DOI 10.3233/AIS-130240
IOS Press

Latent-Dynamic Conditional Random Fields for recognizing activities in smart homes

Yu Tong and Rong Chen*
College of Information Science and Technology, Dalian Maritime University, 116026, Dalian, China

Abstract. As the number of elderly people in our society increases, the need of assistive technologies in home becomes urgent. Existing techniques allow elderly people to be better assisted through monitoring what goes on in smart homes and inferring their activities from sensor data via a recognition model. However, there are various cases that existing models have difficulties in accommodating relational data. In this paper, we present an application of probabilistic graphical model – Latent-Dynamic Conditional Random Field – to detect the goals of the individual subjects when observations have long range dependencies or multiple overlapping features. To validate the proposed method, we apply it to recognize activities in two different datasets which were collected in smart homes. The results demonstrate that Latent-Dynamic Conditional Random Fields favorably outperform other models, especially when there are extrinsic dynamic activities changes and intrinsic actions (sub-activities).

Keywords: Ambient intelligence, smart home, sensor, Latent-Dynamic Conditional Random Fields, activity recognition

1. Introduction

By augmenting everyday artifacts with computation, sensing, and communication abilities, ambient intelligence (AmI) is emerging as omnipresent computing technology that can anticipate people's goals and intentions with contextual sensor data, and hence free them from tedious routine tasks in their daily lives at home and work [1]. One form of the many possible realizations of AmI [2] are smart homes, where AmI technology can not only monitor the functional health of residents who is independently at home, but also learn Activities of Daily Living (ADLs) they performed [3].

As the number of elderly people in our society increases, the need of such assistive technologies in the home becomes urgent. China has been an aging society since 1999, and in November 2010, had 178 million people over 60 years old, 119 million over 65, respectively accounting for 13.26% and 8.87% of the total population [4]. As age-related changes in the brain cause some decline in short-term memory and slowing in learning ability, elderly people run into all sorts of barriers in performing their daily routine tasks such as bathing, toileting, driving, cooking and handling finances. While aging society makes plenty of Chinese accepting public pensions, there's more demand of social care services and technical assistances.

In providing elderly with convenient and additional independent lives, the ability to monitor functional health of elderly people and to gain knowledge about their preferences, needs and habits are seen as the key approach for better assisting elderly people [5]. Sensor data are good indicators of the cognitive and physical capabilities of elderly when they are carrying out activities of daily living in a smart sensing environment. Activity recognition is to design models and algorithms to detect the goals of individual subjects from sensor data and to provide context sensitive services and assistance. In doing so, we need sensors that can observe what goes on in the house and a recognition model to infer the activities from sensor data. Activity recognition is rephrased as a classification problem of streaming sensor data, where the input variables are observation data, sequentially generated by various sensors, annotated with corresponding activities, and then used to train different activity recognition models.

*Corresponding author. E-mail: rchen@dlmu.edu.cn.

Several probabilistic models have been proposed to tackle this classification problem. Graphical models are traditional ways to represent the joint probability distribution $p(\boldsymbol{x}, \boldsymbol{y})$, where the variable $\boldsymbol{y}$ represents the attributes of the activities that we wish to predict, and the input variables $\boldsymbol{x}$ are sensor data representing our observations about the activities. Hidden Markov models (HMMs) [6] and Conditional Random Fields (CRFs) [7] are the mainstream techniques for recognizing activities with graphical models. While HMM models the joint distribution and independencies among inputs, CRF models the conditional distribution. Conditionally independencies are assumed in HMM to circumvent intractable models, but this restriction makes it difficult or impossible to accommodate long range dependencies among observations or multiple overlapping features of the observations. In contrast, CRF can afford the use of rich, local features of the input that can occur in relational sensor data, because the model is conditional, dependencies among the input variables $\boldsymbol{x}$ do not need to be explicitly represented. However, CRF does not always perform better than HMM for every case [8,9]. So when the activity does not involve many different types of sensors and the independence assumption is not affect, HMMs works better than CRFs [8].

To improve the recognition accuracy, this paper proposes a Latent-Dynamic Conditional Random Fields (LDCRF) [10] to accommodate relational data. As we all known, individuals may have to complete several actions (sub-activities) when they complete one task (activities), especially complex activities (i.e., Washing hands imply the following sub steps: moves to the kitchen, turns on water, uses hand soap, washes hands, dries hands). Also, individuals may complete one activity in different ways (i.e., making a phone implies the following ways: sits down during conversation, stands in one place during conversation, and walks around during phone call). These actions and different ways are the intrinsic sub-structure of the whole activity. When some phenomena have distinct sub-structure, models that exploit hidden state are advantageous [9]. The CRF approach models the transitions between labels, thus capturing extrinsic dynamics, but lacks the ability to represent internal sub-structure. In contrast to CRFs, LDCRF incorporates hidden state variables which model the substructure of sequences, thus not only can learn dynamics between labels but also can capture intrinsic sub-structure.

1.1. Related work

A variety of sensing modalities has been used in previous work on smart environment. Among them are tags, cameras and various sensors and detectors. When a large number of objects in a house are associated the Radio Frequency Identification (RFID) tags, the user wearing an RFID reader can detect which object is used, and the detected objects are used as input variables for activity recognition [11]. However, the user need to wear RFID reader and this may cause inconvenient for them. Another way is to exploit cameras installed in smart homes [12,13] for later activity reasoning, while Placelab is a smart house equipped with several hundred wall-mounted sensors, such as reed switches on doors and cupboards, temperature sensors and water flow detectors [14]. However, this may violate residents' privacy and affected by the light. In human-centric AmI setting, to make the sensors suitable for a smart-home environment targeting maximum comfort, it is important to select suitable devices and use non-obtrusive and pervasive sensors for sensing information.

Models for recognizing activities can be probabilistic based [15], logic based [16] or hand-crafted [9,17]. Because activities are performed in a real-world environment and may bring a great deal of variation in the manner that the activity is performed, people investigate the use of probabilistic models for this task which can represent and reason variations in the way an activity may be performed.

Probabilistic models can be categorized into static classification and temporal classification [18]. Typical static classifiers include naive Bayes and decision trees used in [19], k-nearest neighbor (k-NN) and Support Vector Machine used in [20]. In temporal classification, state-space models are typically used to enable the inference of hidden states (i.e., activity labels) given the observations [18]. Generative models can generate samples from the joint distribution whereas discriminative models usually need a lot of training data in order to get good effects. Hidden Markov model (HMM) [21,22] and dynamic Bayesian networks (DBN) [23] are the typical generative models in recognizing activities from sensor data. In recent years, Kasteren et al. [24] used hierarchical hidden Markov model (HHMM) to model actions and their correlations, and it was shown that hierarchical model significantly outperforms earlier techniques in activity recognition. For discriminative models, Conditional Random Fields (CRF) [7,8,25,26] is commonly applied and also gives good

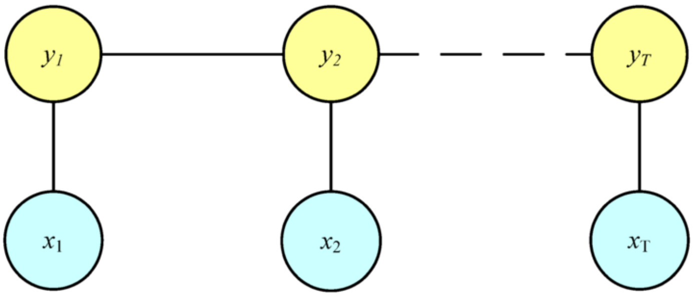

Fig. 1. Conditional Random Fields.

performance in activity recognition. Recently, many modifications of CRFs have been also proposed to recognize complex activities. For example, Wu et al. [27] proposed an algorithm using factorial conditional random field (FCRF) which is one type of DCRF for recognizing multiple concurrent activities. Hu and Yang [28] proposed an algorithm using skip-chain CRF (SCCRF) for modeling interleaving goals. However, there are few efforts made to model actions (sub-activities).

Latent-Dynamic Conditional Random Field (LDCRF) [9] and Hidden State Conditional Random Fields (HCRFs) [29] is the modifications of CRF and introduces hidden states and can capture intrinsic sub-structure. HCRFs can estimate a class given a segmented sequence, and have seen their applications in both the vision [30] and speech community [31]. However, HCRFs were applied to label segmented sequences, leaving segmentation as a preprocessing step. Previously used to model the sub-structure of gesture sequences [9], the LDCRF can be used to label un-segmented sequences, which overcomes the limitation of the HCRF and combine the strengths of CRFs and HCRFs. Since observed sensor data in smart homes are un-segmented sequences, we propose to recognize activities through LDCRF. To our knowledge, LDCRF has not been used to recognize activities in smart homes so far.

In this paper, the LDCRF is developed to model actions (sub-activities) for solving activity recognition problems, and validated in two different datasets generated in smart homes. To show the performance of recognition accuracy, we choose the other models for comparing the empirical results. The remainder of this paper is organized as follows. In the next section, we will present LDCRF that we used for activity recognition. After that, we validate our method and discuss the results. Finally, we conclude by summing up our findings.

2. Latent-Dynamic Conditional Random Fields for activity recognition

The task of activity recognition is to find a sequence of labels $\boldsymbol{y}_{1:T} = \{y_1, y_2,\ldots, y_T\}$ that best explains the sequence of observation $\boldsymbol{x}_{1:T} = \{\boldsymbol{x}_1, \boldsymbol{x}_2,\ldots, \boldsymbol{x}_T\}$ for a total of T time steps.

LDCRF takes root in CRF, which is one of important activity recognition models that can capture extrinsic dynamics between activity labels. Figure 1 picturizes CRF as an undirected graph consisting of sequential variable pairs of state variables y_t and observable variables $\boldsymbol{x}_t$, one every time step. CRF needs a lot of observations as training data to learn activity recognition model, and with a new observation sequence $\boldsymbol{x}_{1:T} = \{\boldsymbol{x}_1, \boldsymbol{x}_2,\ldots, \boldsymbol{x}_T\}$ can infer the activity labels $\boldsymbol{y}_{1:T} = \{y_1, y_2,\ldots, y_T\}$.

The same as CRF, LDCRF is also an undirected graphical model and there are a state variable y_t and an observable variable $\boldsymbol{x}_t$ at each time step. The difference is LDCRF also assume a vector of "substructure" variables $\boldsymbol{h} = \{h_1, h_2,..., h_T\}$ (see Fig. 2). These variables are not observed in the training examples and will therefore form a set of hidden variables in the model. Each h_t is a member of a set H_{y_t} of possible hidden states for the activity label $y_t \in \{1, 2,\ldots, K\}$, where K is the number of activity labels.

For different labels, we restrict them to have disjointed hidden states which will make K disjoint hidden states sets H_k, $k = \{1, 2,\ldots, K\}$. Then, we define a set H and let it contain the union of all H_k sets. So all possible hidden states will be are contained in H.

After introducing the hidden states, the LDCRF define a latent conditional model as

$$P(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta}) = \sum_{\boldsymbol{h}} P(\boldsymbol{y}|\boldsymbol{h},\boldsymbol{x},\boldsymbol{\theta}) P(\boldsymbol{h}|\boldsymbol{x},\boldsymbol{\theta}) \qquad (1)$$

where θ is the parameters of the model.

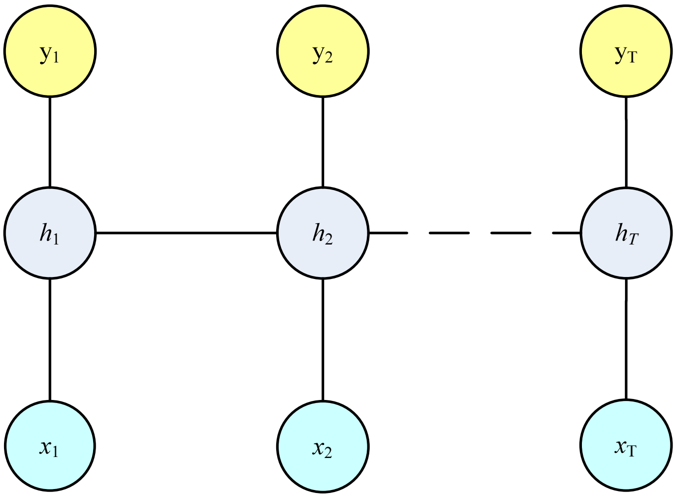

Fig. 2. Latent-Dynamic Conditional Random Fields.

Since for any $h_t \notin H_{y_t}$, there is $P(\boldsymbol{y}|\boldsymbol{h},\boldsymbol{x},\boldsymbol{\theta}) = 0$, the model Eq. (1) can be expressed as

$$P(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta}) = \sum_{h:\forall h_t \in H_{y_t}} P(\boldsymbol{h}|\boldsymbol{x},\boldsymbol{\theta}) \tag{2}$$

As the usual conditional random field formulation, $P(\boldsymbol{h}|\boldsymbol{x},\boldsymbol{\theta})$ can be defined as

$$P(\boldsymbol{h}|\boldsymbol{x},\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{x},\boldsymbol{\theta})} \exp\left(\sum_k \theta_k F_k(\boldsymbol{h},\boldsymbol{x})\right) \tag{3}$$

and the partition function Z is defined as

$$Z(\boldsymbol{x},\boldsymbol{\theta}) = \sum_{\boldsymbol{h}} \exp\left(\sum_k \theta_k F_k(\boldsymbol{h},\boldsymbol{x})\right) \tag{4}$$

where F_k is defined as

$$F_k(\boldsymbol{h},\boldsymbol{x}) = \sum_{t=1}^{m} f_k(h_{t-1}, h_t, \boldsymbol{x}, t) \tag{5}$$

The feature function $f_k(h_{t-1}, h_t, \boldsymbol{x}, t)$ is either a state function $s_k(h_t, \boldsymbol{x}, t)$ or a transition function $t_k(h_{t-1}, h_t, \boldsymbol{x}, t)$. State functions s_k depend on a single hidden variable and observations in the model while transition functions t_k depend on pairs of hidden variables.

For each state functions, there are

$$\begin{aligned} & s_k(h_t, x, t) \\ & = \left[s_k(h_t, x_t^1), s_k(h_t, x_t^2), \ldots, s_k(h_t, x_t^N)\right]^\mathrm{T} \end{aligned} \tag{6}$$

where $s_k(h_t, x_t^n)$ $(n = 1, 2, \ldots, N)$ is defined as

$$s_k(h_t = i, x_t^n) = \begin{cases} 1 & h_t = i \wedge x_t^n = 1 \\ 0 & h_t = i \wedge x_t^n = 0 \end{cases} \tag{7}$$

The number of state functions, s_k, will be equal to the length of the feature vector times the number of possible hidden states. In the case of activity recognition, the length of the feature is the number of sensors N. If our model has K activities and C hidden states per activities, then the total number of state functions, s_k, and total number of associated weights θ_k will be $N \times K \times C$.

For each hidden state pair (h', h''), the transitions functions t_k is defined as

$$\begin{aligned} & t_k(h_{j-1}, h_j, \boldsymbol{x}, j) \\ & = \begin{cases} 1 & \text{if } (h_{j-1} = h') \text{ and } (h_j = h'') \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{8}$$

It is worth noticing that the weights θ_k associated with the transition functions model both the intrinsic and extrinsic dynamics. Weights associated with a transition function for hidden states that are in the same subset H_{yt} will model the substructure patterns, while weights associated with the transition functions for hidden states from different subsets will model the external dynamic between activities.

2.1. Parameter estimation

To label a new sequence, we must know the parameter to build the model. The parameter can be classed obtained by training a set of labeled sequences (x_i, y_i) for $i = 1, 2,\ldots, n$.

The same as literature [7], we use the conditional log-likelihood of the training data and the log of a Gaussian prior to estimate the parameter $\boldsymbol{\theta}^*$:

$$L(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log P(\boldsymbol{y}_i|\mathbf{x}_i, \theta) - \frac{1}{2\sigma^2}\|\theta\|^2 \qquad (9)$$

The second term of the right hand in Eq. (9) is the log of a Gaussian prior with variance σ^2, i.e., $P(\boldsymbol{\theta}) \sim \exp(\|\theta\|^2/(2\sigma^2))$.

Given above defined, gradient ascent is used to search for the optimal parameters $\boldsymbol{\theta}^* = \text{argmax}L(\boldsymbol{\theta})$. There are two kinds of parameter, the θ_s associated with a state function and the θ_t associated with a transition function.

For one particular training sequence $\boldsymbol{x}_i = \{\boldsymbol{x}_1, \boldsymbol{x}_2,\ldots, \boldsymbol{x}_{T1}\}$ and $\boldsymbol{y}_i = \{y_1, y_2,\ldots, y_{T1}\}$, the gradient of $\log P(\boldsymbol{y}_i|\boldsymbol{x}_i, \theta)$ with respect to the parameters θ_s can be written as

$$\sum_{t=1}^{T1} \sum_{a=\mathrm{H}_{y_t}} P(h_t = a|\boldsymbol{y}_i, \boldsymbol{x}_i, \boldsymbol{\theta}) s_k(t, a, \boldsymbol{x}_i)$$
$$-\sum_{\boldsymbol{y}'} \sum_{t=1}^{T1} \sum_{a=\mathrm{H}_{y_t}} P(h_t = a|\boldsymbol{y}_i, \boldsymbol{x}_i, \boldsymbol{\theta}) s_k(t, a, \boldsymbol{x}_i) \qquad (10)$$

where

$$P(h_t = a|\boldsymbol{y}_i, \boldsymbol{x}_i, \boldsymbol{\theta}) = \frac{\sum_{\boldsymbol{h}:h_t = a\wedge(\forall h_t\in\mathrm{H}_{y_t})} P(\boldsymbol{h}|\boldsymbol{x}, \boldsymbol{\theta})}{\sum_{\boldsymbol{h}:\forall h_t\in\mathrm{H}_{y_t}} P(\boldsymbol{h}|\boldsymbol{x}, \boldsymbol{\theta})} \qquad (11)$$

Note that given our definition of $P(\boldsymbol{h}|\boldsymbol{x},\theta)$ in Eq. (3), the summations in Eq. (3) are simply constrained versions of the partition function Z over the conditional random field for $\boldsymbol{h}$. This can be easily shown to be computable in O($T1$) using belief propagation [32].

The gradient of our objective function with respect to the parameters θ_t can be derived in the same way, where $P(h_j = a, h_k = b|\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta})$ can also be computed efficiently using belief propagation. In our experiments, we performed gradient ascent with the BFGS optimization technique [33].

2.2. Inference

After the parameters $\boldsymbol{\theta}^*$ have been learned from training examples, our activity recognition is build.

For a new test sequence $\boldsymbol{x} = \{\boldsymbol{x}_1, \boldsymbol{x}_2,\ldots, \boldsymbol{x}_T\}$, we can estimate the most probable label sequence $\boldsymbol{y}^* = \{y_1, y_2, \ldots, y_T\}$ by maximizing our conditional model

$$\boldsymbol{y}^* = \text{argmax}P(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}^*) \qquad (12)$$

Since we assume each class label is associated with a disjoint set of hidden states, the previous equation can be rewritten as

$$\boldsymbol{y}^* = \text{argmax} \sum_{\boldsymbol{h}:\forall h_t\in\mathrm{H}_{y_t}} P(\boldsymbol{h}|\boldsymbol{x}, \boldsymbol{\theta}^*) \qquad (13)$$

The label y_t associated with the t-th sensor event in testing can be estimated as follows:

First, for all possible hidden states $a \in \mathrm{H}$, compute the marginal probabilities $P(h_t{=}a|\boldsymbol{x}, \boldsymbol{\theta}^*)$ using belief propagation.

Next, compute the sum of marginal probabilities of the disjoint hidden states sets H_k as

$$S_k = \sum_{\forall a\in\mathrm{H}_k} P(h_t{=}a|\boldsymbol{x}, \boldsymbol{\theta}^*),\ k = 1, 2, \ldots, K$$

Finally, find the k maximizing S_k and take it as the label of sensor event at time t.

3. Validation

In this section, we will validate the performance of activity recognition with LDCRF by using two sensor datasets. In doing so, we first introduce simply HMM, CRF, SVM and HCRF as the baseline for comparing with our method. After introducing two datasets and some measurement criteria, we show the empirical results.

3.1. Introduction of baseline models for comparing

Hidden Markov Model Hidden Markov Model (HMM) is a generative probabilistic model consisting of a hidden variable $\boldsymbol{y}$ and an observable variable $\boldsymbol{x}$ at each time step. HMM models joint probability as follows:

$$p(y_{1:T},\boldsymbol{x}_{1:T}) = p(y_1)p(\boldsymbol{x}_1|y_1)\sum_{t=2}^{T}p(y_t|y_{t-1})p(\boldsymbol{x}_t|y_{\mathrm{t}}) \tag{14}$$

We model one activity label as hidden variable $\boldsymbol{y}$ and the sensors observable as observable variable $\boldsymbol{x}$. Thus, the number of hidden variables is equal to that of activity labels. In training and testing, we will use the method described in literature [15] to recognize activities.

Conditional Random Fields We will use linear-chain CRF for activity recognition and models conditional probability as follows:

$$p(y_{1:T}|\boldsymbol{x}_{1:T}) = \frac{1}{Z(\theta)}\exp\sum_{k=1}^{K}\theta_k\sum_{t=1}^{T}f_k(\boldsymbol{x},y_{t-1},y_t,t) \tag{15}$$

where f_k is the feature functions which is either a state function $s_k(y_t,\boldsymbol{x},t)$ or a transition function $t_k(y_{t-1},y_t,\boldsymbol{x},t)$. The nominator of this function is straight forward and fast to compute; the complexity of the model lies in the computation of the normalization term $Z(\theta)$ which takes into account all possible state sequences corresponding to the given observation sequence.

We train a single CRF chain model and let every activity has a corresponding state label. In training and testing, we will also use the method described in literature [15] to recognize activities.

Support Vector Machine SVM is one of the standard tools for machine learning and data mining. The SVM decision function is defined as follows:

$$f(y)=\sum_{i=1}^{N}\alpha_i K(x_i,y)+b \tag{16}$$

Here y is the unclassified tested vector, x_i are the support vectors and α_i their weights and b is a constant bias. $K(x,y)$ is the kernel function introduced into SVM to solve the nonlinear problems by performing implicit mapping into a high-dimensional feature space.

The SVM does not encode the dynamics between activity labels and the training set was decomposed into frame-based samples. In our experiment, we use multiclass SVM [34].

Hidden State Conditional Random Fields Since HCRFs cannot model dynamics between activities, we trained the HCRF on segmented sub-sequence. Given one segmented sensor events $\boldsymbol{x}$ and corresponding label y, we define a conditional probabilistic model as

$$P(y|\boldsymbol{x},\theta)=\sum_{\boldsymbol{h}}P(y,\boldsymbol{h}|\boldsymbol{x},\theta) \tag{17}$$

where $\boldsymbol{h} = \{h_1, h_2,\ldots, h_{\mathrm{m}}\}$ are hidden state and cannot observed on training examples.

We trained HCRF model on all activity labels as done in [29]. The trained HCRF model is applied on the new sequence using a sliding window of fixed size. The class label with the highest likelihood is assigned to the frame at the center of the sliding window. The number of hidden states and the length of the sliding window size (referred as NL in our experiments) is decided at training stage.

3.2. Two datasets that collected in ambient intelligence environment

This subsection will present two datasets which are collected in ambient intelligence environments. One is kasteren Dataset which is collected in a three-room apartment where a 26-year-old man lives and there are 14 state-change sensors were installed in this apartment. Location of sensors installed is list as follows: Microwave, Hall-Toilet door, Hall-Bathroom door, Cups cupboard, Fridge, Plates cupboard, Front door, Dishwasher, Toilet Flush, Freezer, Pans Cupboard, Washing machine, Groceries Cupboard, Hall-Bedroom door (see Fig. 3.) Sensors were left unattended and installed in an unobtrusive and transparent way.

In this apartment, the user lives freely and his ADL primary involves seven different activities which are listed as follows:

1. Leaving house.
2. Toileting.
3. Showering.
4. Sleeping.
5. Preparing breakfast.
6. Preparing dinner.
7. Preparing a beverage.

The times at which no activity is annotated is referred to as ‘Idle’. Activities were annotated by the subject himself using a blue-tooth headset.

The Dataset is collecting for 28 days in the apartment which resulted in 2120 sensor events and 245 activity instances. Table 1 shows the number of separate instances of activities and the percentage of time each activity takes up in the data set. This table clearly shows how some activities occur very frequently,

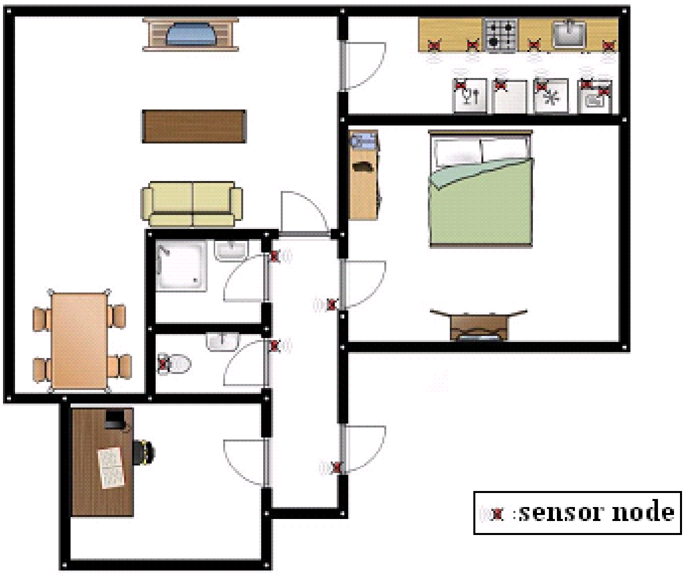

Fig. 3. Floor plan of three-room apartment, red rectangle boxes indicate sensor nodes.

Table 1

Number of instances and percentage of time activities occur in the dataset

	Number of instances	Percentage of time
Idle	–	11.5%
Leaving	34	56.4%
Toileting	114	1.0%
Showering	23	0.7%
Sleeping	24	29.0%
Breakfast	20	0.3%
Dinner	10	0.9%
Drink	20	0.2%

while others that occur less frequently have a longer duration and therefore take up more time.

Another dataset is "WSU Apartment Test bed, ADL adlnormal" which is collected in a smart apartment testbed located on the WSU campus [35]. This dataset is build to recognize and assess the consistency of Activities of Daily Living that individuals perform in their own homes. Sensors in the apartment include monitor motion sensors (M), temperature sensors (T), water sensors (W), burner sensors (B), phone sensors (P), and item sensors (I) (see Fig. 4). The motion sensors are located on the ceiling approximately 1 meter apart to locate the resident, the Voice over IP (VOIP) technology captures phone usage and switch sensors to monitor usage of the phone book, a cooking pot, and the medicine container.

The dataset records 24 WSU undergraduate students performing five ADLs, one at a time. Activities include both basic and more complex ADLs that are found in clinical questionnaires and are listed as follows:

1. Telephone Use: Look up a specified number in a phone book, call the number, and write down the cooking directions given on the recorded message.
2. Hand Washing: Wash hands in the kitchen sink.
3. Meal Preparation: Cook oatmeal on the stove according to the recorded directions, adding brown sugar and raisins (from the kitchen cabinet) once done.
4. Eating and Medication Use: Eat the oatmeal together with a glass of water and medicine (a piece of candy).
5. Cleaning: Clean and put away the dishes and ingredients.

Letting the activities we want to recognize be the states and the collected sensor data to be the observations, we can apply HMM, CRFs and LDCRF on activity recognition. Next subsection will give two measurement criteria to evaluate the recognition accuracy and in the third and fourth subsection, we will perform the experiments on the two smart environments datasets.

3.3. Measurement criteria

The measures we first adopt were used in literature [8], where time slice accuracy is proposed to measure the percentage of correctly classified time

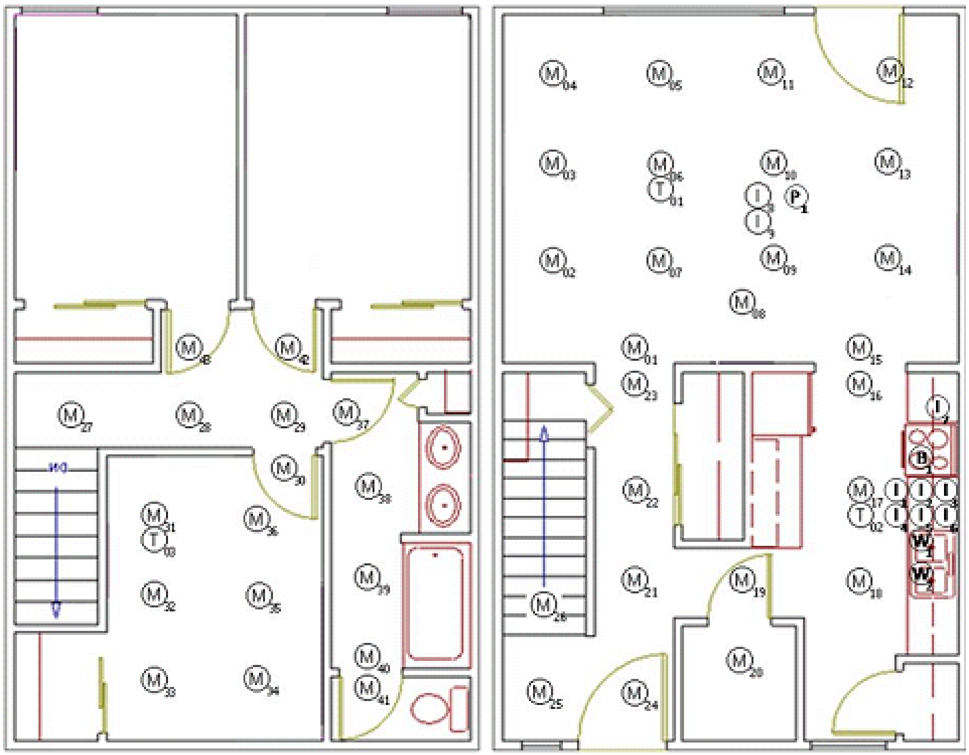

Fig. 4. Sensors in the WSU apartment test bed, monitor motion (M), temperature (T), water (W), burner (B), phone (P), and item use (I).

slices and the class accuracy is used to evaluate the average percentage of correctly classified time slices per class. Formally, they are defined as follows:

$$\text{Time slice}: \frac{\sum_{n=1}^{N}\left[\text{inferred}(n)=\text{true}(n)\right]}{N}$$

$$\text{Class}: \frac{1}{C}\sum_{c=1}^{C}\left\{\frac{\sum_{n=1}^{N_c}\left[\text{inferred}_c(n)=\text{true}_c(n)\right]}{N_c}\right\}$$

where $[a = b]$ is a binary indicator giving 1 when true and 0 when false. N is the total number of time slices, C is the number of classes and N_c the total number of time slices for class c.

Activity recognition is a multi-label classification problem in essence, so we also choose the corresponding measures [36] to evaluate the performance of our models. Quality of the overall classification is assessed in two ways: Macro-averaging and Micro-averaging. We use Macro-averaging because it treats all classes equally and take $\beta = 1$ which weights recall and precision evenly. Our measures for multi-label classification are listed in Table 2.

In the table, tp_i is the number that correctly recognized as the i-th class (true positives), tn_i is the number that correctly recognized but do not belong to the i-th class (true negatives), and fp_i is the number that incorrectly recognized as the i-th class (false positives) while fn_i is the number that incorrectly recognized but do not belong to the i-th class (false negatives).

Also we present the confusion matrix of different models. The columns in each table show the ground-truth labels and the rows show the predicted labels. The values in confusion matrix are the count of each activity is predicted correctly and the count that are predicted as other labels.

The error of training data is defined as

$$\text{Error} = \frac{\sum_{n=1}^{N}\left[\text{inferred}(n)\neq\text{true}(n)\right]}{N}$$

by minimizing which we can define the number of hidden variables.

3.4. Experiment 1

In this experiment we will recognize activities in kasteren Dataset [8] base on probabilistic models. The sensor readings are divided in time slices of length $\Delta t = 60$ second and label the activity for each

Table 2

Measures for multi-label classification

Measure	Formula	Evaluation focus
Average Accuracy	$\left(\sum_{i=1}^{l}\frac{tp_i+tn_i}{tp_i+fn_i+fp_i+tn_i}\right)\Big/l$	The average per-class effectiveness of a classifier
Error Rate	$\left(\sum_{i=1}^{l}\frac{fn_i+fp_i}{tp_i+fn_i+fp_i+tn_i}\right)\Big/l$	The average per-class classification error
Precision	$\left(\sum_{i=1}^{l}\frac{tp_i}{tp_i+fp_i}\right)\Big/l$	An average per-class agreement of the data class labels with those of a classifiers
Recall	$\left(\sum_{i=1}^{l}\frac{tp_i}{tp_i+fn_i}\right)\Big/l$	An average per-class effectiveness of a classifier to identify class labels
Fscore	$\frac{(\beta^2+1)Precision_M Recall_M}{\beta^2 Precision_M + Recall_M}$	Relations between data's positive labels and those given by a classifier based on a per-class average

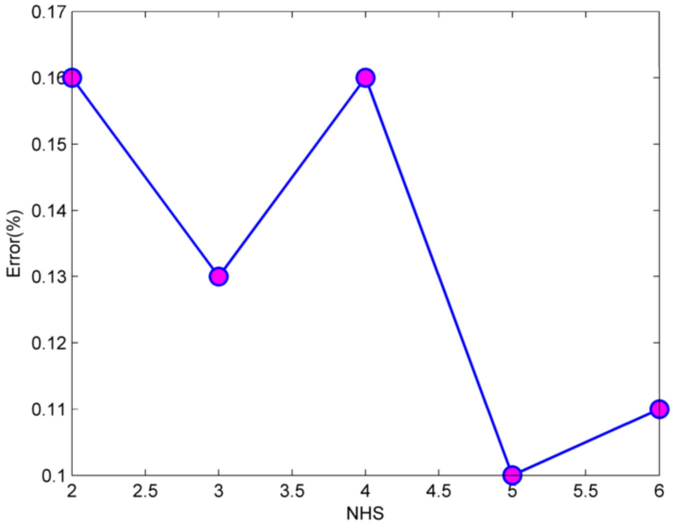

Fig. 5. Error on training data set with 16000 training data.

slice. The dataset can divide into 40006 time slices and there are three representations: raw, changepoint and last. The raw sensor representation gives a "1" to time slices when the sensor is firing and a "0" otherwise; the changepoint representation means when sensor reading changes the sensor gives a "1" where the sensor reading changes and last representation means last sensor that changed state continues to give "1" and changes to "0" when a different sensor changes state. Since literature [8] has demonstrated that "changepoint + last" representation achieves the overall highest class accuracy, we only take "changepoint + last" representation for both models and inference methods.

We do experiment using previous 20000 time slices and take previous 16000 time slices for training and the following 4000 for testing. When training with LDCRF, we varied the number of hidden states (NHS) (from 2 to 6 states) and compute the error on training data set. Figure 5 is the error changes with different hidden states and by which we get NHS = 5 that minimizing the error for testing.

Table 3 is the result for different models with the measures used in literature [8] which shown that

Table 3

Time slice and class accuracies for the different models with 16000 training data

	SVM	HMM	CRF	HCRF(NHS = 5,NL = 50)	LDCRF(NHS = 5)
Rate	0.9868	0.9845	0.9868	0.9496	0.9932
Class rate	0.5877	0.7520	0.7105	0.4042	0.8753

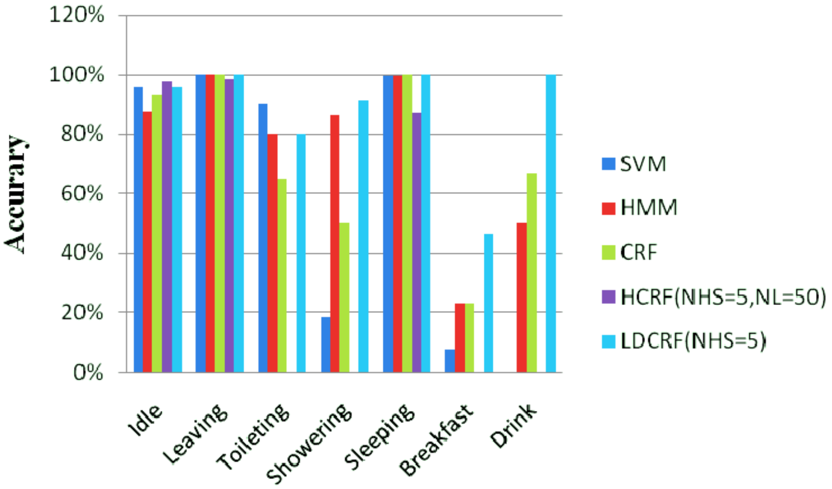

Fig. 6. The accuracies of different model for eight individual activities with 16000 training data.

LDCRF have achieved both highest time slice accuracy and highest class accuracy. For the Time slice accuracy, LDCRF gives the highest result. For the Class accuracy rate, LDCRF gives an average accuracy as high as 87.53%. However, other models are all lower than 80%.

Furthermore, Fig. 6 compares the accuracies of different models for the individual activities. Because the activity ‘Dinner’ is not happened in the test dataset, there are only seven activities in the figure. From the figure, we can see that LDCRF get best result for all activities compared other models, HCRF gets worst result and SVM follows HCRF. Figure 6 also shows that the activities which take more time and generate more sensor events (i.e., Leaving) tend to be recognized with greater accuracy. The activities which are very quick (i.e., Drink) and do not generate enough sensor events (i.e., breakfast) to be distinguished from other activities result lower recognition results. For this case, our model LDCRF can also get better result.

The reason is that LDCRF not only can learn dynamics between labels but also can capture intrinsic sub-structure. Because HMM and CRF can model dynamics between labels, the result of them follows LDCRF. Since HCRF is designed for segmented sub-sequence and use a sliding window when testing, it gets worst result. Because SVM cannot encode the dynamics between activity labels, it gets worse result than HMM, CRF and LDCRF. For activity ‘Idle’, which means no activity carried out, HCRF and SVM get better result. This is because the activity involves many sensors and learning dynamics between labels contribute nothing since there is not regular pattern in it.

3.5. *Experiment 2*

In the above experiment, there are only seven activities in the test dataset. Since the activity ‘Preparing dinner’ is very sparse and even did not happen in the test dataset, we cut the previous 20000 time slices equally into two datasets in the second experiment in order to include all the activities in the test dataset for training and testing. In this way, we first do experiment with varying length of sub-sequences, and then validate the LDCRF with the optimal length. Apart from the measure used in literature [8], the measures for multi-label classification are also used to evaluate the performance of our models.

First of all, we hypothesize that test results are influenced by the length of sub-sequences. To validate

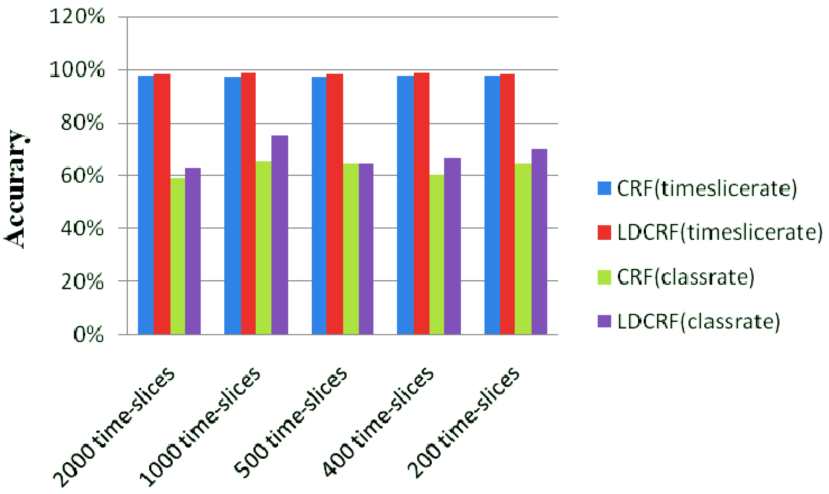

Fig. 7. The time slice rate and class rate of CRF and LDCRF for different length of sub-sequences.

Table 4

Time slice and class accuracies of CRF and LDCRF for different length of sub-sequences

N time slices	2000 time slices	1000 time slices	500 time slices	400 time slices	200 time slices
CRF(time slice rate)	97.67%	97.29%	97.29%	97.79%	97.83%
LDCRF(time slice rate)	98.27%	98.76%	98.08%	98.51%	98.37%
CRF(class rate)	58.97%	65.33%	64.66%	60.41%	64.52%
LDCRF(class rate)	63.27%	75.21%	64.35%	66.86%	70.35%

our hypothesis, we divide the train sequences into different length of the sub-sequences and train with them. For example, cut the training sequence into five equal sub-sequences which has 2000 time slices and train with them.

Figure 7 picturizes the time slice rate and class rate for CRF and LDCRF for different length of sub-sequences and Table 4 shows their corresponding values. As they reveal, recognition accuracy are influenced by the length of sub-sequences and the sub-sequences with 1000 time slices get better result. This is because sub-sequences of different length have different sub-structures. Anyway, LDCRF is not subject to this influence when it has better recognition accuracy than CRF.

The above experiment reveals that it is a good choice to divide the train sequences into ten equal sub-sequences with 1000 time slices. So we cut training sequence into ten equal sub-sequences and everyone have 1000 time slices, and then recognize activities via LDCRF and baseline models. When training with LDCRF, we vary the number of hidden states (NHS) (from 2 to 6 states) and compute the error on training data set. Figure 8 is the line chart of error rates varying with different hidden states, and by which we figure out that NHS = 6 minimizes the error for testing.

Table 5 summarizes the results for different models with the measures used in literature [8], while Table 6 the results for different models with the measures for multi-label classification and the time that used to infer the test activity labels with the trained models. They show that the Class rate and Recall is equal and they both evaluate the average percentage of correctly classified time slices per class. Because HCRF is designed for segmented sub-sequence, it gets worst result. Because SVM cannot encode the dynamics between activity labels, it gets bad Recall and Fscore compared with HMM, CRF and LDCRF. CRF gets better Average Accuracy, Precision and Fscore than HMM, but Recall is worse. LDCRF has achieved best results and small Error Rate. The time used for LCRF to infer the test activity labels is short despite slower than SVM, HMM, CRF. LDCRF is the good choice for activity recognition when the time is not critical.

The Confusion Matrix of SVM, HMM, CRF, HCRF and LDCRF are listed in Tables 7–11. From the tables, we can see that LDCRF get better results for all activities compared with SVM and HCRF.

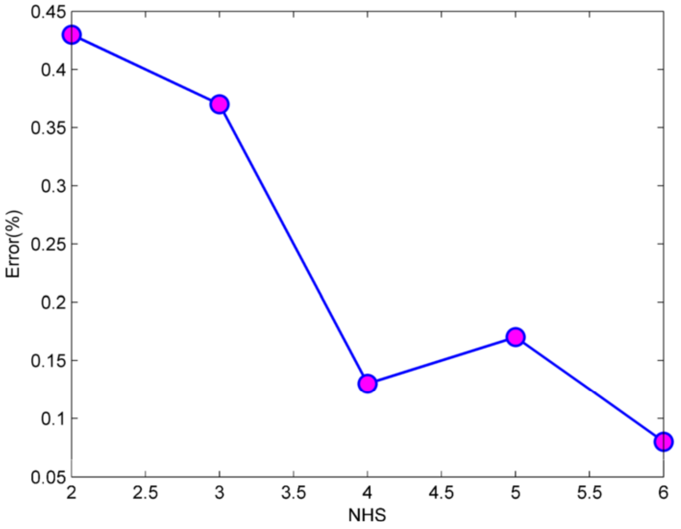

Fig. 8. Error on training data set with 10000 training data.

Table 5

Time slice and class accuracies for the different models

	SVM	HMM	CRF	HCRF(NHS = 5, NL = 30)	LDCRF(NHS = 6)
Rate	0.9346	0.9170	0.9354	0.8961	0.9556
Class rate	0.5280	0.6834	0.6332	0.4063	0.7402

Table 6

The result for different models with the measures for multi-label classification

	SVM	HMM	CRF	HCRF(NHS = 5, NL = 30)	LDCRF(NHS = 6)
Average Accuracy	0.98365	0.9793	0.9836	0.9740	0.9889
Error Rate	0.01635	0.0208	0.0162	0.0260	0.0111
Precision	0.6340	0.6257	0.7144	0.3710	0.7994
Recall	0.5280	0.6834	0.6332	0.4063	0.7402
Fscore	0.2881	0.3267	0.3357	0.1939	0.3843
Time(second)	0.4249	0.3568	0.3297	25.0298	2.2657

Compared with CRF, LDCRF get better results for seven activities. For simple activity which involve little act (showing), CRF may perform better than LDCRF. As compared with HMM, LDCRF gets better result for five activities. There are some activities in which HMM perform better than CRF and LDCRF, this is because: they do not involve many different types of sensors and thus they will make less intrinsic actions (sub-activities). In addition, a separate model $p(\boldsymbol{x}|\boldsymbol{y})$ is learned in HMM for each class and Bayes rule is used to calculate the posterior probability $p(\boldsymbol{y}|\boldsymbol{x})$ for a novel point. But in the case of CRF and LDCRF, a single model is used for all classes when calculating $p(\boldsymbol{y}|\boldsymbol{x})$ directly and parameters are learned by maximizing the conditional likelihood $p(\boldsymbol{y}|\boldsymbol{x})$.

3.6. Experiment 3

We also used the dataset collected in “WSU Apartment Testbed, ADL adlnormal” to further test the accuracy of our algorithm. The adlnormal dataset includes sensor event data for 24 individuals who were asked to perform the 5 ADL activities, yielding a total of 120 activity traces containing 6425 sensor events (time slices). This data reflects normal performance of the targeted activities. This experiment

Table 7

Confusion Matrix of SVM

	Idle	Leaving	Toileting	Showering	Sleeping	Breakfast	Dinner	Drink
Idle	756	5	9	42	308	12	55	8
Leaving	5	6253	3	0	0	0	1	1
Toileting	3	0	57	3	8	0	1	0
Showering	8	0	2	14	0	0	0	0
Sleeping	3	0	0	1	2239	0	0	0
Breakfast	0	0	0	0	0	5	3	0
Dinner	154	0	0	0	0	11	21	6
Drink	2	0	0	0	0	0	0	1

Table 8

Confusion Matrix of HMM

HMM2	Idle	Leaving	Toileting	Showering	Sleeping	Breakfast	Dinner	Drink
Idle	496	8	4	2	303	1	16	3
Leaving	5	6248	1	0	0	0	1	0
Toileting	4	0	59	3	13	0	2	2
Showering	135	2	7	55	0	0	0	0
Sleeping	3	0	0	0	2239	0	0	0
Breakfast	0	0	0	0	0	9	2	0
Dinner	284	0	0	0	0	17	60	7
Drink	4	0	0	0	0	1	0	4

Table 9

Confusion Matrix of CRF

	Idle	Leaving	Toileting	Showering	Sleeping	Breakfast	Dinner	Drink
Idle	805	69	10	14	315	8	56	4
Leaving	5	6189	1	0	0	0	1	2
Toileting	3	0	43	0	6	0	0	2
Showering	86	0	17	46	2	0	0	0
Sleeping	10	0	0	0	2232	0	0	0
Breakfast	0	0	0	0	0	10	0	0
Dinner	20	0	0	0	0	8	24	3
Drink	2	0	0	0	0	2	0	5

Table 10

Confusion Matrix of HCRF

	Idle	Leaving	Toileting	Showering	Sleeping	Breakfast	Dinner	Drink
Idle	675	137	56	28	402	13	66	10
Leaving	2	6106	3	0	0	0	0	0
Toileting	7	0	0	0	0	0	0	0
Showering	43	0	0	32	0	0	0	0
Sleeping	6	0	3	0	2106	1	0	0
Breakfast	1	0	0	0	0	0	0	0
Dinner	197	0	9	0	14	9	15	6
Drink	0	0	0	0	18	5	0	0

Table 11

Confusion Matrix of LDCRF

	Idle	Leaving	Toileting	Showering	Sleeping	Breakfast	Dinner	Drink
Idle	888	3	8	16	305	5	19	3
Leaving	4	6255	1	0	0	0	2	1
Toileting	3	0	53	0	9	0	1	0
Showering	13	0	9	44	0	0	0	0
Sleeping	4	0	0	0	2241	0	0	0
Breakfast	0	0	0	0	0	10	3	0
Dinner	17	0	0	0	0	13	56	3
Drink	2	0	0	0	0	0	0	9

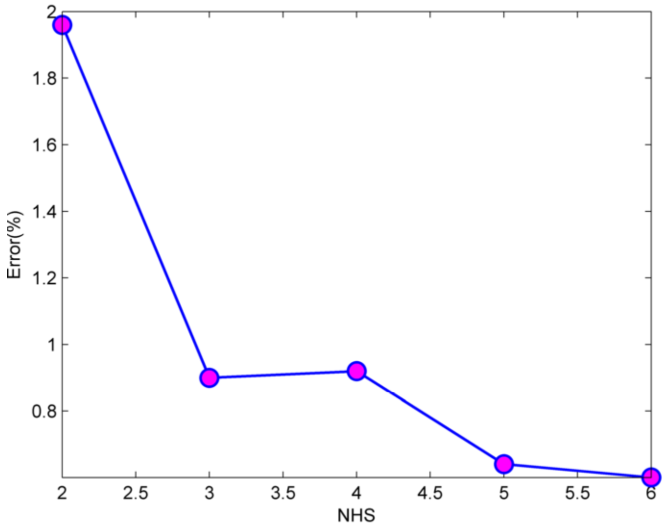

Fig. 9. Error on training data set with 5000 training data.

Table 12

Time slice and class accuracies for the different models with 5000 training data

	SVM	HMM	CRF	HCRF(NHS = 5,NL = 40)	LDCRF(NHS = 6)
Rate	0.6779	0.9165	0.9502	0.8144	0.9811
Classrate	0.5535	0.9124	0.9400	0.7987	0.9785

takes the previous 5000 time slices as train sequence and next 1425 time-slices as test sequence.

For the train sequence, we cut it into fifty equal sub-sequences with 100 time slices, then training with all the models. For LDCRF, we varied the number of hidden states (from 2 to 6 states) and computed the error on training data set. Figure 9 show that NHS = 6 is the minimized error for the different hidden states.

As shown in Table 12, the time slice accuracy of LDCRF is 3.9% higher than CRF and 6.46% than HMM, the class accuracy is 3.85% higher than CRF and 6.61% than HMM. The Time slice accuracy and the Class accuracy of SVM are 30.32% and 42.50 % lower than LDCRF, respectively, since it cannot encode the dynamics between activity labels. The results demonstrate again that LDCRF not only can best recognize time slices, but also can get the highest average classification accuracy.

Figure 10 compares recognition accuracy for different models in the five activities from which we can see the recognition accuracies of LDCRF get good results. There are five activities in which LDCRF are always better than SVM and HCRF. LDCRF get higher accuracy than CRF in four out of five activities and the accuracy of LDCRF is only 1% lower than CRF in activity ‘Cooking’. For simple activity which involve little act (Cook), CRF may get better result than LDCRF. LDCRF get higher accuracy than HMM for three activities and the accuracy of LDCRF is only 1% lower than HMM in the other two activities. For activity ‘cooking’ and’ making Phone-Call’, HMM performs better than LDCRF, this is because the sensors used in ‘cooking’ and other activities are very different, thus the independence assumption of HMM work better than the dependence of LDCRF.

Table 13 compares the results of different models with the measures for multi-label classification and the time that used to infer the test activity labels with the trained models. Because HCRF is designed for segmented sub-sequence and SVM cannot encode the dynamics between activity labels, they get bad results compared with HMM, CRF and LDCRF. CRF is better than HMM but worse than LDCRF. The Confusion Matrix of SVM, HMM, CRF, HCRF and LDCRF are listed from Table 14 to Table 18.

The experiments in the two datasets validate our LDCRF can get better results compared with baseline model SVM, HMM, CRF and HCRF. HCRF gets

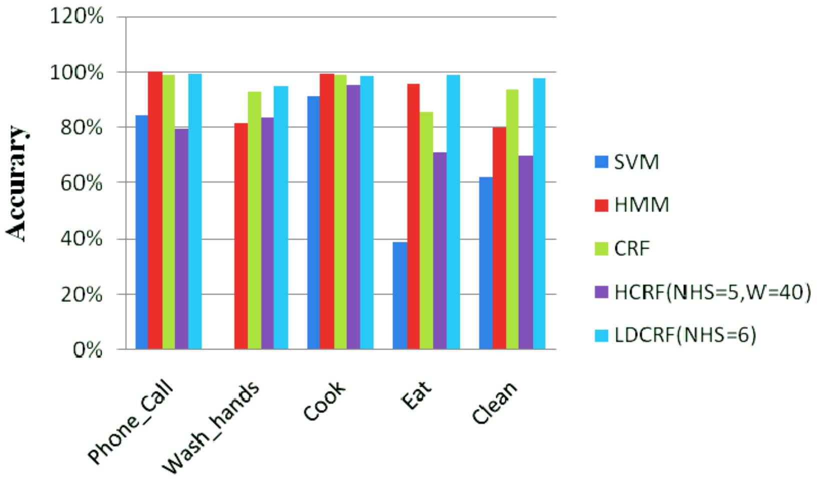

Fig. 10. The accuracies of different model for five individual activities.

Table 13

The result for different models with the measures for multi-label classification

	SVM	HMM	CRF	HCRF(NHS = 5,NL = 40)	LDCRF(NHS = 6)
Average Accuracy	0.8712	0.9666	0.9801	0.9258	0.9924
Error Rate	0.1288	0.0334	0.0199	0.0742	0.0076
Precision	0.7076	0.9110	0.9446	0.7710	0.9702
Recall	0.5535	0.9124	0.9400	0.7987	0.9785
Fscore	0.3106	0.4558	0.4711	0.3923	0.4872
Time(second)	0.1842	0.0486	0.0345	3.2158	0.1190

Table 14

Confusion Matrix of SVM

	Phone_Call	Wash_hands	Cook	Eat	Clean
Phone_Call	164	13	0	45	18
Wash_hands	0	0	0	0	0
Cook	0	23	452	42	93
Eat	30	29	2	77	55
Clean	0	33	43	33	273

Table 15

Confusion Matrix of HMM

	Phone_Call	Wash_hands	Cook	Eat	Clean
Phone_Call	194	12	0	0	0
Wash_hands	0	80	3	0	0
Cook	0	6	494	9	0
Eat	0	0	0	188	89
Clean	0	0	0	0	350

worst result for it is designed for segmented subsequence. Because SVM cannot encode the dynamics between activity labels, it gets bad Recall and Fscore compared with HMM, CRF and LDCRF. Since LDCRF can model relation between activities and actions in activity, it gets best result. Since CRF and HMM which are used in [8] cannot capture the relation between actions, they get lower recognition effect than LDCRF.

4. Conclusions

To allow elderly people to be better assisted with context-aware services, this paper introduces the

Table 16

Confusion Matrix of CRF

	Phone_Call	Wash_hands	Cook	Eat	Clean
Phone_Call	192	4	0	0	0
Wash_hands	2	91	0	0	0
Cook	0	3	492	2	0
Eat	0	0	4	169	29
Clean	0	0	1	26	410

Table 17

Confusion Matrix of HCRF

	Phone_Call	Wash_hands	Cook	Eat	Clean
Phone_Call	154	2	0	19	6
Wash_hands	28	82	16	11	97
Cook	0	2	473	21	0
Eat	6	12	8	126	23
Clean	6	0	0	0	293

Table 18

Confusion Matrix of LDCRF

	Phone_Call	Wash_hands	Cook	Eat	Clean
Phone_Call	193	4	0	0	0
Wash_hands	1	93	4	0	0
Cook	0	1	489	1	0
Eat	0	0	4	195	11
Clean	0	0	0	1	428

LDCRF model for inferring people's activities from sensor data in smart environment. The LDCRF model incorporates hidden state variables which can model the sub-structure of a class sequence and learn dynamics between class labels.

To validate the proposed method, LDCRF as well as the model SVM, HMM, CRF and HCRF are used in comparison for recognizing activities on two datasets. Different measurement criteria are used to measure the recognizing effect for those models. The results show LDCRF outperforms other techniques on an average. The results also confirm that modeling actions (sub-activities) and the underlying correlations do contribute to accurate activity recognition.

Each contribution to activity recognition will brings us one step closer to the realization of Ambient Intelligence. This progress allows the AmI systems to understand what the user wants and needs as well as also where, when and to whom to deliver a service. As future work, we plan to expand the LDCRF to recognize activities in multiple-residents environments.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No. 61175056), the Fundamental Research Funds for the Central Universities (No. 3132013335), and IT Industry Development of Jilin Province.

References

[1] J.C. Augusto and P. McCullagh, Ambient intelligence: concepts and applications, *Computer Science and Information Systems* **4**(1) (2007), 1–28.

[2] J.C. Augusto, Ambient intelligence: The confluence of pervasive computing and artificial intelligence, in: *Intelligent Computing Everywhere*, Springer, Berlin, 2007, pp. 213–234.

[3] V. Wadley, O. Okonkwo, M. Crowe and L.A. Ross-Meadows, Mild cognitive impairment and everyday function: Evidence of reduced speed in performing instrumental activities of daily living, *American Journal of Geriatric Psychiatry* **16**(5) (2007), 416–424.

[4] http://www.worldcrunch.com/china-tops-100-million-elderly-crunching-numbers-demographic-time-bomb/culture-society/china-tops-100-million-elderly-crunching-the-numbers-of-a-demographic-time-bomb/c3s3653/.

[5] A. Coronato, Uranus: A middleware architecture for dependable AAL and vital signs monitoring applications, *Sensors* **12** (2012), 3145–3161.

[6] L. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, *Proc. of the IEEE* **77**(2) (1989), 257–286.

[7] J. Lafferty, A. McCallum and F. Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data, in: *International Conference on Machine Learning*, 2001, pp. 282–289.

[8] T.V. Kasteren, A. Noulas, G. Englebienne and B. Krose, Accurate activity recognition in a home setting, in: *Proc. of the 10th International Conference on Ubiquitous Computing*, Seoul, Korea, 2008.

[9] E. Nazerfard, B. Das, L.B. Holder and D.J. Cook, Conditional random fields for activity recognition in smart environments, in: *Proc. of the 1st ACM International Health Informatics Symposium*, 2010, pp. 282–286.

[10] L.P. Morency, A. Quattoni and T. Darrell, Latent-dynamic discriminative models for continuous gesture recognition, in: *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, 2007.

[11] D.J. Patterson, D. Fox, H.A. Kautz and M. Philipose, Fine-grained activity recognition by aggregating abstract object usage, in: *ISWC*, IEEE Computer Society, 2005, pp. 44–51.

[12] L. McCowan, D. Gatica-Perez, S. Bengio, G. Lathoud, M. Barnard and D. Zhang, Automatic analysis of multimodal group actions in meetings, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **27** (2005), 305–317.

[13] N. Nguyen, S. Venkatesh and H. Bui, Recognising behaviours of multiple people with hierarchical probabilistic model and statistical data association, in: *British Machine Vision Conference*, 2005.

[14] S.S. Intille, K. Larson, E.M. Tapia, J. Beaudin, P. Kaushik, J. Nawyn and R. Rockinson, *Using a Live-In Laboratory for Ubiquitous Computing Research Pervasive Computing*, 2006, pp. 349–365.

[15] T.L.M. van Kasteren, G. Englebienne and B.J.A. Kröse, Human activity recognition from wireless sensor network data: Benchmark and software, *Activity Recognition in Pervasive Intelligent Environments* **4** (2011), 165–186.

[16] F. Mastrogiovanni, A. Scalmato, A. Sgorbissa and R. Zaccaria, Smart environments and activity recognition: A logic-based approach, in: *Atlantis Ambient and Pervasive Intelligence*, L. Chen, C.D. Nugent, J. Biswas and J. Hoey, eds, Atlantis Press, 2011, pp. 83–109.

[17] J. Fogarty, C. Au and S.E. Hudson, Sensing from the basement: A feasibility study of unobtrusive and low-cost home activity recognition, in: *UIST 06: Proc. of the 19th Annual ACM Symposium on User Interface Software and Technology*, New York, NY, USA, 2006, pp. 91–100.

[18] L. Wang, T. Gu, X.P. Tao, H.H. Chen and J. Lu, Recognizing multi-user activities using wearable sensors in a smart home, *Pervasive and Mobile Computing* **7**(3) (2011), 287–298.

[19] B. Logan, J. Healey, M. Philipose, E. Munguia-Tapia and S. Intille, A long-term evaluation of sensing modalities for activity recognition, in: *Ubi Comp 2007*, LNCS, Vol. 4717, pp. 483–500.

[20] T. Huynh, U. Blanke and B. Schiele, Scalable recognition of daily activities from wearable sensors, in: *LoCA 2007*, LNCS, Vol. 4718, pp. 50–67.

[21] G. Singla, D. Cook and M. Schmitter-Edgecombe, Recognizing independent and joint activities among multiple residents in smart environments, *Ambient Intelligence and Humanized Computing Journal* (2010), 57–63.

[22] G. Singla, D. Cook and M. Schmitter-Edgecombe, Tracking activities in complex settings using smart environment technologies, *International Journal of BioSciences, Psychiatry and Technology* **1**(1) (2009), 25–35.

[23] M. Philipose, K.P. Fishkin, M. Perkowitz, D.J. Patterson, D. Fox, H. Kautz and D. Hähnel, Inferring activities from interactions with objects, *IEEE Pervasive Computing* **3**(4) (2004), 50–57.

[24] T.M. Kasteren, G. Englebienne and B.A. Kröse, Hierarchical activity recognition using automatically clustered actions, in: *Ambient Intelligence*, 2011, pp. 82–91.

[25] D.L. Vail, M.M. Veloso and J.D. Lafferty, Conditional random fields for activity recognition, in: *International Conference on Autonomous Agents and Multi-Agent Systems*, 2007.

[26] K.C. Hsu et al., Strategies for inference mechanism of conditional random fields for multiple-resident activity recognition in a smart home, in: *Twenty Third International Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems (IEA-AIE)*, Córdoba, Spain, 2010.

[27] T. Wu, C. Lian and J.Y. Hsu, Joint recognition of multiple concurrent activities using factorial conditional random fields, in: *AAAI Workshop PAIR 2007*.

[28] D. Hao Hu and Q. Yang, *CIGAR: Concurrent and interleaving goal and activity recognition*, in: *AAAI*, 2008, pp. 1363–1368.

[29] A. Quattoni, M. Collins and T. Darrel, Conditional random fields for object recognition, in: Advances in Neural Information Processing Systems, Vol. 17, 2005.

[30] S. Wang, A. Quattoni, L. Morency, D. Demirdjian and T. Darrell, Hidden conditional random fields for gesture recognition, in: *CVPR*, 2006.

[31] A. Gunawardana, M. Mahajan, A. Acero and J.C. Platt, Hidden conditional random fields for phone classification, in: *INTERSPEECH*, 2005.

[32] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, 1988.

[33] D.C. Liu, J. Nocedal, On the limited memory bfgs method for large scale optimization, *Math. Program* **45**(3) (1989), 503–528.

[34] C.W. Hsu and C.J. Lin, A comparison of methods for multiclass support vector machines, in: *IEEE Transactions on Neural Networks*, 2002.

[35] D. Cook and M. Schmitter-Edgecombe, Assessing the quality of activities in a smart environment, in: *Methods of Information in Medicine*, 2009.

[36] M. Sokolova and G. Lapalme, A systematic analysis of performance measures for classification tasks, *Information Processing & Management* **45**(4) (2009), 427–437.