to w_6.
4. If the user stays still at certain locations in the kitchen, they are workspace. w_7 ActEvent(SS,t2) $\wedge$ IsAt(g,t2) $\wedge$ After(t2,t1) $\wedge$ IsWS(g,t1) $\Rightarrow$ IsWS(g,t2). The second formula changes IsWS(g,t1) to !IsWS(g,t1), and w_7 to w_8.

6. REFERENCES

[1] B. de Ruyter, E. van Loenen, and V. Teeven. User centered research in experiencelab. In *Ambient Intelligence*, volume 4794, pages 305–313, 2007.

[2] P. Domingos, S. Kok, D. Lowd, H. Poon, M. Richardson, and P. Singla. Markov logic. *Probabilistic Inductive Logic Programming*, pages 92–117, 2008.

[3] L. Fei-Fei, R. Fergus, and A. Torralba. Recognizing and learning object categories. *Tutorial at ICCV*, 2005.

[4] T. Moeslund, A. Hilton, and V. Kruger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 103(2-3):90–126, November 2006.

[5] D. Moore, I. Essa, and I. Hayes, M.H. Exploiting human actions and object context for recognition tasks. In *The Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 1, pages 80–86, 1999.

[6] P. Peursum, G. West, and S. Venkatesh. Combining image regions and human activity for indirect object recognition in indoor wide-angle views. In *Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05)*, volume 1, pages 82–89, 2005.

[7] A. Pinz. *Object Categorization*. Foundations and Trends in Computer Graphics and Vision by Now Publishers, 2006.

[8] M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 62:107–136, 2006.

[9] C. Sminchisescu, A. Kanaujia, and D. Metaxas. Conditional models for contextual human motion recognition. *Comput. Vis. Image Underst.*, 104(2):210–220, 2006.

[10] A. Torralba. Contextual priming for object detection. *Int. J. Comput. Vision*, 53(2):169–191, 2003.

[11] M. Veloso, F. von Hundelshausen, and P. E. Rybski. Learning visual object definitions by observing human activities. In *The proceedings of IEEE-RAS International Conference on Humanoid Robots (HUMANOIDS)*, 2005.

[12] Z.Zivkovic and F. van der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters*, 27(7):773–780, 2006.