Hidden-state Conditional Random Fields

A. Quattoni, S. Wang, L.-P Morency, M. Collins, T. Darrell; MIT CSAIL

Abstract

We present a discriminative latent variable model for classification problems in structured domains where inputs can be represented by a graph of local observations. A hidden-state Conditional Random Field framework learns a set of latent variables conditioned on local features. Observations need not be independent and may overlap in space and time. We evaluate our model on object detection and gesture recognition tasks.

1. Introduction

It is well known that models which include latent, or hidden-state, structure may be more expressive than fully observable models, and can often find relevant substructure in a given domain. Hidden Markov Models (HMMs) and Dynamic Bayesian Networks use hidden state to model observations, and have a clear generative probabilistic formulation. In this paper we develop a hidden-state conditional random field model, and demonstrate its ability to outperform generative hidden-state and discriminative fully-observable models on object and gesture recognition tasks.

Conditional Random Fields have been shown to be powerful discriminative models because they can incorporate essentially arbitrary feature-vector representations of the observed data points [13]. However, they are limited in that they cannot capture intermediate structures using hidden-state variables. In this paper we propose a new model for classification based on CRFs augmented with latent state, which we call Hidden-state Conditional Random Fields (HCRFs). While HMMs are the natural extension of MRFs, HCRFs are the analogous extension for CRFs.

Figure 3 shows the difference between HCRFS and HMMS or Hidden markov random fields. Hidden Markov random fields are directed graphical models [8], where a random variable **h** is modeled as a markov process and it is assumed that the observation variable **x** is a deterministic or stochastic function of **h**. One way of using HMMS for classification is to assume a hidden variable **h** for each category and train a model for each of them independently. That is, given k categories and m samples (where D_l is the training data for category l)in a maximum likelihood framework the parameters θ_l of each of the k models are trained to maximize $P(D_l|\theta_l)$.

Differently, an HCRF models the distribution $P(c, \mathbf{h}|\mathbf{x})$ directly, where c is a category and **h** is an intermediate hidden variable modeled as a markov random field globally conditioned on observation **x**. The parameters θ of the model are trained discriminatively to optimize $P(c|\mathbf{x})$.

There is an extensive literature dedicated to gesture recognition; for hand and arm gestures, a comprehensive survey was presented in Pavlovic *et al.* [20]. Generative models have been used successfully to recognize arm gestures [2] and a number of sign languages [1, 24]. Kapoor and Picard presented a HMM-based, real time head nod and head shake detector [9]. Fugie *et al.* also used HMMs to perform head nod recognition [6].

The main limitation of latent generative approaches is that they require a model of local features given underlying variables, and generally presume independence of the observations. Accurately specifying such a generative model may be challenging, particulary in cases where we wish to incorporate long range dependencies in the model and allow hidden variables to depend on several local features. These observations led to the introduction of discriminative models for sequence labeling, including MEMM's [15], [22] and Conditional Random Fields (CRFs). CRFs were first introduced by Lafferty *et al.* [13] and have been widely used since then in the natural language processing community for tasks such as noun co-reference resolution [17], named entity recognition [16] and information extraction [3].

In computer vision, CRF's have been applied to the task of detecting man-made structures in natural images and have been shown to outperform Markov Random Fields (MRF) [12]. Sminchisescu [23] applied CRFs to classify human motion activity and demonstrated their model was more accurate than MEMMs and could discriminate subtle motion styles. Torralba *et al.* [25] introduced Boosted Random Fields, a model that combines local and global image information for contextual object recognition.

Our hidden-state discriminative approach for object recognition is related to the work of Kumar and Herbert [12], [11], who train a discriminative model using fully-labeled data where each image region is assigned a part label from a discrete set of object parts. A CRF is trained and detection and segmentation are performed by finding the most likely labeling of the image under the learned model. The main difference between our approach and Kumar's is that we do not assume that the part assignment variables are fully observed and are instead regarded as latent variables. Incorporating hidden variables allows use of training data not explicitly labeled with part (hidden-state) structure.

Another related model is presented in [26], which builds a discriminative classifier based on a part-based feature representation. Such a representation is obtained by measuring the similarity between image patches (detected with an interest point detector) to a pre-defined dictionary of parts. The dictionary is built by extracting and clustering patches from a set of representative images of the target class. Again, a significant difference between their approach and ours is that we do not perform a pre-selection of discriminative parts, but rather incorporate such a step during training. In parallel to our work on object recognition [21], [7] developed a hidden-state CRF model for phone recognition and demonstrated the equivalence of HMM models to a subset of CRF models. Also, Koo and Collins [10] describe a similar hidden state model applied to a reranking approach for natural language parsing.

In previous work on CRFs label sequences are typically taken to be fully observed on training examples. In our approach category labels are observed, but an additional layer of subordinate labels are learned. These intermediate hidden variables model the latent structure of the input domain; our model defines the joint probability of a class label and hidden state labels conditioned on the observations, with dependencies between the hidden variables expressed by an undirected graph. The result is a model where inference and parameter estimation can be carried out using standard graphical model algorithms such as loopy belief propagation.

2. Hidden Conditional Random Fields

We presume a task where we wish to predict a label y given inputs. Each y is a member of a set $\mathcal{Y}$ of possible labels and each vector $\mathbf{x}$ is a vector of local observations $\mathbf{x} = \{x_1, x_2, \ldots, x_m\}$. The number of local observations can vary across examples; for convenience of notation we omit dependence on the example index and simply refer to the number of observations as m in each case. Each local observation x_j is represented by a feature vector $\phi(x_j) \in \Re^d$, where d is the dimensionality of the representation. For our object recognition task this corresponds to an image patch descriptor, while for our gesture recognition task this contains body motion observations. Our training set consists of labeled examples $(\mathbf{x}_i, y_i)$ for $i = 1 \ldots n$, where each $y_i \in \mathcal{Y}$, and each $\mathbf{x}_i = \{x_{i,1}, x_{i,2}, \ldots, x_{i,m}\}$. For any example $\mathbf{x}$ we also assume a vector of latent variables $\mathbf{h} = \{h_1, h_2, \ldots, h_m\}$, which are not observed on training examples, and where each h_j is a member of $\mathcal{H}$ where $\mathcal{H}$ is a finite set of possible hidden labels in the model. Intuitively, each h_j corresponds to a labeling of x_j with some member of $\mathcal{H}$, which may correspond to "part" or "sub-gesture" structure in an observation.

Given these definitions of labels y, observations $\mathbf{x}$, and latent variables $\mathbf{h}$, we define a conditional probabilistic model:

Given a new test example $\mathbf{x}$, and parameter values θ^* induced from a training set, we will take the label for the example to be $\arg\max_{y \in \mathcal{Y}} P(y \mid \mathbf{x}, \theta^*)$. Following previous work on CRFs [13, 12], we use the following objective function to estimate the parameters:

$$L(\theta) = \sum_i \log P(y_i \mid \mathbf{x}_i, \theta) - \frac{1}{2\sigma^2}||\theta||^2 \tag{1}$$

The first term in Eq. 1 is the log-likelihood of the data. The second term is the log of a Gaussian prior with variance σ^2, i.e., $P(\theta) \sim \exp\left(\frac{1}{2\sigma^2}||\theta||^2\right)$. We use gradient ascent to search for the optimal parameter values, $\theta^* = \arg\max_\theta L(\theta)$, under this criterion. As with other hidden state models (e.g., HMMs), adding hidden state makes the optimization non-convex; we search for parameters by initializing from multiple random start points and searching for the best local optimum."

We encode structural constraints with an undirected graph structure, where the hidden variables

$\{h_1, \ldots, h_m\}$ correspond to vertices in the graph. The set of graph edges $(j, k) \in E$ denotes links between variables h_j and h_k. The graph E can be arbitrary; intuitively it should capture any domain specific knowledge that we have about the structure of $\mathbf{h}$. In our object recognition task it is a local mesh that encodes spatial consistency between local appearance features, while in our gesture recognition task it is a chain that captures temporal dynamics.

We define Ψ to take the following form:

$$\Psi(y, \mathbf{h}, \mathbf{x}; \theta) = \sum_{j=1}^{m} \sum_{l \in L^1} f_l^1(j, y, h_j, \mathbf{x})\theta_l^1 + \sum_{(j,k) \in E} \sum_{l \in L^2} f_l^2(j, k, y, h_j, h_k, \mathbf{x})\theta_l^2 \tag{2}$$

where L^1 is the set of node features, L^2 the set of edge features , f_l^1, f_l^2 are functions defining the features in the model, and θ_l^1, θ_l^2 are the components of θ. The f^1 features depend on single hidden variable values in the model; the f^2 features can depend on pairs of values. Note that Ψ is linear in the parameters θ, and the model in Eq. **??** is a log-linear model. Moreover the features respect the structure of the graph, in that no feature depends on more than two hidden variables h_j, h_k, and if a feature does depend on variables h_j and h_k there must be an edge (j, k) in the graph E.

Assuming that the edges in E form a tree, and that Ψ takes the form in Eq. 2, then exact methods exist for inference and parameter estimation in the model. This follows because belief propagation can be used to calculate the following quantities in $O(|E||\mathcal{Y}|)$ time:

$$\forall y \in \mathcal{Y}, \quad Z(y \mid \mathbf{x}, \theta) = \sum_{\mathbf{h}} \exp\{\Psi(y, \mathbf{h}, \mathbf{x}; \theta)\}$$

$$\forall y \in \mathcal{Y}, j \in 1 \ldots m, a \in \mathcal{H}, \quad P(h_j = a \mid y, \mathbf{x}, \theta) = \sum_{\mathbf{h}: h_j = a} P(\mathbf{h} \mid y, \mathbf{x}, \theta)$$

$$\forall y \in \mathcal{Y}, (j, k) \in E, a, b \in \mathcal{H}, \quad P(h_j = a, h_k = b \mid y, \mathbf{x}, \theta) = \sum_{\mathbf{h}: h_j = a, h_k = b} P(\mathbf{h} \mid y, \mathbf{x}, \theta)$$

The first term $Z(y \mid \mathbf{x}, \theta)$ is a partition function defined by a summation over the $\mathbf{h}$ variables. Terms of this form can be used to calculate $P(y \mid \mathbf{x}, \theta) = Z(y \mid \mathbf{x}, \theta) / \sum_{y'} Z(y' \mid \mathbf{x}, \theta)$. Hence inference—calculation of $\arg\max P(y \mid \mathbf{x}, \theta)$— can be performed efficiently in the model. The second and third terms are marginal distributions over individual variables h_j or pairs of variables h_j, h_k corresponding to edges in the graph. The gradient of $L(\theta)$ can be defined in terms of these marginals, and hence can be calculated efficiently. If E contains cycles then approximate methods, such as loopy belief-propagation, may be necessary for inference and parameter estimation.

In brief, since $P(\mathbf{h}, y|\mathbf{x}) = P(\mathbf{h}|y, \mathbf{x})P(y|\mathbf{x})$ and since both terms on the right hand side can be efficiently computed using belief propagation it follows that the joint distribution can also be computed efficiently."

We estimate parameters $\theta^* = \arg\max L(\theta)$ from a training set using a quasi-Newton method. The gradient of $L(\theta)$ can be calculated efficiently using belief propogation update steps; the likelihood term due to the i'th training example is:

$$L_i(\theta) = \log P(y_i \mid \mathbf{x}_i, \theta) = \log \left(\frac{\sum_{\mathbf{h}} e^{\Psi(y_i, \mathbf{h}, \mathbf{x}_i; \theta)}}{\sum_{y', \mathbf{h}} e^{\Psi(y', \mathbf{h}, \mathbf{x}_i; \theta)}}. \right) \tag{3}$$

We first consider derivatives with respect to the parameters θ^1_l corresponding to features $f^1_l(j, y, h_j, \mathbf{x})$ that depend on single hidden variables. Taking derivatives gives

$$\begin{aligned}
\frac{\partial L_i(\theta)}{\partial \theta^1_l} &= \sum_{\mathbf{h}} P(\mathbf{h} \mid y_i, \mathbf{x}_i, \theta) \frac{\partial \Psi(y_i, \mathbf{h}, \mathbf{x}_i; \theta)}{\partial \theta^1_l} - \sum_{y', \mathbf{h}} P(y', \mathbf{h} \mid \mathbf{x}_i, \theta) \frac{\partial \Psi(y', \mathbf{h}, \mathbf{x}_i; \theta)}{\partial \theta^1_l} \\
&= \sum_{\mathbf{h}} P(\mathbf{h} \mid y_i, \mathbf{x}_i, \theta) \sum_{j=1}^{m} f^1_l(j, y_i, h_j, \mathbf{x}_i) - \sum_{y', \mathbf{h}} P(y', \mathbf{h} \mid \mathbf{x}_i, \theta) \sum_{j=1}^{m} f^1_l(j, y', h_j, \mathbf{x}_i) \\
&= \sum_{j, a} P(h_j = a \mid y_i, \mathbf{x}_i, \theta) f^1_l(j, y_i, a, \mathbf{x}_i) - \sum_{y', j, a} P(h_j = a, y' \mid \mathbf{x}_i, \theta) f^1_l(j, y', a, \mathbf{x}_i)
\end{aligned}$$

It follows that $\frac{\partial L_i(\theta)}{\partial \theta^1_l}$ can be expressed in terms of components $P(h_j = a \mid \mathbf{x}_i, \theta)$ and $P(y \mid \mathbf{x}_i, \theta)$, which can be calculated using belief propagation, provided that the graph E forms a tree structure. A similar calculation gives

$$\begin{aligned}\frac{\partial L_i(\theta)}{\partial \theta_l^2} &= \sum_{(j,k)\in E,a,b} P(h_j = a, h_k = b \mid y_i, \mathbf{x}_i, \theta) f_l^2(j,k,y_i,a,b,\mathbf{x}_i) \\ &\quad - \sum_{y',(j,k)\in E,a,b} P(h_j = a, h_k = b, y' \mid \mathbf{x}_i, \theta) f_l^2(j,k,y',a,b,\mathbf{x}_i)\end{aligned}$$

hence $\partial L_i(\theta)/\partial\theta_l^2$ can also be expressed in terms of expressions that can be calculated using belief propagation.

3. Experiments

We explored the performance of our HCRF model on both object and gesture recognition tasks, measuring the effect of different degrees of connectivity in the mesh of local observations in the former task and the chain of motion observations in the latter task.

In our experiments we use a restricted form of Ψ where observations interact only with the hidden states:

$$\Psi(y, \mathbf{h}, \mathbf{x}; \theta) = \sum_j \phi(x_j) \cdot \theta(h_j) + \sum_j \theta(y, h_j) + \sum_{(j,k)\in E} \theta(y, h_j, h_k) \tag{4}$$

where $\theta(h_j) \in \Re^d$ for $h_j \in \mathcal{H}$ is a parameter vector corresponding to the j'th latent variable. The inner-product $\phi(x_j) \cdot \theta(h_j)$ can be interpreted as a measure of the compatibility between observation x_j and hidden-state h_j, the parameter $\theta(y, h_j) \in \Re$ for $h_j \in \mathcal{H}$, $y \in \mathcal{Y}$ can be interpreted as a measure of the compatibility between latent variable h_j and category label y, and each parameter $\theta(y, h_i, h_j) \in \Re$ for $y \in \mathcal{Y}$, and $h_i, h_j \in \mathcal{H}$ measures the compatibility between an edge with labels h_i and h_j and the label y. For these experiments we chose the number of hidden states that minimized the training error,the same is true for the number of hidden states and mixtures used in the HMMs.

In general however the number of hidden states could be better optimized through a held out validation set. To give an idea of the sensitivity of the results to this number [Table xxx nips] shows object recognition results for different number of hidden states.

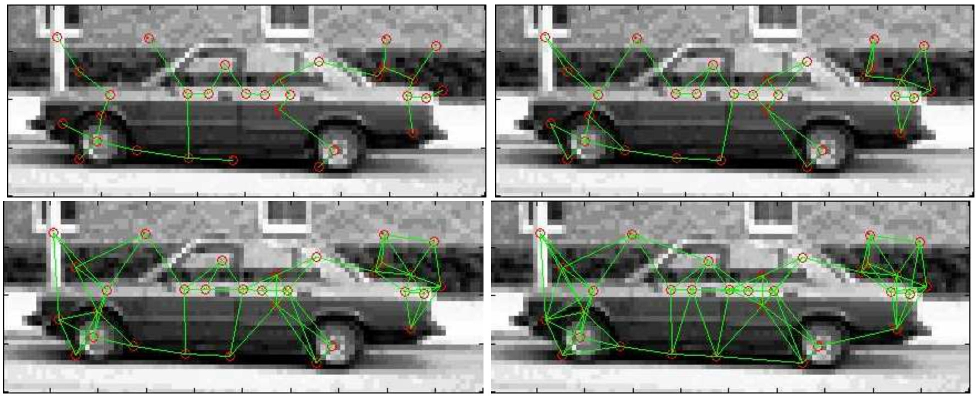

Figure 1: Encoding Part Dependencies in the model: images show min-spanning tree , 1-lattices (top) , 2 , and 3-lattices (bottom) over detected features.

Data set	5 parts	10 parts
Car Side	94	99
Car Rear	91	91.7

In the object recognition domain patches $x_{i,j}$ in each image are obtained using the SIFT detector [14]: each patch $x_{i,j}$ is then represented by a feature vector $\phi(x_{i,j})$ that incorporates a combination of SIFT descriptor and relative location and scale features. We assume that parts conditioned on proximate observations are likely to be dependent, as expressed in the neighborhood graph structure.

The graph E encodes the amount of connectivity between the hidden variables h_j. Intuitively, E determines the ability of our model to capture conditional dependencies between part assignments. Such dependencies between hidden part assignments can be encoded using n-neighbor lattices over local observations. However, increasing connectivity leads to an increase in the computational complexity of performing inference in such models. If E has no connectivity (i.e E contains no edges) the potential function for our model reduces to:

$$\Psi(y, \mathbf{h}, \mathbf{x}; \theta) = \sum_j \phi(x_j) \cdot \theta(h_j) + \sum_j \theta(y, h_j) \tag{5}$$

This graph may be too poor to capture important dependencies between part assignments, especially given that our observations often contain overlapping image patches. Another option for defining E is to use a minimum spanning tree where the weights on the edges are the distances between the correspond-

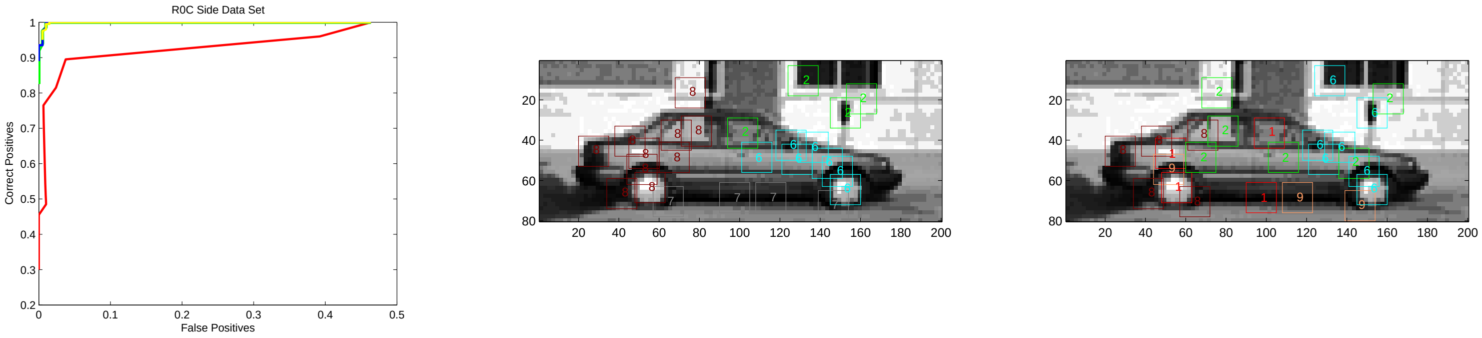

Figure 2: ROC curves for the 4 variants of the model: the red curve corresponds to a model with no connectivity, the green curve to a model with minimum spanning tree connectivity, the blue curve to a model with 2-Lattice connectivity and the yellow curve to a model with 3-Lattice connectivity; Viterbi assignments of hidden states to local image patches for min spanning tree and unconnected model, center and right respectively.

ing image patches. Distances could in general be based on any aspect of the location or feature space; in our experiments below we relied on distance in the image plane. Note that the structure of E will vary across different images. The advantage of using such a graph is that, as we mentioned earlier, when E contains no cycles, and Ψ takes the form in Eq. 2, we can perform exact inference on E, using belief propagation in time $O(|E||\mathcal{Y}|)^2$.

More generally, we define E to be an n-Lattice over the local observations. We build an n-neighbor lattice by linking every node to its n closest nodes, (i.e. the nodes that correspond to the n closest local observations). When E contains cycles computing exact inference becomes untractable, and we need to resort to approximate methods using loopy-belief propagation techniques.

We evaluated the effect of different neighborhood structures on recognition performance in a simple object category recognition task. Here for brevity we report results only for the well-known UIUC car side dataset. Given a neighborhood structure for our model we trained a binary classifier to distinguish between a category and a background set formed from the remaining UIUC images. The data set was split into 3 data sets: a training data set of 200 images, a validation set of 100 images and a testing set of 100 images. The validation set was used to select the regularization term (i.e. the variance of the Gaussian prior) and as a stopping criteria for the gradient ascent.

Data set	Our Model	Others [1]
Car Side	99 %	-
Car Rear	94.6 %	90.3 %
Face	99 %	96.4 %
Plane	96 %	90.2 %
Motorbike	95 %	92.5 %

Figure 3: Comparison with state of the art approach for object recognition (Equal Error Rates)

For the first experiment we defined E to be an unconnected graph, for the second a minimum spanning tree, for the third a 2-lattice, and for the fourth a 3-lattice, as shown in Figure 1. For the first and second experiments gradient ascent was initialized randomly while for the third and fourth experiments we used the minimum spanning tree solution as initial parameters. Figure 2 shows the ROC curves for the 4 variants of the model: the red curve corresponds to a model with no connectivity, the green curve to a model with minimum spanning tree connectivity, the blue curve to a model with 2-Lattice connectivity and the yellow curve to a model with 3-Lattice connectivity.

From this Figure we observe a significant improvement in performance when the model incorporates some degree of dependency between the latent variables (Figure 1).

Figure 2 shows the most likely assignment of parts to features for the min-spanning tree model and the unconnected model for an example in which the former gives a correct classification but the latter fails to do so. Notice that both models give smooth part assignments, which is expected as the normalized location is a feature of the patch representation. In some cases the model relies more heavily on relative location than appearance, labeling a part based on its location and thus learning the shape of the object rather than its appearance. A possible reason for this is that the appearance information might not be very useful for discriminating between classes when the image resolution is too low; as it is the case for the car dataset.

For this type of task the min-spanning tree model shows equivalent recognition performance to the models that use more densely connected graphs. Thus it is clear that the minimum-spanning tree can encode sufficient dependency constraints for certain categories. Table [NIPS table] shows a comparison between our model (with minimum spanning tree connectivity)and previous approaches to object

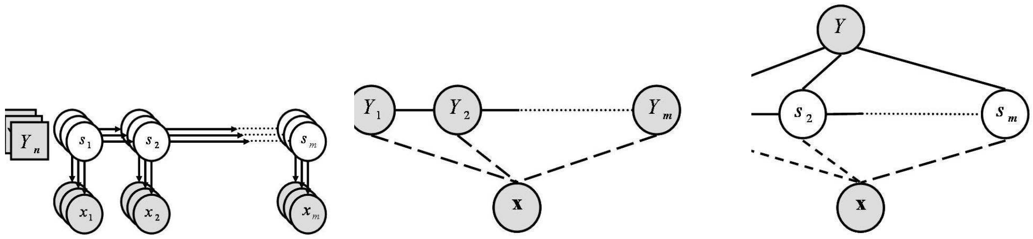

Figure 4: Models used for comparative experiments on the gesture recognition task, Y is the gesture label and S the hidden state labels. The left most figure shows a 'stack of HMMs' model where a separate HMM is trained for each gesture class, the middle figure shows a CRF model and the right most figure the proposed HCRF model.

recognition [5] for a standard dataset (Calteq 4).

While results using local appearance-based feature descriptors have been promising, our model is not limited to such features and could, for example, be defined on a region-based appearance model.

We also explored our HCRF model on body and head gesture recognition, using a chain of observed motion features as the input representation. We evaluated HCRFs with varying levels of long range dependencies, and compared performance to baseline CRF and HMM models. Figure 4 shows graphical representations of the HCRF, HMM, and CRF models used in our experiments.

For each gesture class, we first trained a separate HCRF model to discriminate the gesture class from other classes. For a given test sequence, we compared the probabilities given by each of the two-class HCRFs, and the highest scoring model was selected as the recognized gesture. Next, we trained a single joint multi-class HCRF to recognize all classes. Test sequences were run with this model and the gesture class with the highest probability was selected as the recognized gesture. Finally, we conducted experiments that incorporated different long range dependencies. To incorporate long range dependencies in the CRF and HCRF models, we modify the potential function to include a window parameter ω that defines the amount of past and future history to be used when predicting the state at time t. ($\omega = 0$ indicates only the current observation is used).

The HMM models were trained using maximum likelihood; the number of Gaussian mixtures and states were set by minimizing the error on the training data; in general this parameters can be optimized with a held out dataset, we didn't use a held-out validation because we had a small dataset available

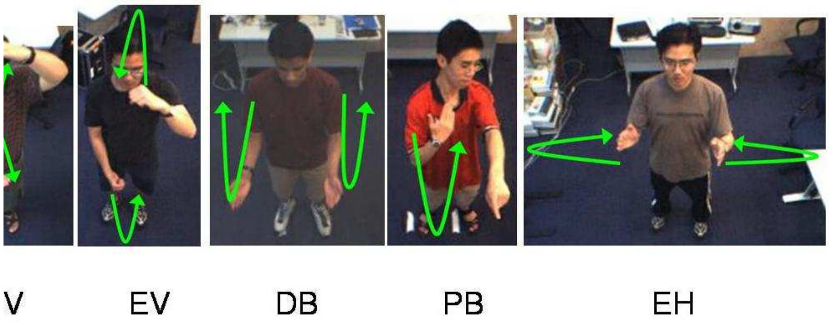

Figure 5: Illustrations of the six gesture classes for the experiments. Below each image is the abbreviation for the gesture class. The green arrows are the motion trajectory of the fingertip and the numbers next to the arrows symbolizes the order of these arrows

for training. The CRF was trained as a multi-way classifier where each state in the model represented one gesture class. Six hidden states were used for the one-vs-all HCRFs, 12 for the multi-class HCRFS, these states where shared among all the classes. For the HMM model we used 4 hidden states for each class, these states were not shared among the different classes.

We ran experiments in two domains: arm and head gestures. In the arm gesture domain, we used a dataset of gestures defined for a virtual manipulation task (see Figure 5). There were six gestures in the dataset. In the Expand Horizontally (EH) arm gesture, the user starts with both arms close to the hips, moves both arms laterally apart and retracts back to the resting position. In the Expand Vertically (EV) arm gesture the arms move vertically apart and return to the resting position. In the Shrink Vertically (SV) gesture both arms begin from the hips, move vertically together and back to the hips. In the Point and Back (PB) gesture the user points with one hand and beckons with the other. In the Double Back (DB) gesture, both arms beckon towards the user. Lastly in the Flip Back (FB) gesture, the user simulates holding a book with one hand while the other hand makes a flipping motion, to mimic flipping the pages of the book.

Users were asked to perform these gestures in front of a stereo camera. From each image frame, a 3D cylindrical body model, consisting of a head, torso, arms and forearms was estimated using a stereo-tracking algorithm [4]. From these body models, both the joint angles and the relative co-ordinates of the joints of the arms are used as observations for our experiments. Thirteen users were asked to perform

Arm Gesture	Avg. Accuracy(%)
HMM ω = 0	84.22
CRF ω = 0	86.03
CRF ω = 1	81.75
HCRF (one-vs-all) ω = 0	87.49
HCRF (multiclass) ω = 0	91.64
HCRF (multiclass) ω = 1	93.81
HCRF (multiclass) ω = 2	93.07
HCRF (multiclass) ω = 3	92.50

Table 1: Comparison of recognition performance (percentage accuracy) for body poses estimated from image sequences on 6-way classification task.

these six gestures; an average of 90 gestures per class were collected.

Table 1 summarizes results for the arm gesture recognition experiments. In these experiments the CRF performed better than HMMs at window size zero. At window size one, however, the CRF performance was poorer; this may be due to overfitting when training the CRF model parameters. Both multi-class and one-vs-all HCRFs perform better than HMMs and CRFs. The most significant improvement in performance was obtained when we used a multi-class HCRF, suggesting that it is important to jointly learn the best discriminative structure.

Figure 6 shows the distribution of states for different gesture classes learned by the best performing model (multi-class HCRF). This graph was obtained by computing the Viterbi path for each sequence (i.e. the most likely assignment for the hidden state variables) and counting the number of times that a given state occurred among those sequences. As we can see, the model has found a unique distribution of hidden states for each gesture, and there is a significant amount of state sharing among different gesture classes.

From the results in table 1, we can see that incorporating some degree of long range dependencies is important, since the HCRF performance improved when the window size was increased from 0 to 1. However, we also see that further increasing the window size did not improve performance.

We also conducted experiments with a head gesture datase obtained using the pose tracking system of [18]. The fast Fourier transform of the 3D angular velocities were used as input features. The

Models	Accuracy (%)
HMM ω = 0	65.33
CRF ω = 0	66.53
CRF ω = 1	68.24
HCRF (multi-class) ω = 0	71.88
HCRF (multi-class) ω = 1	85.25

Table 2: Comparison of recognition performance for head gestures.

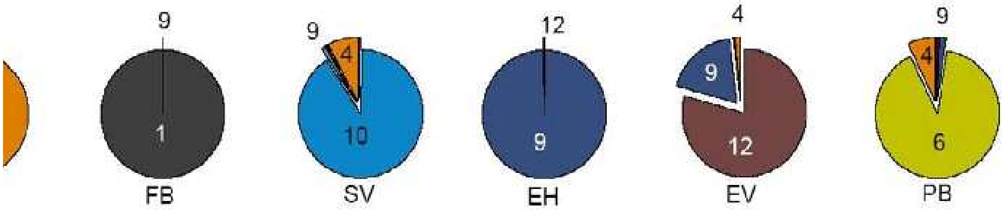

Figure 6: Graph showing the distribution of the hidden states for each gesture class. The numbers in each pie represent the hidden state label, and the area enclosed by the number represents the proportion.

data consisted of interactions between human participants and a robotic character [19]. A total of 16 participants interacted with a robot, with each interaction lasting between 2 to 5 minutes. A total of 152 head nods, 11 head shakes and 159 junk sequences were extracted based on ground truth labels. The junk class had sequences that did not contain any head nods or head shakes during the interactions with the robot. For all experiments in this paper, we separated the data such that the testing dataset had no participants from the training set.

Table 2 summarizes the results for the head gesture experiments. The multi-class HCRF model performs better than the HMM and CRF models at a window size of zero. The CRF has slightly better performance than the HMMs for the head gesture task, and this performance improved with increased window sizes. The HCRF multi-class model made a significant improvement when the window size was increased, which indicates that incorporating long range dependencies was useful.

Notice that for the CRF model increasing the window size from 0 to 1 degrades its performance. This seems surprising since one would expect that adding contextual features could never harm the predictive power of the model. It is very likely that this degrade in performance is caused by over-fitting;

since adding contextual features increases the complexity of the model. However, we do not observe a performance drop for the HCRF model which would seem to suggest that this model is less susceptible to over-fitting; perhaps the presence of local minima prevents the model from over-optimizing its parameters.

4. Summary and Conclusions

We have developed a discriminative hidden-state model and demonstrated its utility on visual recognition tasks. Our model combines the ability of CRFs to use dependent input features and the ability of HMMs to learn latent structure; we train a single joint model which shares hidden states for all classes. Our results have shown that our HCRFs outperform both CRFs and HMMs for certain gesture recognition tasks. For arm gestures, the multi-class HCRF model outperforms HMMs and CRFs even when long range dependencies are not used, demonstrating the advantages of joint discriminative learning. For the object recognition dataset our results have shown that incorporating dependencies between latent variables is important and that the minimum-spanning tree formulation can be a good approximation to more highly connected models.

References

[1] M. Assan and K. Groebel. Video-based sign language recognition using hidden markov models. In *Int'l Gest Wksp: Gest. and Sign Lang.*, 1997.

[2] M. Brand, N. Oliver, and A. Pentland. Coupled hidden markov models for complex action recognition. In *CVPR*, 1996.

[3] A. Culotta and P. V. amd A. Callum. Interactive information extraction with constrained conditional random fields. In *AAAI*, 2004.

[4] D. Demirdjian and T. Darrell. 3-d articulated pose tracking for untethered deictic reference. In *Int'l Conf. on Multimodal Interfaces*, 2002.

[5] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, 2003.

[6] S. Fujie, Y. Ejiri, K. Nakajima, Y. Matsusaka, and T. Kobayashi. A conversation robot using head gesture recognition as para-linguistic information. In *Proceedings of 13th IEEE International Workshop on Robot and Human Communication, RO-MAN 2004*, pages 159–164, September 2004.

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

[8] A. K. H.Kuensch, S. Geman. Hidden markov random fields. In *Annals of Appied Probability, Vol.5.*, 2005.

[9] A. Kapoor and R. Picard. A real-time head nod and shake detector. In *Proceedings from the Workshop on Perspective User Interfaces*, November 2001.

[10] T. Koo and M. Collins. Hidden-variable models for discriminative reranking. In *In proceedings of EMNLP 2005.*, 2005.

[11] S. Kumar and M. Hebert. Multiclass discriminative fields for parts-based object detection. In *Snowbird Learning Workshop*, 2004.

[12] S. Kumar and M. Herbert. Discriminative random fields: A framework for contextual interaction in classification. In *ICCV*, 2003.

[13] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: probabilistic models for segmenting and labelling sequence data. In *ICML*, 2001.

[14] D. Lowe. Object recognition from local scale-invariant features. In *IEEE Int Conference on Computer Vision*, 1999.

[15] A. McCallum, D. Freitag, and F. Pereira. Maximum entropy markov models for information extraction and segmentation. In *ICML*, 2000.

[16] A. McCallum and W. Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *CoNLL*, 2003.

[17] A. McCallum and B. Wellner. Toward conditional models of identity uncertainty with application to proper noun coreference. In *IJCAI Workshop on Information Integration on the Web*, 2003.

[18] L.-P. Morency, A. Rahimi, and T. Darrell. Adaptive view-based appearance model. In *CVPR*, 2003.

[19] L.-P. Morency, C. Sidner, C. Lee, and T. Darrell. Contextual recognition of head gestures. In *ICMI*, 2005.

[20] V. I. Pavlovic, R. Sharma, and T. S. Huang. Visual interpretation of hand gestures for human-computer interaction. In *PAMI*, volume 19, pages 677–695, 1997.

[21] A. Quattoni, M. Collins, and T. Darrell. Conditional random fields for object recognition. In *NIPS*, 2004.

[22] A. Ratnaparkhi. A maximum entropy part-of-speech tagger. In *EMNLP*, 1996.

[23] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Conditional models for contextual human motion recognition. In *Int'l Conf. on Computer Vision*, 2005.

[24] T. Starner and A. Pentland. Real-time asl recognition from video using hidden markov models. In *ISCV*, 1995.

[25] A. Torralba, K. Murphy, and W. Freeman. Contextual models for object detection using boosted random fields. In *NIPS*, 2004.

[26] M. Yang, D. Roth, and N. Ahuja. Learning to recognize 3d objects with snow. In *Proceedings of the Sixth European Conference on Computer Vision*, 2000.