Joint Recognition of Multiple Concurrent Activities using Factorial Conditional Random Fields

Tsu-yu Wu and Chia-chun Lian and Jane Yung-jen Hsu
Department of Computer Science and Information Engineering
National Taiwan University
yjhsu@csie.ntu.edu.tw

Abstract
Recognizing patterns of human activities is an important enabling technology for building intelligent home environments. Existing approaches to activity recognition often focus on mutually exclusive activities only. In reality, people routinely carry out multiple concurrent activities. It is therefore necessary to model the co-temporal relationships among activities. In this paper, we propose using Factorial Conditional Random Fields (FCRFs) for recognition of multiple concurrent activities. We designed experiments to compare our FCRFs model with Linear Chain Condition Random Fields (LCRFs) in learning and performing inference with the MIT House_n data set, which contains annotated data collected from multiple sensors in a real living environment. The experimental results show that FCRFs can effectively improve the F-score in activity recognition for up to 8% in the presence of multiple concurrent activities.

Introduction
Recognizing patterns of human activities is an essential building block for providing context-aware services in an intelligent environment, either at home or in the work place. Take as an example, the application of elder care in a home setting, activity recognition enables the intelligent environment to monitor an elder’s activities of daily living and to offer just-in-time assistance, playing the role of a responsible care giver.

Automatic activity recognition presents difficult technical challenges. To tackle the problem, one often makes the simplifying assumption by focusing on mutually exclusive activities only. In other words, most existing approaches do not take into account the co-temporal relationship among multiple activities. Such solutions are not accurate enough in practice as people routinely carry out multiple activities concurrently in their daily living.

House_n is an ongoing project by the Department of Architecture at Massachusetts Institute of Technology (Intille et al. 2006b) that offers a living laboratory for the study of ubiquitous technologies in home settings. Hundreds of sensing components are installed in nearly every part of the home, which is a one-bedroom condominium. The living lab is being occupied by volunteer subjects who agree to live in the home for varying lengths of time. Sensor data are collected as the occupants interact with digital information in this naturalistic living environment. As a result, the House_n data set (Intille et al. 2006a) is potentially a good benchmark for research on activity recognition. In this paper, we use the data set as the material to design our experiment.

Our analysis of the House_n data set reveals a significant observation. That is, in a real-home environment, people often do multiple activities concurrently. For example, the participant would throw the clothes into the washing machine and then went to the kitchen doing some meal preparation. As another example, the participant had the habit of using the phone and the computer at the same time. As is shown in figure 1, there are many cases in the House_n data set in which people would perform multiple activities concurrently.

![Figure 1: Annotated activities in the House_n data set from 10 AM to 1 PM.](image)

Traditionally, research on activity recognition has focused on dealing with mutually exclusive activities. In other words, they assumed that there is at most one activity occurring at every time point. For any given time point, the primary concern is to label with the most probable activity. Given that multiple concurrent activities do exist, we should no longer make the assumption of mutually exclusive activities. In addition, we should not ignore the fact that multiple activities interact with each other, as it could be of great help to take such relationship into account.

Conditional Random Fields (CRFs) (Lafferty, McCallum,
conditional independence assumption required by Hidden Markov Models (HMMs). The CRF model has been applied to learning patterns of human behavior (Chieu, Lee, & Kaelbling 2006; Sminchisescu, Kanaujia, & Metaxas 2006; Liao, Fox, & Kautz 2007). Nevertheless, as mentioned above, previous research ignored the possible interactions between multiple concurrent activities.

To address this problem, we advocate using a factorial conditional random fields (FCRFs) (Sutton & McCallum 2004) model to conduct inference and learning from patterns of multiple activities. Compared with basic CRFs, FCRFs utilize a structure of distributed states to avoid the exponential complexity problem. In addition, the FCRF model accommodates the relationship among multiple concurrent activities and can effectively label their states.

This paper is organized as follows. At first, we introduce the CRFs and FCRFs model. And then we present FCRFs for multiple concurrent activities recognition including the model design, the inference algorithm, and the learning method. Finally, we describe our experiment for performance evaluation of this model, followed by the conclusion and future work.

**Conditional Random Fields**

CRFs are undirected graphical models conditioned on observation sequences which have been successfully applied to sequence labelling problems such as bioinformatics (Sato & Y. 2005; Liu et al. 2005), natural language processing (Lafferty, McCallum, & Pereira 2001; Sha & Pereira 2003; Sutton & McCallum 2004; Sarawagi & Cohen 2005) and computer vision (Sminchisescu, Kanaujia, & Metaxas 2006; Vishwanathan et al. 2006).

Unlike generative models such as Dynamic Bayesian Networks (DBNs) and HMMs, CRFs are conditional models that relax the independence assumption of observations and avoid enumerating all possible observation sequences. Maximum Entropy Markov Models (MEMMs) are an alternative conditional models, but they suffer from the label bias problem (Lafferty, McCallum, & Pereira 2001) due to per-state normalization. In contrast, CRFs are undirected structures and globally normalized, thereby solving the label bias problem.

Let $G$ be an undirected graph consisting of two vertex sets $X$ and $Y$, where $X$ represents the set of observed random variables and $Y$ represents the set of hidden random variables conditioned on $X$. Given graph $G$, let $C$ be the set of maximum cliques, where each clique includes vertices $X_c \in X$ and $Y_c \in Y$. For a given observation sequence $Y$, CRFs define the conditional probability by the potential function $\Phi(X_c, Y_c)$ such that

$$P(Y | X) = \frac{1}{Z(X)} \prod_{c \in C} \Phi(X_c, Y_c)$$

where $Z(X)$ is the normalization constant.

**An FCRFs Model for Multiple Concurrent Activities Recognition**

**Model Design**

Let $Y_t^i$ be a random variable whose value represents the state of activity $i$ at time $t$ and $Y^t = \{Y_1^t, Y_2^t, \ldots, Y_N^t\}$ where $N$ is the number of activities to be recognized. In a total time interval $T$, let $Y = \{Y_1^1, Y_2^1, \ldots, Y_T^1\}$ be a sequence of vector $Y^t$. We can define the observation sequence $X = \{X_1^1, X_2^1, \ldots, X^1_T\}$ in the same way. Suppose that there is a graph $G = \{V, E\}$. Let each element in $Y^t$
and $X^t$ be a vertex in $G$. Edges in $G$ represent the relationships between these random variables.

Figure 3 shows a sample FCRF model for recognition of three concurrent activities. The FCRF is represented in a dynamic form by unrolling the structure of two time slices. There are two sets of edges standing for different relational meanings. The edge set $E_c$, which includes pairs of activity variables in the same time slice, represents the co-temporal relationship; the edge set $E_t$, which includes pairs of activity variables across time slices, represents the temporal relationship.

Figure 3: An FCRF example of three concurrent activities.

We define pair-wise potential functions $\Phi_e(Y^t_i, Y^t_j, X)$ for each edge $(Y^t_i, Y^t_j)$ in $E_c$ and $\Phi_t(Y^t_i, Y^{t+1}_i, X)$ for each edge $(Y^t_i, Y^{t+1}_i)$ in $E_t$. And we define the local potential function $\Psi(Y^t_i, X)$ for each vertex $Y^t_i$ in $G$. The FCRFs will be determined as

$$p(Y|X) = \frac{1}{Z(X)} \left( \prod_{t=1}^{T} \prod_{i,j} \Phi_e(Y^t_i, Y^t_j, X) \right)$$

$$\left( \prod_{t=1}^{T-1} \prod_{i} \Phi_t(Y^t_i, Y^{t+1}_i, X) \right) \left( \prod_{t=1}^{T} \prod_{i} \Psi(Y^t_i, X) \right)$$

where $Z(X)$ is the normalization constant.

The potential functions $\Phi_e$, $\Phi_t$ and $\Psi$ are then defined by a set of feature functions $f = \{f_1, f_2, \ldots, f_K\}$ and their corresponding weights $\lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_K\}$ such that

$$\Phi_e(Y^t_i, Y^t_j, X) = \exp \left( \sum_k \lambda_k f_k(Y^t_i, Y^t_j, X) \right)$$

$$\Phi_t(Y^t_i, Y^{t+1}_i, X) = \exp \left( \sum_k \lambda_k f_k(Y^t_i, Y^{t+1}_i, X) \right)$$

$$\Psi(Y^t_i, X) = \exp \left( \sum_k \lambda_k f_k(Y^t_i, Y^t_i, X) \right)$$

Now, we have formally defined the FCRFs.

### Inference Algorithm

Given any observation sequence $O$, there are two kinds of inference tasks of concern. One task is to compute the marginal probability of each node pair, which will be used in the learning algorithm to be introduced later. The other task is to perform MAP (Maximum A Priori) inference to infer the most possible sequence of activities states.

There are many inference algorithms for CRFs, including the forward-backward algorithm, mean field free energy, junction tree, and loopy belief propagation (LBP), which is one of the most popular CRFs inference methods. Even though it can only approximate the probabilities and does not guarantee convergence for graphs with loops, LBP has been shown to be effective (Sutton & McCallum 2004; Vishwanathan et al. 2006; Liao, Fox, & Kautz 2007).

#### Sum-Product Algorithm

To compute the marginal probability, the LBP sum-product algorithm is adopted. We introduce a "message" $m_{ij}(A_j)$ for each pair of neighboring nodes $A_i$ and $A_j$, which is a distribution sent from node $A_i$ to node $A_j$ about which state variable $A_j$ should be in.

All messages $m_{ij}(A_j)$ are initialized as uniform distributions over $A_j$. The message $m_{ij}(A_j)$ sent from node $A_i$ to its neighbor $A_j$ is updated based on all the messages to $A_i$ received from its neighbors $A_k$, except those from node $A_j$.

$$m_{ij}(A_j) = \kappa \sum_i \left( \Phi(A_i, O) \Phi(A_i, A_j, O) \prod_{k \neq i,j} m_{ki}(A_i) \right)$$

where $\Phi(A_i, O)$ is the local potential, $\Phi(A_i, A_j, O)$ is the pair-wise potential, and $\kappa$ is the normalization constant.

The messages propagate through the CRF graph until they converge, when every message varies for less than a threshold. Given that LBP does not guarantee convergence, we have to set a limit on the maximum number of iterations. Even though the update order of messages may affect the convergent speed, empirical study (Sutton & McCallum 2004) showed that a random schedule is sufficient.

After LBP converges, the marginal probability of nodes $A_i$ and $A_j$ is then determined as

$$P(A_i, A_j) = \kappa' \Theta(A_i, A_j, O) \prod_{k \neq i,j} m_{ki}(A_i) \prod_{l \neq i,j} m_{lj}(A_j)$$

where

$$\Theta(A_i, A_j, O) = \Psi(A_i, O) \Psi(A_j, O) \Phi(A_i, A_j, O)$$

$k$ enumerates over all neighbors of $A_i$, $l$ enumerates over all neighbors of $A_j$ and $\kappa'$ is the normalization constant.

#### MAP Inference

To do MAP inference, we replace summation with maximization in the message update rule of the sum-product algorithm. We have

$$m_{ij}(A_j) = \kappa \max_i \left( \Psi(A_i, O) \Phi(A_i, A_j, O) \prod_{k \neq i} m_{ki}(A_i) \right)$$

where $\kappa$ is the normalization constant.
After LBP converges, the MAP probability of node \( A_i \) is defined as

\[
P(A_i) = \Psi(A_i, O) \prod_{j \neq i} m_{ji}(A_i),
\]

where \( A_j \) is the neighbor of \( A_i \).

To do the inference, we can label every hidden variable by choosing the most likely value according to the MAP probability.

**Learning Algorithm**

The purpose of learning is to determine the weight \( \lambda_k \) for each feature function \( f_k \). We can do this by maximizing the log-likelihood of the training data. Given the training data \( D = \{ A(1), O(1); A(2), O(2) \ldots A(M), O(M) \} \), the log-likelihood \( L(D|\lambda) \) is defined as follows.

\[
L(D|\lambda) = \sum_{m} \log P(A(m)|O(m))
\]

The partial derivative of the log-likelihood with respect to \( \lambda_k \) is derived as

\[
\frac{\partial L(D|\lambda)}{\partial \lambda_k} = \sum_{m} \sum_{i,j} f_k(A(m)_i, A(m)_j, O(m)) \left( \frac{\partial P(A(m)|f_k)}{\partial \lambda_k} \right)
\]

\[
- \sum_{m} \sum_{i,j} p(A(m)_j, A(m)_j|\lambda_k) f_k(A(m)_i, A(m)_j, O(m)),
\]

where edge \( (A_i, A_j) \) can be either in \( E_c \) or \( E_t \).

The former part of the partial derivative is easy to compute, while the latter part can be difficult. The marginal probability for the latter part can be computed by using loopy belief propagation introduced in the previous subsection. As solving the equation \( \frac{\partial L(D|\lambda)}{\partial \lambda_k} = 0 \) does not yield a closed-form solution, \( \lambda \) may be updated iteratively using some optimization techniques such as iterative scaling, gradient descent, conjugate gradient or BFGS (Wallach & Weiss 2003) to find the weights that maximize the log-likelihood.

Previous research has shown that L-BFGS works well in such an optimization problem for CRFs learning and many current CRFs packages implement this method. L-BFGS is a variant of the Newton’s method, which is a second order optimization method. It is computationally expensive to calculate the second order derivative, Hessian matrix, Newton’s method. BFGS provides an iterative way to approximate Hessian matrix by updating the matrix in the previous step. L-BFGS implicitly updates the Hessian matrix by memorizing the update progress and gradient information in the previous \( m \) steps. As a result, L-BFGS can reduce the amount of memory and computation needs.

In practice, L-BFGS requests the partial derivative as well as the log-likelihood at each iteration. So far, we have explained how to compute partial derivative. As to the log-likelihood, computing the normalization constant directly is infeasible, so we use Bethe free energy (Yedidia, Freeman, & Weiss 2003) to approximate the normalization constant.

In order to reduce over-fitting, we define a zero mean Gaussian prior \( P(\lambda_k) = \exp(-\lambda_k^2/2\sigma^2) \) with variance \( \sigma^2 \) for each parameter \( \lambda_k \) so that we maximize the penalized log-likelihood \( L(D|\lambda) = L(D|\lambda) + \sum_k \log P(\lambda_k) \). As a result, the partial derivative becomes

\[
\frac{\partial L(D|\lambda)}{\partial \lambda_k} = \frac{\partial L(D|\lambda)}{\partial \lambda_k} - \frac{\lambda_k}{\sigma^2}.
\]

**Experiments**

We have introduced the FCRFs model for multiple concurrent activities recognition. Let us find out how it works for the House data set. In this section, we describe the experimental design as well as the experimental result for the evaluation of our model.

**Experimental Design**

We extracted our experimental data from the MIT House data set, which is freely available for academic research. The data set was recorded on Friday March 4, 2005 from 9 AM to 1 PM with a volunteer performing a set of common household activities. The data set recorded the information from a variety of digital sensors such as switch sensors, light sensors, and current sensors, etc. The data set was then manually labelled with ground truths of multiple activities and location information. Unfortunately, with only four hours of recorded activities in the released data set, the sensor data were too sparse to be useful for effective training. As the House data are still being collected, cleaned, and labelled, so more comprehensive data set may be forthcoming. Meanwhile, we decide to utilize the location data as our primary observations. The location information in the current House data set is manually annotated and assumed to be quite accurate. Nevertheless, our system should perform in the same way should such data come from some (room-level) indoor location tracking system.

Now, let’s explain the data used in our experiments in more detail. The annotation of the data recorded from 9 AM to 10 AM is somewhat unorganized, so we decided to use only the data set recorded from 10 AM to 1 PM for a total of 10800-second. We labeled the activities for every second and divide the data set into 18 parts, each containing 10 minutes of data. This way, we can do 6-fold cross-validation for the evaluation. For simplicity, we clustered the original 89 activities into 6 classes of activities, including cleaning, laundry, dishwashing, meal preparation, personal and information/leisure. Because the 6 classes of activities may overlap with each other, the property of multitasking is suitable for our multiple concurrent activities recognition. In addition, there are 9 mutually exclusive locations that may be observed, including living room, dining area, kitchen, office, bedroom, hallway, bathroom, powder room and outside.

Our FCRFs model is constructed in the following way. First, for each activity class, we build one hidden variable node which contains a Boolean state representing either happening or not-happening. And then we let all the hidden variable nodes in the same time slice to be fully connected, assuming that every activity has cotemporal relationship with each other. To represent the temporal relationship, we connect the two hidden variable nodes across two time slices for the same activity class. In addition, we connect
every hidden variable node with one observed variable node which represents the states of location.

The FCRFs model defined are used to do learning and inference for multiple concurrent activities recognition in our experiments. The variance of the Gaussian prior is 9. To evaluate the importance of co-temporal relationship between activities, we constructed 6 linear chain CRFs (LCRFs) models as for comparison. Each LCRFs model recognizes only one activity class at a time.

Performance Evaluation

We want to test if the LCRFs model and the FCRFs model can correctly predict the activity sequences in the testing data set. There are 6 activities labels at each time stamp. The value of each activity label can be either positive (for happening) or negative (for not-happening). In each part of the testing data set, there are 600 seconds and total 3600 activities labels. A label is considered to be True Positive (TP) if the model correctly predicts its value to be positive. Similarly, a model is True Negative (TN) if the model correctly predicts the value as negative. On the contrary, the label is considered to be False Positive (FP) if the model wrongly guesses the value as positive and False Negative (FN) if the model wrongly guesses the value as negative. To evaluate our performance, we calculate the recall, precision and F-score for these two models.

The results are summarized in Table 1, which compares the recall, precision and F-score for the testing data set.

<table>
<thead>
<tr>
<th>Data set</th>
<th>Recall</th>
<th>Precision</th>
<th>F-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCRF(%)</td>
<td>48.2</td>
<td>51.5</td>
<td>49.8</td>
</tr>
<tr>
<td>LCRF(%)</td>
<td>34.3</td>
<td>53.3</td>
<td>41.7</td>
</tr>
</tbody>
</table>

Table 1: Performance Comparison of LCRFs and FCRFs.

As we can see, the FCRFs model outperforms the LCRFs models. Even though the location information may be ambiguous for the recognition of activity, the consideration for co-temporal relationship in FCRFs complements this deficiency. As a result, the proposed FCRFs model improves the F-score up to 8%. This experimental result provides us the conclusion that it is helpful to utilize the co-temporal relationship for activity recognition.

Conclusion and Future Work

This paper proposes the FCRFs model for joint recognition of multiple concurrent activities. We designed the experiments based on the MIT House_n data set and compared our FCRFs model with the LCRFs model. The initial experiment showed that FCRFs improve the F-score for up to 8% for recognition of multiple concurrent activities in daily living.

The current experiment presents just an initial step towards concurrent activity recognition. As we mentioned earlier, using a single sensor such as location may not be sufficient for disambiguating among the activities. To further improve the recognition accuracy, data from multiple heterogeneous sensors must be combined and taken into consideration in any practical activity recognition system.

In addition, we may extend our activity model to represent long-range temporal dependency as well as the relationship among different activities across time slices. In our current implementation, both learning and inference are performed off-line. To deploy activity inference in real-world context-aware applications, we will need to develop online inference such as the CRF-filter algorithm proposed in (Limketkai, Liao, & Fox 2007).

References


