Hierarchical Semi-Markov Conditional Random Fields for Recursive Sequential Data

Tran The Truyen [†], Dinh Q. Phung [†], Hung H. Bui [‡][*], and Svetha Venkatesh [†]
[†]Department of Computing, Curtin University of Technology
GPO Box U1987 Perth, WA 6845, Australia
thetruyen.tran@postgrad.curtin.edu.au
{D.Phung,S.Venkatesh}@curtin.edu.au

[‡]Artificial Intelligence Center, SRI International
333 Ravenswood Ave, Menlo Park, CA 94025, USA
bui@ai.sri.com

Abstract

Inspired by the hierarchical hidden Markov model (HHMM), we present the *hierarchical semi-Markov conditional random field* (HSCRF), a generalisation of embedded undirected Markov chains to model complex hierarchical, nested Markov processes. It is parameterised in a discriminative framework and has polynomial time algorithms for learning and inference. Importantly, we develop efficient algorithms for learning and constrained inference in a partially supervised setting, which is an important issue in practice where labels can only be obtained sparsely. We demonstrate the HSCRF in two applications: (i) recognising human activities of daily living (ADLs) from indoor surveillance cameras, and (ii) noun-phrase chunking. We show that the HSCRF is capable of learning rich hierarchical models with reasonable accuracy in both fully and partially observed data cases.

1 Introduction

Modelling hierarchical aspects in complex stochastic processes is an important research issue in many application domains including computer vision, text information extraction, computational linguistics and bioinformatics. For example, in a syntactic parsing task known as noun-phrase chunking, noun-phrases (NPs) and part-of-speech (POS) tags are two layers of semantics associated with words in the sentence. Previous approaches usually first tag the POS and then feed these tags as input to the chunker. The POS tagger does not take into account information about the NPs. This may not be optimal, as a noun-phrase is often very informative for inferring the POS tags belonging to the phrase. Thus, it is more desirable to *jointly* model and infer both the NPs and the POS tags at the same time.

Many graphical models have been proposed to address this challenge, typically extending the flat hidden Markov models (e.g., hierarchical HMM (HHMM) [2], DBN [6]). These models are, however, *generative* in that they are forced to model the joint distribution $\Pr(x, z)$ for both the observation z and the label x. An attractive alternative is to model the distribution $\Pr(x|z)$ directly, avoiding the modelling of z. This line of research has recently attracted much interest, and one of the significant results was the introduction of the *conditional random field* (CRF) [4]. Work in

[*]SRI International is supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. FA8750-07-D-0185/0004. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA, or the Air Force Research Laboratory (AFRL).

CRFs was originally limited to flat structures for efficient inference, and subsequently extended to hierarchical structures, such as the dynamic CRFs (DCRF) [10], and hierarchical CRFs [5]. These models assume predefined structures; therefore, they are not flexible enough to adapt to many real-world datasets. For example, in the noun-phrase chunking problem, no prior hierarchical structures are known. Rather, if such a structure exists, it can be discovered only *after* the model has been successfully built and learned.

In addition, most discriminative structured models are trained in a completely supervised fashion using fully labelled data, and limited research has been devoted to dealing with the *partially labelled* data (e.g., [3, 12]). In several domains, it is possible to obtain some labels with minimal effort. Such information can be used either for training or for decoding. We term the process of learning with partial labels *partial supervision*, and the process of inference with partial labels *constrained inference*. Both processes require the construction of appropriate constrained inference algorithms.

We are motivated by the HHMM [2], a directed, generative model parameterised as a standard Bayesian network. To address the above issues, we propose the *Hierarchical Semi-Markov Conditional Random Field* (HSCRF), which is a recursive, undirected graphical model that generalises the undirected Markov chains and allows hierarchical decomposition. The HSCRF is parameterised as a standard log-linear model, and thus can naturally incorporate discriminative modelling. For example, the noun-phrase chunking problem can be modeled as a two-level HSCRF, where the top level represents the NP process and the bottom level the POS process. The two processes are conditioned on the sequence of words in the sentence. Each NP generally spans one or more words, each of which has a POS tag. Rich contextual information such as starting and ending of the phrase, the phrase length, and the distribution of words falling inside the phrase can be effectively encoded. At the same time, like the HHMM, exact inference in the HSCRFs can be performed in polynomial time in a manner similar to the Asymmetric Inside-Outside algorithm (AIO) [1].

We demonstrate the effectiveness of HSCRFs in two applications: (i) segmenting and labelling activities of daily living (ADLs) in an indoor environment and (ii) jointly modelling noun-phrases and parts-of-speech in shallow parsing. Our experimental results in the first application show that the HSCRFs are capable of learning rich, hierarchical activities with good accuracy and exhibit better performance when compared to DCRFs and flat CRFs. Results for the partially observable case also demonstrate that significant reduction of training labels still results in models that perform reasonably well. We also show that observing a small amount of labels can significantly increase the accuracy during decoding. In noun-phrase chunking, the HSCRFs can achieve higher accuracy than standard CRF-based techniques and the recent DCRFs. Our contributions from this paper are thus: i) the introduction of the novel and Hierarchical Semi-Markov Conditional Random Field to model nested Markovian processes in a discriminative framework, ii) the development of an efficient generalised Asymmetric Inside-Outside (AIO) algorithm for partially supervised learning and constrained inference, and iii) the applications of the proposed HSCRFs in human activities recognition, and in shallow parsing of natural language.

Due to space constraints, in this paper we present only main ideas and empirical evaluations. Complete details and extensions can be found in the technical report [11]. The next section introduces necessary notations and provides a model description for the HSCRF, followed by the discussion on learning and inference for fully and partially data cases in Sections 3 and 4, respectively. Applications for recognition of activities and natural language parsing are presented in Section 5. Finally, discussions on the implications of the HSCRF and conclusions are given in Section 6.

2 Model Definition and Parameterisation

2.1 Hierarchical Semi-Markov Conditional Random Fields

Consider a hierarchically nested Markov process with D levels where, by convention, the top level is the dummy root level that generates all subsequent Markov chains. Then, as in the generative process of the hierarchical HMMs [2], the parent state embeds a child Markov chain whose states may in turn contain grandchild Markov chains. The relation among these nested Markov chains is defined via the *model topology*, which is a state hierarchy of depth D. It specifies a set of states S^d at each level d, i.e., $S^d = \{1...|S^d|\}$, where $|S^d|$ is the number of states at level d and $1 \leq d \leq D$. For each state $s^d \in S^d$ where $d \neq D$, the model also defines a set of children associated with it at the next level $ch(s^d) \subset S^{d+1}$, and thus conversely, each child s^{d+1} is associated with a set of parental states at the upper level $pa(s^{d+1}) \subset S^d$. Unlike the original HHMMs proposed in [2] where tree structure is explicitly enforced on the state hierarchy, the HSCRFs allow arbitrary sharing of children among parental states as addressed in [1]. This topology generalization requires a smaller number of sub-states when D is large, and thus leads to fewer parameters and possibly less training data and time complexity [1].

To provide an intuition, the temporal evolution can be informally described as follows. Start with the root node at the top level; as soon as a new state is created at level $d \neq D$, it *initialises* a child state at level $d+1$. The initialisation continues downward until reaching the bottom level[1]. This child process at level $d+1$ continues its execution recursively until it *terminates*, and when it does, the control of execution returns to its parent at the upper level d. At this point, the parent makes a decision either to *transit* to a new state at the same level or return the control to the grandparent at the upper level $d-1$.

The key intuition for this hierarchical nesting process is that the life span of a child process is a sub-segment in the life span of its parent. To be more precise, consider the case in which a parent process $s^d_{i:j}$ at level d starts a new state[2] at time i and persists until time j. At time i, the parent initialises a child state s^{d+1}_i that continues until it ends at time $k < j$, at which the child state transits to a new child state s^{d+1}_{k+1}. The child process exits at time j, at which the control from the child level is returned to the parent $s^d_{i:j}$. Upon receiving the control, the parent state $s^d_{i:j}$ may transit to a new parent state $s^d_{j+1:l}$, or end at j and return the control to the grandparent at level $d-1$.

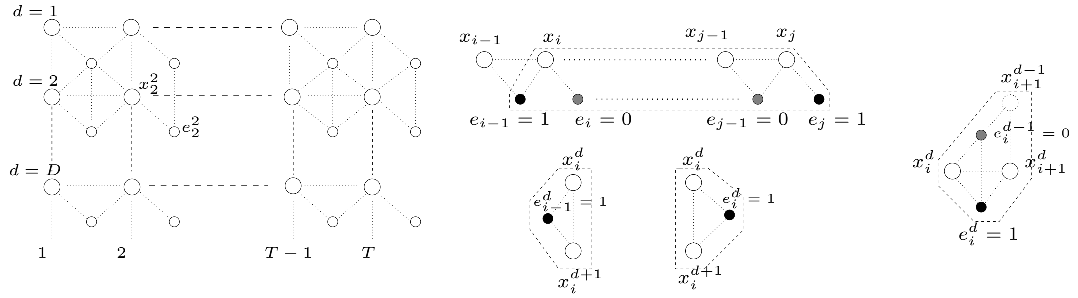


Figure 1: Graphical presentation for HSCRFs (leftmost). Graph structures for state-persistence (middle-top), initialisation and ending (middle-bottom), and state-transition (rightmost).

The HSCRF, which is a multi-level temporal graphical model of length T with D levels, can be described formally as follows (Fig. 1). It starts from the root level indexed as 1, runs for T time slices and at each time slice a hierarchy of D states is generated. At each level d and time index i, there is a node representing a state variable $x^d_i \in S^d = \{1, 2, ..., |S^d|\}$. Associated with each x^d_i is an ending indicator e^d_i that can be either 1 or 0 to signify whether the state x^d_i terminates or continues its execution to the next time slice. The nesting nature of the HSCRFs is formally realised by imposing the specific constraints on the value assignment of ending indicators:

- The root state persists during the course of evolution, i.e., $e^1_{1:T-1} = 0$, $e^1_T = 1$, and all states end at the last time-slice, i.e., $e^{1:D}_T = 1$.
- When a state finishes, all its descendants must also finish, i.e., $e^d_i = 1$ implies $e^{d+1:D}_i = 1$; when a state persists, all its ancestors must also persist, i.e., $e^d_i = 0$ implies $e^{1:d-1}_i = 0$.
- When a state transits, its parent must remain unchanged, i.e., $e^d_i = 1$, $e^{d-1}_i = 0$, and states at the bottom level terminate at every single slice, i.e., $e^D_i = 1$ for all $i \in [1, T]$.

Thus, specific value assignments of ending indicators provide *contexts* that realise the evolution of the model states in both hierarchical (vertical) and temporal (horizontal) directions. Each context at

[1] In HHMMs, the bottom level is also called *production* level, in which the states emit observational symbols. In HSCRFs, this generative process is not assumed.

[2] Our notation $s^d_{i:j}$ is to denote the set of variables from time i to j at level d, i.e., $s^d_{i:j} = \{s^d_i, s^d_{i+1}, \ldots, s^d_j\}$.

a level and associated state variables form a *contextual clique*, and here we identify four contextual clique types (cf. Fig. 1):

- *State-persistence* : This corresponds to the life time of a state at a given level. Specifically, given a context $c = (e^d_{i-1:j} = (1, 0, .., 0, 1))$, then $\sigma^{persist,d}_{i:j} = (x^d_{i:j}, c)$, is a contextual clique that specifies the life span $[i, j]$ of any state $s = x^d_{i:j}$.
- *State-transition* : This corresponds to a state at level $d \in [2, D]$ at time i transiting to a new state. Specifically, given a context $c = (e^{d-1}_i = 0, e^d_i = 1)$ then $\sigma^{transit,d}_i = (x^{d-1}_{i+1}, x^d_{i:i+1}, c)$ is a contextual clique that specifies the transition of x^d_i to x^d_{i+1} at time i under the same parent x^{d-1}_{i+1}.
- *State-initialisation* : This corresponds to a state at level $d \in [1, D-1]$ initialising a new child state at level $d+1$ at time i. Specifically, given a context $c = (e^d_{i-1} = 1)$, then $\sigma^{init,d}_i = (x^d_i, x^{d+1}_i, c)$ is a contextual clique that specifies the initialisation at time i from the parent x^d_i to the first child x^{d+1}_i.
- *State-exiting* : This corresponds to a state at level $d \in [1, D-1]$ to end at time i. Specifically, given a context $c = (e^d_i = 1)$, then $\sigma^{exit,d}_i = (x^d_i, x^{d+1}_i, c)$ is a contextual clique that specifies the ending of x^d_i at time i with the last child x^{d+1}_i.

In the HSCRF, we are interested in the *conditional* setting, in which the entire state and ending variables $(x^{1:D}_{1:T}, e^{1:D}_{1:T})$ are conditioned on an observational sequence z. For example, in computational linguistics, the observation is often the sequence of words, and the state variables might be the POS tags and the phrases.

To capture the correlation between variables and such conditioning, we define a nonnegative potential function $\phi(\sigma, z)$ over each contextual clique σ. Figure 2 shows the notations for potentials that correspond to the four contextual clique types we have identified above. Details of potential specification are described in Section 2.2.

State persistence potential	$R^{d,s,z}_{i:j} = \phi(\sigma^{persist,d}_{i:j}, z)$ where $s = x^d_{i:j}$.
State transition potential	$A^{d,s,z}_{u,v,i} = \phi(\sigma^{transit,d}_i, z)$ where $s = x^{d-1}_{i+1}$ and $u = x^d_i, v = x^d_{i+1}$.
State initialization potential	$\pi^{d,s,z}_{u,i} = \phi(\sigma^{init,d}_i, z)$ where $s = x^d_i, u = x^{d+1}_i$.
State ending potential	$E^{d,s,z}_{u,i} = \phi(\sigma^{exit,d}_i, z)$ where $s = x^d_i, u = x^{d+1}_i$.

Figure 2: Shorthand for contextual clique potentials.

Let $\mathcal{V} = (x^{1:D}_{1:T}, e^{1:D}_{1:T})$ denote the set of all variables and let $\tau^d = \{i_k\}^m_{k=1}$ denote the set of all time indices where $e^d_{i_k} = 1$. A *configuration* ζ of the model is a complete assignment of all the states and ending indicators $(x^{1:D}_{1:i}, e^{1:D}_{1:T})$ that satisfies the set of hierarchical constraints described earlier in this section. The joint potential defined for each configuration is the product of all contextual clique potentials over all ending time indexes $i \in [1, T]$ and all semantic levels $d \in [1, D]$:

$$\Phi(\zeta, z) = \prod_d \left\{ \left[\prod_{(i_k, i_{k+1}) \in \tau^d} R^{d,s,z}_{i_k+1:i_{k+1}} \right] \left[\prod_{i_k \in \tau^d, i_k \notin \tau^{d-1}} A^{d,s,z}_{u,v,i_k} \right] \left[\prod_{i_k \in \tau^d} \pi^{d,s,z}_{u,i_k+1} \right] \left[\prod_{i_k \in \tau^d} E^{d,s,z}_{u,i_k} \right] \right\}$$

The conditional distribution is given as

$$\Pr(\zeta|z) = \frac{1}{Z(z)} \Phi(\zeta, z) \tag{1}$$

where $Z(z) = \sum_\zeta \Phi(\zeta, z)$ is the partition function for normalisation.

2.2 Log-linear Parameterisation

In our HSCRF setting, there is a feature vector $\mathbf{f}^d_\sigma(\sigma, z)$ associated with each type of contextual clique σ, in that $\phi(\sigma^d, z) = \exp\left\{\theta^d_\sigma \bullet \mathbf{f}^d_\sigma(\sigma, z)\right\}$. where $a \bullet b$ denotes the inner product of two vectors a and b. Thus, the features are active only in the context in which the corresponding contextual cliques appear. For the state-persistence contextual clique, the features incorporate *state-duration*, start time i and end time j of the state. Other feature types incorporate the time index in which the features are triggered. In what follows, we omit z for clarity, and implicitly use it as part of the partition function Z and the potential $\Phi(.)$.

3 Unconstrained Inference and Fully Supervised Learning

Typical inference tasks in the HSCRF include computing the partition function, MAP assignment and feature expectations. The key insight is the context-specific independence, which is due to hierarchical constraints described in Section 2.1. Let us call the set of variable assignments $\Pi^{d,s}_{i:j} = (x^d_{i:j} = s, e^{d:D}_{i-1} = 1, e^{d:D}_j = 1, e^d_{i:j-1} = 0)$ the *symmetric Markov blanket*. Given $\Pi^{d,s}_{i:j}$, the set of variables inside the blanket is independent of those outside it. A similar relation holds with respect to the *asymmetric Markov blanket*, which includes the set of variable assignments $\Gamma^{d,s}_{i:j}(u) = (x^d_{i:j} = s, x^{d+1}_j = u, e^{d:D}_{i-1} = 1, e^{d+1:D}_j = 1, e^d_{i:j-1} = 0)$. Figure 3 depicts an asymmetric Markov blanket (the covering arrowed line) containing a smaller asymmetric blanket (the left arrowed line) and a symmetric blanket (the double-arrowed line). Denote by $\Delta^{d,s}_{i:j}$ the *sum of products* of all clique potentials falling inside the symmetric Markov blanket $\Pi^{d,s}_{i:j}$. The sum is taken over all possible value assignments of the set of variables inside $\Pi^{d,s}_{i:j}$. In the same manner, let $\alpha^{d,s}_{i:j}(u)$ be the sum of products of all clique potentials falling inside the asymmetric Markov blanket $\Gamma^{d,s}_{i:j}(u)$. Let $\hat{\Delta}^{d,s}_{i:j}$ be a shorthand for $\Delta^{d,s}_{i:j} R^{d,s}_{i:j}$. Using the context-specific independence described above and the decomposition depicted Figure 3, the following *recursions* arise:

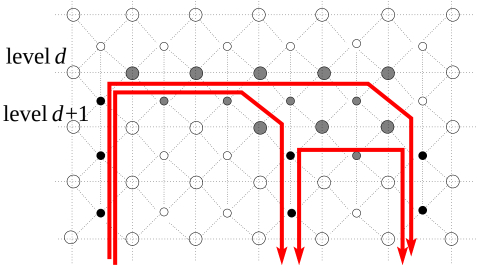


Figure 3: Decomposition with respect to symmetric/asymmetric Markov blankets.

$$\Delta^{d,s}_{i:j} = \sum_{u \in S^{d+1}} \alpha^{d,s}_{i:j}(u) E^{d,s}_{u,j}; \quad \alpha^{d,s}_{i:j}(u) = \sum_{k=i+1}^{j} \sum_{v \in S^{d+1}} \alpha^{d,s}_{i:k-1}(v) \hat{\Delta}^{d+1,u}_{k:j} A^{d+1,s}_{v,u,k-1} + \hat{\Delta}^{d+1,u}_{i:j} \pi^{d+1,s}_{u,i} \qquad (2)$$

As the symmetric Markov blanket $\Pi^{1,s}_{1:T}$ and the set $x^1_{1:T} = s$ covers every state variable, the partition function can be computed as $Z = \sum_{s \in S^1} \hat{\Delta}^{1,s}_{1:T}$.

MAP assignment is essentially the *max-product* problem, which can be solved by turning all summations in (2) into corresponding maximisations.

Parameter estimation in HSCRFs, as in other log-linear models, requires the computation of feature expectations as a part of the log-likelihood gradient (e.g., see [4]). The gradient is then fed into any black-box standard numerical optimisation algorithms. As the feature expectations are rather involved, we intend to omit the details. Rather, we include here as an example the expectation of the state-persistence features

$$\sum_{i \in [1,T]} \sum_{j \in [i,T]} \mathbb{E}[\mathbf{f}^{d,s}_{\sigma^{persist}}(i,j) \delta(\Pi^{d,s}_{i:j} \in \zeta)] = \frac{1}{Z} \sum_{i \in [1,T]} \sum_{j \in [i,T]} \Delta^{d,s}_{i:j} \Lambda^{d,s}_{i:j} R^{d,s}_{i:j} \mathbf{f}^{d,s}_{\sigma^{persist}}(i,j)$$

where $\mathbf{f}^{d,s}_{\sigma^{persist}}(i,j)$ is the state-persistence feature vector for the state $s = x^d_{i:j}$ starting at i and ending at j; $\Lambda^{d,s}_{i:j}$ is the sum of products of all clique potentials falling *outside* the symmetric Markov blanket $\Pi^{d,s}_{i:j}$; and $\delta(\Pi^{d,s}_{i:j} \in \zeta)$ is the indicator function that the Markov blanket $\Pi^{d,s}_{i:j}$ is part of the random configuration ζ.

4 Constrained Inference and Partially Supervised Learning

It may happen that the training data is not completely labelled, possibly due to lack of labelling resources [12]. In this case, the learning algorithm should be robust enough to handle missing labels. On the other hand, during inference, we may partially obtain high quality labels from external sources [3]. This requires the inference algorithm to be responsive to the available labels which may help to improve the performance.

In general, when we make observations, we observe some states and some ending indicators. Let $\tilde{\mathcal{V}} = \{\tilde{x}, \tilde{e}\}$ be the set of observed state and end variables respectively. The procedures to compute the auxiliary variables such as $\Delta^{d,s}_{i:j}$ and $\alpha^{d,s}_{i:j}(u)$ must be modified to address constraints arisen from these observations. For example, computing $\Delta^{d,s}_{i:j}$ assumes $\Pi^{d,s}_{i:j}$, which implies the constraint that the state s at level d starting at i and persisting until terminating at j. Then, if any observations (e.g., there is an $\tilde{x}^d_k \neq s$ for $k \in [i, j]$) are made causing this constraint to be invalid, $\Delta^{d,s}_{i:j}$ will be zero. Therefore, in general, the computation of each auxiliary variable is multiplied by an identity function that enforces the consistency between the observations and the required constraints associated with the computation of that variable.

As an example, we consider the computation of $\Delta^{d,s}_{i:j}$. The sum $\Delta^{d,s}_{i:j}$ is consistent only if all the following conditions are satisfied: (a) if there are observed states at level d within the interval $[i, j]$ they must be s, (b) if there is any observed ending indicator $\tilde{e}^d_{i-1}$, then $\tilde{e}^d_{i-1} = 1$, (c) if the ending indicator $\tilde{e}^d_k$ is observed for some $k \in [i, j-1]$, then $\tilde{e}^d_k = 0$, and (d) if the ending indicator $\tilde{e}^d_j$ is observed, then $\tilde{e}^d_j = 1$. These conditions are captured in the following identity function

$$\mathbb{I}[\Delta^{d,s}_{i:j}] = \delta(\tilde{x}^d_{k\in[i,j]} = s)\delta(\tilde{e}^d_{i-1} = 1)\delta(\tilde{e}^d_{k\in[i:j-1]} = 0)\delta(\tilde{e}^d_j = 1) \qquad (3)$$

When observations are made, the first equation in (2) is thus replaced by

$$\Delta^{d,s}_{i:j} = \mathbb{I}[\Delta^{d,s}_{i:j}]\left(\sum_{u\in S^{d+1}} \alpha^{d,s}_{i:j}(u) E^{d,s}_{u,j}\right) \qquad (4)$$

5 Applications

We describe two applications of the proposed hierarchical semi-Markov CRFs: activity recognition in Section 5.1 and shallow parsing in Section 5.2.

5.1 Recognising Indoor Activities

In this experiment, we evaluate the HSCRFs with a relatively small dataset from the domain of indoor video surveillance. The task is to recognise trajectories and activities, which a person performs in a kitchen, from his noisy locations extracted from video. The data, originally described in [7], has 45 training and 45 test sequences, each of which corresponds to one of three persistent activities: (1) *preparing short-meal*, (2) *having snack* , and (3) *preparing normal-meal*. The persistent activities share some of the 12 sub-trajectories. Each sub-trajectory is a sub-sequence of discrete locations. Thus, naturally, the data has a state hierarchy of depth 3: the dummy root for each location sequence, the persistent activities, and the sub-trajectories. The input observations to the model are simply sequences of discrete locations.

At each level d and time t we count an error if the predicted state is not the same as the ground truth. First, we examine the fully observed case where the HSCRF is compared against the DCRF [10] at both data levels, and against the Sequential CRF (SCRF) [4] at the bottom level. Table 1 (the left half) shows that (a) both the multilevel models significantly outperform the flat model and (b) the HSCRF outperforms the DCRF.

Alg.	$d = 2$	$d = 3$	Alg.	$d = 2$	$d = 3$
HSCRF	100	93.9	PO-HSCRF	80.2	90.4
DCRF	96.5	89.7	PO-SCRF	-	83.5
SCRF	-	82.6	-	-	-

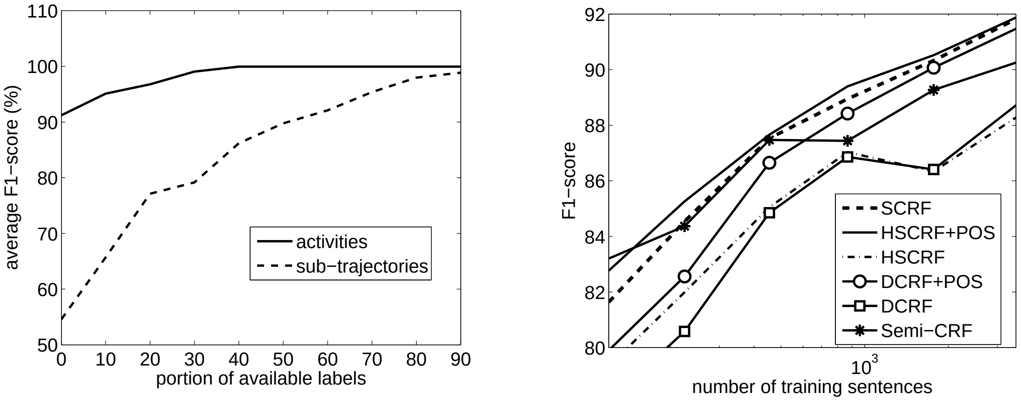
Table 1: Accuracy (%) for fully observed data (left), and partially observed (PO) data (right).

Next, we consider partially supervised learning in which about 50% of start/end times of a state and state labels are observed at the second level. All ending indicators are known at the bottom level. The results are reported in Table 1 (the right half). As can be seen, although only 50% of the state labels and state start/end times are observed, the model learned is still performing well with accuracy of 80.2% and 90.4% at levels 2 and 3, respectively.

We next consider the issue of partially observing labels during decoding and test the effect using degraded learned models. Such degraded models (emulating noisy training data or lack of training time) are extracted from the 10th iteration of the fully observed data case. The labels are provided at random times. Figure 4a shows the decoding accuracy as a function of available state labels. It is interesting to observe that a moderate amount of observed labels (e.g., $20-40$%) causes the accuracy rate to go up considerably.

5.2 POS Tagging and Noun-Phrase Chunking

In this experiment, we apply the HSCRF to the task of noun-phrase chunking. The data is from the CoNLL-2000 shared task [3], in which 8926 English sentences from the Wall Street Journal corpus are used for training and 2012 sentences are for testing. Each word in a preprocessed sentence is given two labels: the part-of-speech (POS) and the noun-phrase (NP). There are 48 POS labels and 3 NP labels (B-NP for beginning of a noun-phrase, I-NP for inside a noun-phrase or O for others). Each noun-phrase generally has more than one word. To reduce the computational burden, we reduce the POS tag-set to five groups: *noun*, *verb*, *adjective*, *adverb*, and *others*. Since in our HSCRFs we do not have to explicitly indicate which node is the beginning of a segment, the NP label set can be reduced further into NP for noun-phrase, and O for anything else.



(a) (b)

Figure 4: (a) Decoding accuracy of indoor activities as a function of available information on label/start/end time. (b) Performance of various models on Conll2000 noun phrase chunking. HSCRF+POS and DCRF+POS mean HSCRF and DCRF with POS given at test time, respectively.

We build an HSCRF topology of three levels, where the root is just a dummy node, the second level has two NP states, and the bottom level has five POS states. For comparison, we implement a DCRF, an SCRF, and a semi-Markov CRF (Semi-CRF) [8]. The DCRF has grid structure of depth 2, one for modelling the NP process and another for the POS process. Since the state spaces are relatively small, we are able to run exact inference in the DCRF by collapsing both the NP and POS state spaces to a combined state space of size $3 \times 5 = 15$. The SCRF and Semi-CRF model only the NP process, taking the POS tags and words as input.

We extract raw features from the text in a similar way to [10]. The features for SCRF and the Semi-CRF also include the POS tags. Words with fewer than three occurrences are not used. This reduces the vocabulary and the feature size significantly. We also make use of bi-grams with similar selection criteria. Furthermore, we use the contextual window of 5 instead of 7 as in [10]. This setting gives rise to about 32K raw features. The model feature is factorised as $f(x_c, z) = \mathbb{I}(x_c)g_c(z)$, where $\mathbb{I}(x_c)$ is a binary function on the assignment of the clique variables x_c, and $g_c(z)$ are the raw features.

Although both the HSCRF and the Semi-CRF are capable of modelling arbitrary segment durations, we use a simple exponential distribution (i.e., weighted features activated at each time step are added

[3] http://www.cnts.ua.ac.be/conll2000/chunking/

up) since it can be processed sequentially and thus is very efficient. For learning, we use a simple online stochastic gradient ascent method. At test time, since the SCRF and the Semi-CRF are able to use the POS tags as input, it is not fair for the DCRF and HSCRF to predict those labels during inference. Instead, we also give the POS tags to the DCRF and HSCRF and perform constrained inference to predict *only* the NP labels. This significantly boosts the performance of the two multi-level models.

Let us look at the difference between the flat setting of SCRF and Semi-CRF and the multi-level setting of DCRF and HSCRF. Let $x = (x_{np}, x_{pos})$. Essentially, we are about to model the distribution $\Pr(x|z) = \Pr(x_{np}|x_{pos}, z) \Pr(x_{pos}|z)$ in the multi-level models while we ignore the $\Pr(x_{pos}|z)$ in the flat models. During test time of the multi-level models, we predict only the x_{np} by finding the maximiser of $\Pr(x_{np}|x_{pos}, z)$. The $\Pr(x_{pos}|z)$ seems to be a waste because we do not make use of it at test time. However, $\Pr(x_{pos}|z)$ does give extra information about the joint distribution $\Pr(x|z)$, that is, modelling the POS process may help to get a smoother estimate of the NP distribution.

The performance of these models is depicted in Figure 4b and we are interested in only the prediction of the noun-phrases since this data has POS tags. Without POS tags given at test time, both the HSCRF and the DCRF perform worse than the SCRF. This is not surprising because the POS tags are always given in the case of SCRF. However, with POS tags, the HSCRF consistently works better than all other models.

6 Discussion and Conclusions

The HSCRFs presented here are not a standard graphical model since the clique structures are not predefined. The potentials are defined on-the-fly depending on the assignments of the ending indicators. Although the model topology is identical to that of shared structure HHMMs [1], the unrolled temporal representation is an undirected graph, and the model distribution is formulated in a discriminative way. Furthermore, the state persistence potentials capture duration information that is not available in the DBN representation of the HHMMs in [6]. Thus, the segmental nature of the HSCRF incorporates the recent semi-Markov CRF [8] as a special case [11].

Our HSCRF is related to the conditional probabilistic context-free grammar (C-PCFG) [9] in the same way that the HHMM is to the PCFG. However, the context-free grammar does not limit the depth of semantic hierarchy, thus making it unnecessarily complicated to map hierarchical structures into its form. Further, it lacks a graphical model representation, and thus cannot utilize the rich set of approximate inference techniques available for standard graphical models.

References

[1] H. H. Bui, D. Q. Phung, and S. Venkatesh. Hierarchical hidden Markov models with general state hierarchy. In *AAAI*, pages 324–329, San Jose, CA, Jul 2004.

[2] S. Fine, Y. Singer, and N. Tishby. The hierarchical hidden Markov model: Analysis and applications. *Machine Learning*, 32(1):41–62, 1998.

[3] T. Kristjannson, A. Culotta, P. Viola, and A. McCallum. Interactive information extraction with constrained conditional random fields. In *AAAI*, pages 412–418, San Jose, CA, 2004.

[4] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, pages 282–289, 2001.

[5] L. Liao, D. Fox, and H. Kautz. Hierarchical conditional random fields for GPS-based activity recognition. In *Proceedings of the International Symposium of Robotis Research (ISRR)*. Springer Verlag, 2005.

[6] K. Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, Computer Science Division, University of California, Berkeley, Jul 2002.

[7] N. Nguyen, D. Phung, S. Venkatesh, and H. H. Bui. Learning and detecting activities from movement trajectories using the hierarchical hidden Markov models. In *CVPR*, volume 2, pages 955–960, Jun 2005.

[8] S. Sarawagi and W. W. Cohen. Semi-Markov conditional random fields for information extraction. In *NIPS*. 2004.

[9] C. Sutton. Conditional probabilistic context-free grammars. Master's thesis, Uni. of Massachusetts, 2004.

[10] C. Sutton, A. McCallum, and K. Rohanimanesh. Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. *JMLR*, 8:693–723, Mar 2007.

[11] T. T. Truyen, D. Q. Phung, H. H. Bui, and S. Venkatesh. Hierarchical semi-Markov conditional random fields for recursive sequential data. Technical report, Curtin University of Technology, http://www.computing.edu.au/˜trantt2/pubs/hcrf.pdf, 2008.

[12] J. Verbeek and B. Triggs. Scene segmentation with CRFs learned from partially labeled images. In *Advances in Neural Information Processing Systems 20*, pages 1553–1560. MIT Press, 2008.