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 recognition accuracy up to 10% in the presence of multiple concurrent activities.

1. 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.* 2006) 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 is potentially a good benchmark for research on activity recognition.

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.

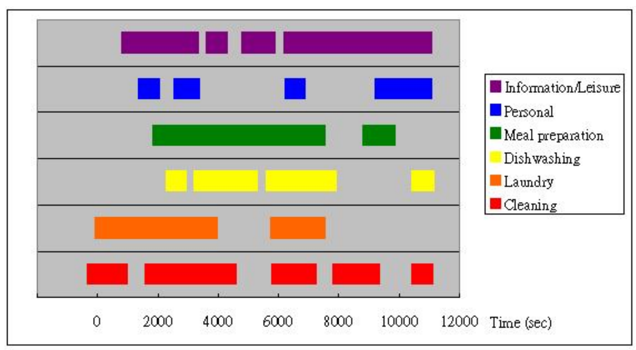

Figure 1: Annotated activities in the House_n data set from 10 AM to 1 PM.

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, & Pereira 2001) provide a powerful probabilistic framework for labelling and segmenting structured data. By defining a conditional probability distribution over label sequences given a particular observation sequence, CRFs relax the *Markov 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. In section 2, we introduce the CRFs model. Section 3 presents FCRFs for multiple concurrent activities recognition including the model definition, the inference algorithm, and the learning method. Our experiment for performance evaluation of this model is described in section 4, followed by the conclusion and future work.

2. Conditional Random Fields

CRFs are undirected graphical models conditioned on observation sequences which have been successfully applied to sequence labelling problems such as bioinfomatics (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.

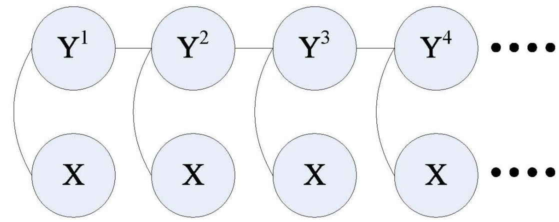

Figure 2: An LCRF example for activity recognition.

Figure 2 shows an example of applying linear chain CRFs (LCRFs) to activity recognition. We can represent the states of activities as a time sequence. Thus, we have a set of hidden variable nodes $Y = \{Y^1, Y^2, Y^3 \ldots\}$ standing for the activity sequence. Given the observation sequence X, the dependency can be defined by the feature functions upon the cliques. Several researchers have applied CRFs to activity recognition (Chieu, Lee, & Kaelbling 2006; Sminchisescu, Kanaujia, & Metaxas 2006; Liao, Fox, & Kautz 2007) , but they did not address the issue of multiple concurrent activities.

3. Factorial CRFs Model

As was discussed in the previous section, recognition of mutually exclusive activities can be modelled with CRFs. To recognize multiple concurrent activities, it is possible to construct a unique model for each individual activity. However, this approach ignores the relationship between different activities. For example, the House_n data set (Intille *et al.* 2006) shows that the participant often used the phone and the computer concurrently in the experiment. In addition, a person typically cannot sleep and watch TV at the same time.

To take the co-temporal relationship between multiple activities into account, we may treat each combination of multiple activities as a new activity. However, the model complexity of this approach grows exponentially with the number of activities to be recognized. As a result, inference and learning may become computationally intractable when the number of activities is large.

Factorial CRFs (FCRFs) (Sutton & McCallum 2004), which are like Factorial HMMs (FHMMs) (Ghahramani & Jordan 1997), suggest a structure of distributed states to avoid the exponential complexity problem. In addition, FCRFs can model the dependency between multiple concurrent activities by introducing co-temporal connections. In section 3.1, we provide the formal definition of the FCRFs model. Sections 3.2 and 3.3 introduce the inference and learning algorithms for FCRFs.

3.1 Model Representation

Let Y_i^t 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 $Y = \{Y^1, Y^2 \dots Y^T\}$ be a sequence of vector Y^t. We can define the observation sequence $X = \{X^1, X^2 \dots X^T\}$ 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.

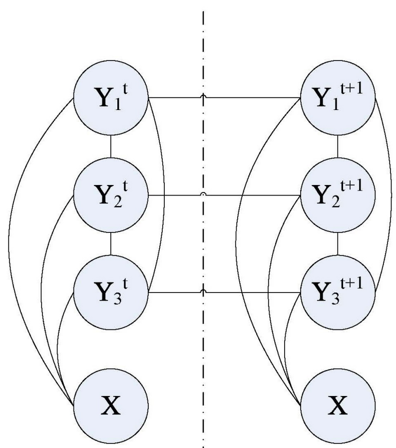

Figure 3: An FCRF example of three concurrent activities.

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

$$p(Y|X) = \frac{1}{Z(X)} \left(\prod_{t=1}^{T} \prod_{i,j}^{N} \Phi_c(Y_i^t, Y_j^t) \right)$$
$$\left(\prod_{t=1}^{T-1} \prod_{i}^{N} \Phi_t(Y_i^t, Y_i^{t+1}) \right) \left(\prod_{t=1}^{T} \prod_{i}^{N} \Psi_t(Y_i^t) \right)$$

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

$$Z(X) = \sum_{Y} \left(\prod_{t=1}^{T} \prod_{i,j}^{N} \Phi_c(Y_i^t, Y_j^t) \right)$$
$$\left(\prod_{t=1}^{T-1} \prod_{i}^{N} \Phi_t(Y_i^t, Y_i^{t+1}) \right) \left(\prod_{t=1}^{T} \prod_{i}^{N} \Psi_t(Y_i^t) \right)$$

The potential functions Φ_c, Φ_t and Ψ are then defined by a set of feature functions $f = \{f_1, f_2 \dots f_K\}$ and their corresponding weights $\lambda = \{\lambda_1, \lambda_2 \dots \lambda_K\}$ such that

$$\Phi_c(Y_i^t, Y_j^t) = exp\left(\sum_k \lambda_k f_k(Y_i^t, Y_j^t, X) \right)$$
$$\Phi_t(Y_i^t, Y_i^{t+1}) = exp\left(\sum_k \lambda_k f_k(Y_i^t, Y_i^{t+1}, X) \right)$$
$$\Psi(Y_i^t) = exp\left(\sum_k \lambda_k f_k(Y_i^t, Y_i^t, X) \right)$$

Now, we have formally defined the FCRFs.

3.2 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. The marginal probability is needed in the learning algorithm to be introduced later. The other task is performing MAP inference to infer the most possible sequence of activities states.

There are many inference algorithms for CRFs, including forward-backward algorithm, mean field free energy, junction tree, and loopy belief propagation (LBP). LBP is one of the most popular methods for performing inference in CRFs. 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).

3.2.1 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_n except A_j.

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

where $\Psi(A_i)$ is the local potential, $\Phi(A_i, A_j)$ is the pair-wise potential, and κ 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 set a limit on the maximum number of iterations. Although 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

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

where

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

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

3.2.2 MAP Inference To do MAP inference, the summation part of the message update rule in sum-product algorithm is replaced by maximization.

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

where κ is the normalization constant.

After the LBP converges, the MAP probability of node A_i is defined as

$$P(A_i) = \Psi(A_i) \prod_{j \neq i} m_{ji}(A_i)$$

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.

3.3 Learning Algorithm

The purpose of learning is to determine the weight λ_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 respect to λ_k is derived as

$$\frac{\partial L(D|\lambda)}{\partial \lambda_k} = \sum_m \sum_{i,j} f_k(A(m)_i, A(m)_j, O(m))$$

$$- \sum_m \sum_{i,j} p(A(m)_i, A(m)_j|\lambda) f_k(A(m)_i, A(m)_j, O(m))$$

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 is more difficult. The marginal probability for the latter part can be computed by using loopy belief propagation which we have introduced in the previous subsection. Since to solve the $\partial L(D|\lambda)/\partial \lambda_k = 0$ does not yield a close form solution of λ, we may use some optimization techniques such as iterative scaling, gradient descent, conjugate gradient or BFGS (Wallach 2003; Sha & Pereira 2003) to iteratively update λ in order to find the weights that maximize the log-likelihood.

Previous research has shown that L-BFGS works well in such a optimization problem for CRFs learning and many current CRFs packages implement this method . L-BFGS is a variant of the Newton method, which is a second order optimization method. In Newton method, the computation for the second order derivative, Hessian matrix, is computationally expensive. BFGS provides an iterative way to approximate the Hessian matrix by updating the Hessian matrix in the previous step. L-BFGS implicitly updates the Hessian matrix by memorizing the update progress and gradient information in previous m steps. Therefore, L-BFGS 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 σ^2 for each parameter λ_k so that we maximize the penalized log-likelihood $L(\lambda|D) = L(D|\lambda) + \sum_k logP(\lambda_k)$. As a result, the partial derivative becomes

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

4. Experiments

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

4.1 Experimental Design

We extracted our experimental data from the MIT House_n dataset, which is freely available for academic research. The dataset was recorded on Friday March 4, 2005 from 9 AM to 1 PM with a volunteer performing a set of common household activities. The dataset recorded the information from a variety of digital sensors such as switch sensors, light sensors, and current sensors, etc. The dataset was then manually labelled with ground truths of multiple activities and location information. Unfortunately, with only four hours of recorded activities in the released dataset, the sensor data were too sparse to be useful for effective training. As the House_n data are still being collected, cleaned, and labelled, so more comprehensive dataset may be forthcoming. Meanwhile, we decide to utilize the location data as our primary observations. The location information in the current House_n dataset 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 experiment in more detail. The annotation data recorded from 9 AM to 10 AM is a little unorganized, so we decide to use the dataset recorded from 10 AM to 1 PM. We label the activities every one second and separate the total 10800-second dataset into 18 parts, 10 minutes for each. To do cross-validation, we randomly select 15 parts from them as the training dataset and 3 parts as the testing dataset. For convenience, we cluster 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 considered to be observations, including *living room*, *dining area*, *kitchen*, *office*, *bedroom*, *hallway*, *bathroom*, *powder room* and *outside*.

We construct our FCRFs model in the following way. First, for each activity class, we build one hidden variable node which contains a boolean states representing *happening* or *not-happening*. And then we let all the hidden variable nodes in the same time slice to be fully connected, which means they have co-temporal relationship. 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.

In our experiments, we use our FCRFs model to do learning and inference for multiple concurrent activities recognition. To evaluate the importance of co-temporal relationship between activities, we construct 6 linear chain CRFs (LCRFs) models as a comparison. Each LCRFs model recognizes only one activity class at a time.

4.2 Performance Evaluation

We want to test if the LCRFs model and the FCRFs model can correctly predict the activity sequences by using the 3 parts of the testing dataset. There are 6 activities labels at each time stamp whose values can be either positive or negative where positive stands for *happening* and negative stands for *not-happening*. In each part of the testing dataset, there are 600 seconds and total 3600 activities labels. A label is considered to be *True Positive* (TP) if the model correctly predicts the value as positive and *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 accuracy for these two models given each part of the testing dataset. The recall, precision and accuracy are defined as follow.

$$recall = TP/(TP + FN)$$

$$precision = TP/(TP + FP)$$

$$accuracy = (TP + TN)/(TP + FP + TN + FN)$$

The results are summarized in Table 1, which compares the recall, precision and accuracy for the three parts of the testing data set. Each column is labelled with either D_i^L for LCRFs on the ith part or D_i^F for FCRFs on the ith part.

Table 1: Performance Comparison of LCRFs and FCRFs.

Dataset	D_1^L	D_1^F	D_2^L	D_2^F	D_3^L	D_3^F
Recall(%)	19.3	45.9	45.6	60.7	95.1	84.8
Precision(%)	39.8	61.8	97.3	94.2	94.5	69.3
Accuracy(%)	60.6	70.4	74.6	80.3	98.1	90.2

As we can see, the FCRFs model outperforms the LCRFs models in most of the cases. The only exception is the third part of the testing dataset. Notice that both models achieve the accuracy higher than 90% and the accuracy of LCRFs are even higher than 98%. It means that the location information itself is almost sufficient to recognize the activities. Therefore, taking co-temporal connections into account in this case may reduce the accuracy a little bit.

In other cases, especially in the first one, because the participant goes around in the house, the location information is inadequate for the recognition of activity. Therefore, the consideration for co-temporal relationship in FCRFs complements this deficiency. As a result, the FCRFs model improves the accuracy up to 10%. This experiment result provides us the conclusion that it is pretty helpful to utilize the co-temporal relationship for activity recognition.

5. 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 using FCRFs, the improvement in accuracy of activity recognition varies from 6% to 10%. We may therefore conclude that FCRFs can be effectively applied to joint recognition of multiple concurrent activities in daily living.

The current experiment presents just an initiative step towards concurrent activity recognition. As we mentioned, 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 the 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

Chieu, H. L.; Lee, W. S.; and Kaelbling, L. P. 2006. Activity recognition from physiological data using conditional random fields. Technical report, SMA Symposium.

Ghahramani, Z., and Jordan, M. I. 1997. Factorial hidden markov models. *Machine Learning* 29(2-3):245–273.

Intille, S. S.; Larson, K.; Tapia, E. M.; Beaudin, J. S.; Kaushik, P.; Nawyn, J.; and Rockinson, R. 2006. Using a live-in laboratory for ubiquitous computing research. In Fishkin, K. P.; Schiele, B.; Nixon, P.; and Quigley, A., eds., *PERVASIVE 2006*, volume l, 349–365.

Lafferty, J.; McCallum, A.; and Pereira, F. 2001. Conditional random fields: Probabilistic models. for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Confernence on Machine Learning (ICML-01)*.

Liao, L.; Fox, D.; and Kautz, H. 2007. Extracting places and activities from gps traces using hierarchical conditional random fields. *International Journal of Robotics Research* 26:119–134.

Limketkai, B.; Liao, L.; and Fox, D. 2007. Crf-filters: Discriminative particle filters for sequential state estimation. In *2007 IEEE International Conference on Robotics and Automation*.

Liu, Y.; Carbonell, J. G.; P., W.; and V., G. 2005. Segmentation conditional random fields (scrfs): A new approach for protein fold recognition. In *Research in Computational Molecular Biology, 9th Annual International Conference*.

Sarawagi, S., and Cohen, W. W. 2005. Semi-markov conditional random fields for information extraction. In *Advances in Neural Information Processing Systems 17*.

Sato, K., and Y., S. 2005. Rna secondary structural alignment with conditional random fields. In *ECCB/JBI (Supplement of Bioinformatics)*.

Sha, F., and Pereira, F. 2003. Shallow parsing with conditional random fields. In *Proceedings of the 2003 Human Language Technology Conference and North American Chapter of the Association for Computational Linguistics*.

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

Sutton, C. A. Rohanimanesh, K., and McCallum, A. 2004. Dynamic conditional random fields: factorized probabilistic models for labeling and segmenting sequence data. In *Proceedings of the Twenty First International Confernence on Machine Learning (ICML-04)*.

Vishwanathan, S. V. N.; Schraudolph, N. N.; Schmidt, M. W.; and Murphy, K. 2006. Accelerated training of conditional random. fields with stochastic meta-descent. In *Proceedings of the Twenty Third International Confernence on Machine Learning (ICML-06)*.

Wallach, H. M. 2003. Efficient training of conditional random fields. In *Proceedings of the 6th Annual CLUK Research Colloquium*. University of Edinburgh.

Yedidia, J. S.; Freeman, W. T.; and Weiss, Y. 2003. *Understanding Belief Propagation and its Generalizations*. Kaufmann, M. chapter 8, 236–239.