Infinite Hidden Conditional Random Fields for Human Behavior Analysis

Konstantinos Bousmalis, *Student Member, IEEE*,
Stefanos Zafeiriou, *Member, IEEE*,
Louis-Philippe Morency, *Member, IEEE*,
and Maja Pantic, *Fellow, IEEE*

***Abstract*— Hidden conditional random fields (HCRFs) are discriminative latent variable models that have been shown to successfully learn the hidden structure of a given classification problem (provided an appropriate validation of the number of hidden states). In this brief, we present the infinite HCRF (iHCRF), which is a nonparametric model based on hierarchical Dirichlet processes and is capable of automatically learning the optimal number of hidden states for a classification task. We show how we learn the model hyperparameters with an effective Markov-chain Monte Carlo sampling technique, and we explain the process that underlines our iHCRF model with the Restaurant Franchise Rating Agencies analogy. We show that the iHCRF is able to converge to a correct number of represented hidden states, and outperforms the best finite HCRFs—chosen via cross-validation—for the difficult tasks of recognizing instances of agreement, disagreement, and pain. Moreover, the iHCRF manages to achieve this performance in significantly less total training, validation, and testing time.**

***Index Terms*— Discriminative models, hidden conditional random fields, nonparametric Bayesian learning.**

I. Introduction

Hidden conditional random fields (HCRFs) [1] are discriminative models that learn the joint distribution of a class label and a sequence of latent variables conditioned on a given observation sequence, with dependencies among latent variables expressed by an undirected graph. HCRFs learn not only the hidden states that discriminate one class label from all the others, but also the structure that is shared among labels. HCRFs are well suited for a number of problems, including object recognition, gesture recognition [1], speech modeling [2], and multimodal cue modeling for agreement/disagreement recognition [3]. A limitation of the HCRFs is that finding the optimal number of hidden states for a given classification problem is not always intuitive, and learning the correct number of states is often a trial-and-error process involving cross-validation, which can be computationally very expensive. This limitation motivated our nonparametric HCRF model, which automatically learns the optimal number of hidden states given a specific dataset.

Over the past decade, nonparametric methods have been successfully applied to many existing graphical models, allowing them to grow the number of latent states as necessary to fit the data. A prominent and well-studied example is the infinite hidden Markov model (iHMM or HDP-HMM) [4], [5], which is a Hierarchical Dirichlet Process (HDP)-driven HMM with an infinite number of potential hidden states. Other notable examples include the first such model, i.e., the infinite Gaussian mixture model [6], but also the more recent infinite factorial HMM [7], the Dirichlet process mixture/Markov random field (MRF) [8], and the infinite hidden Markov random field model (iHMRF) [9]. HCRFs are related to HMRFs in that both employ a layer of latent variables with an undirected graph specifying dependencies between those variables. However, there is the important difference that HMRFs model a joint distribution over latent variables and observations, whereas an HCRF is a discriminative model optimizing the conditional probability over latent variables and label given the observations. In fact, all models mentioned above are generative and, to our knowledge, the infinite HCRF (iHCRF) introduced in this brief is the first discriminative nonparametric sequential model with latent variables.

The main contribution of this brief is the use of HDPs to allow an infinite number of hidden states for iHCRF. Since exact inference for an infinite model is intractable, we propose an approximation method based on beam sampling, which is a Markov-chain Monte Carlo (MCMC) sampling technique used effectively to sample iHMM models [5]. We also provide an analogy that can prove helpful in understanding the process underlying iHCRF, namely, the Restaurant Franchise Rating Agencies. We present experiments with beam sampling for iHCRFs on the real-world problems of recognizing instances of agreement, disagreement, and pain in recordings of spontaneous human behavior.

In the following section, we concisely present the HDPs. We present in Section III our iHCRF model, beam sampling for iHCRFs, and our analogy. Finally, we evaluate our model performance in Section IV, and conclude with Section V.

II. Background

Our iHCRF model, like many other nonparametric Bayesian models, rely on HDPs. We present in this section, a brief introduction to Dirichlet Processes (DPs) and HDPs, along with the Chinese Restaurant Franchise, which is an analogy that has proved helpful in explaining HDPs and their generalizations.

Manuscript received November 14, 2011; revised September 21, 2012; accepted September 26, 2012. Date of publication November 30, 2012; date of current version December 18, 2012. This work was supported in part by the European Community's 7th Framework, under Program FP7/20072013, under Grant 231287 (SSPNet), the National Science Foundation, under Grant 1118018, and the U.S. Army Research, Development, and Engineering Command. The work of K. Bousmalis was supported in part by the Google Europe Fellowship in Social Signal Processing. The work of M. Pantic was supported in part by the European Research Council under the ERC Starting Grant Agreement ERC-2007-StG-203143 (MAHNOB).

K. Bousmalis and S. Zafeiriou with the Imperial College London, London SW7 2AZ, U.K. (e-mail: k.bousmalis@imperial.ac.uk; s.zafeiriou@imperial.ac.uk).

M. Pantic is with the Imperial College London, London SW7 2AZ, U.K., and also with the Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, Enschede 7522NB, Netherlands (e-mail: m.pantic@imperial.ac.uk).

L.-P. Morency is with the University of Southern California, Institute for Creative Technologies, Playa Vista, CA 90094 USA (e-mail: morency@ict.usc.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2012.2224882

For a concise but complete discussion of DPs and HDPs, the reader is advised to read [10].

A. Dirichlet Processes

A DP is a distribution of distributions, parameterized by a scale parameter γ and a probability measure Ξ, the basis around which the distributions $G \sim \mathrm{DP}(\gamma, \Xi)$ are drawn, with variability governed by the γ parameter. Sethuraman [11] presented the so-called stick-breaking construction for DPs, which is based on random variables $(\beta'_k)_{k=1}^{\infty}$ and $(h_k)_{k=1}^{\infty}$, where $\beta'_k|\gamma, \Xi \sim \mathrm{Beta}(1, \gamma)$, and $h_k|\gamma, \Xi \sim \Xi$

$$\beta_k = \beta'_k \prod_{l=1}^{k-1}(1-\beta'_l) \qquad G = \sum_{k=1}^{\infty}\beta_k \delta_{h_k} \tag{1}$$

where δ is the Dirac delta function. By letting $\boldsymbol{\beta} = (\beta_k)_{k=1}^{\infty}$, we abbreviate this construction as $\boldsymbol{\beta}|\gamma \sim \mathrm{GEM}(\gamma)$ [11].

Successive draws from G are conditionally independent given G. By integrating G out, the conditional distribution of a draw c_i, given all past draws $\{c_1, c_2, \ldots, c_{i-1}\}$, is

$$c_i|c_1, c_2, \ldots, c_{i-1}, \gamma, \Xi \sim \sum_{k=1}^{K}\frac{n_k}{i-1+\gamma}\delta_{h_k} + \frac{\gamma}{i-1+\gamma}\Xi \tag{2}$$

where n_k is the number of times a draw was assigned h_k.

B. HDPs and the Chinese Restaurant Franchise Analogy

A HDP is a DP, the distributions G_j of which are drawn with scale parameter α_0 and base probability measure G_0, which is itself drawn from a DP

$$G_j|\alpha_0, G_0 \sim \mathrm{DP}(\alpha_0, G_0) \qquad G_0 \sim \mathrm{DP}(\gamma, \Xi).$$

Let $\boldsymbol{\pi_j} = (\pi_{jk})_{k=1}^{\infty}$; according to the stick-breaking construction, the HDP can then be expressed as follows:

$$\boldsymbol{\beta}|\gamma \sim \mathrm{GEM}(\gamma) \qquad \boldsymbol{\pi_j}|\alpha_0, \boldsymbol{\beta} \sim \mathrm{DP}(\alpha_0, \boldsymbol{\beta}) \qquad h_k|\Xi \sim \Xi$$
$$G_j = \sum_{k=1}^{\infty}\pi_{jk}\delta_{h_k}.$$

A useful analogy for understanding HDPs, and their explicit clustering effect, is the Chinese Restaurant Franchise Analogy. According to this analogy, the HDP represents a Chinese Restaurant Franchise, and every G_j represents one of the restaurants in the franchise. All restaurants share the same menu of dishes—hidden states h_k. Using the notation from [10], each restaurant has $m_{j\cdot}$ tables, each of which serves only one dish. There is no limit to the number of tables, and many tables can serve the same dish. The number of tables serving dish k in restaurant j is symbolized by m_{jk}, and the number of tables serving dish k in the entire franchise is $m_{\cdot k}$. The table groupings is a quantity helpful for hyperparameter learning and sampling. A customer c_{ji}, the ith customer to walk in a restaurant j, sits on table l with a probability proportional to the number n_{jl} of previous customers sitting at that table. Customer c_{ji} will refuse to sit on an already occupied table with a probability proportional to α_0. In the former case, the customer will have whatever dish everyone is having at the chosen table. In the latter case, the customer will sit at an unoccupied table and will choose a dish for the table—the number of tables will be incremented and a hidden state will be drawn from G_0, with a probability of choosing a completely new dish proportional to γ.

Therefore, the HDP equivalent of (2) is

$$c_{ji}|c_{j1}, \ldots, c_{j(i-1)}, \alpha_0, G_0 \sim \sum_{l=1}^{m_{j\cdot}}\frac{n_{jl}}{i-1+\alpha_0}\delta_{\psi_{jl}} + \frac{\alpha_0}{i-1+\alpha_0}G_0$$

where ψ_{jl} represents the dish served by a specific table, making it clear that an HDP is only parameterized by its hyperparameters α_0, γ, and a row of counts $\mathbf{n}_j$ for each DP G_j of the hierarchy. The above can be reexpressed as

$$\psi_{j,l}|\psi_{1,1}, \ldots, \psi_{j,(l-1)}, \gamma, \Xi \sim \sum_{k=1}^{K}\frac{m_{\cdot k}}{m_{\cdot\cdot}+\gamma}\delta_{h_k} + \frac{\gamma}{i-1+\gamma}\Xi.$$

III. iHCRFs

We concisely discuss, in this section, the finite HCRFs for sequence classification. We then present our iHCRFs as an HDP-based nonparametric extension of the finite model.

A. (Finite) HCRFs

HCRFs—discriminative models that contain hidden states—are well suited to a number of problems. Quattoni *et al.* [1] presented and used them to capture temporal dependencies across frames and recognize different gesture classes. They did so successfully by learning a state distribution among the different gesture classes in a discriminative manner, allowing them to not only uncover the distinctive configurations that uniquely identify each class but also to learn a shared common structure among the classes.

We represent T observations as $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T]$. Each observation at time $t \in \{1, \ldots, T\}$ is represented by a feature vector $\mathbf{f}_t \in \Re^d$, where d is the number of features, which can include any features of the observation sequence. In this brief, we assume that every dimension of $\mathbf{f}_t$ is independent. We wish to learn a mapping between observation sequence $\mathbf{X}$ and class label $y \in \mathcal{Y}$, where $\mathcal{Y}$ is the set of available labels. The HCRF does so by estimating the conditional joint distribution over a sequence of latent variables $\mathbf{s} = [s_1, s_2, \ldots, s_T]$, each of which is assigned to a hidden state $h_i \in \mathcal{H}$, and a label y, given $\mathbf{X}$. One of the main representational powers of HCRFs is that the latent variables can depend on arbitrary features of the observation sequence. This allows us to model long-range contextual dependencies; i.e., s_t, the latent variable at time t, can depend on observations that happened earlier or later than t.

An HCRF, being discriminative, models the conditional probability of a class label given an observation sequence by

$$P(y \mid \mathbf{X}; \boldsymbol{\theta}) = \sum_{\mathbf{s}} P(y, \mathbf{s} \mid \mathbf{X}; \boldsymbol{\theta}) = \frac{\sum_{\mathbf{s}}\Psi(y, \mathbf{s}, \mathbf{X}; \boldsymbol{\theta})}{\sum_{y' \in \mathcal{Y}, \mathbf{s}}\Psi(y', \mathbf{s}, \mathbf{X}; \boldsymbol{\theta})}. \tag{3}$$

The potential function $\Psi(y, \mathbf{s}, \mathbf{X}; \boldsymbol{\theta}) \in \Re$ is parameterized by $\boldsymbol{\theta}$, which measures the compatibility between a label, a

sequence of observations, and a configuration of the hidden states. In this brief, the graph of our model is a chain where each node corresponds to a latent variable s_t at time t. Our parameter vector $\boldsymbol{\theta}$ is made up of three components: $\boldsymbol{\theta} = [\boldsymbol{\theta}^{\boldsymbol{x}T}\ \boldsymbol{\theta}^{\boldsymbol{y}T}\ \boldsymbol{\theta}^{\boldsymbol{e}T}]^T$. Parameter vector $\boldsymbol{\theta}^{\boldsymbol{x}}$ models the relationship between features $\mathbf{f}_t$ and hidden states $h_i \in \mathcal{H}$ and is typically of length $(d \times |\mathcal{H}|)$. $\boldsymbol{\theta}^{\boldsymbol{y}}$ models the relationship of the hidden states $h_i \in \mathcal{H}$ and labels $y \in \mathcal{Y}$ and is of length $(|\mathcal{Y}| \times |\mathcal{H}|)$. $\boldsymbol{\theta}^{\boldsymbol{e}}$ represents the links between hidden states. It is equivalent to the transition matrix in a HMM, but an important difference is that an HCRF keeps a matrix of "transition" weights for each label and $\boldsymbol{\theta}^{\boldsymbol{e}}$ is of length $(|\mathcal{Y}| \times |\mathcal{H}| \times |\mathcal{H}|)$. We define potential functions for each of these relationships, and our Ψ as their product along the chain

$$\Psi(y, \mathbf{s}, \mathbf{X}; \boldsymbol{\theta}) = \Psi^x(\mathbf{s}, \mathbf{X}; \boldsymbol{\theta}^x)\Psi^y(y, \mathbf{s}; \boldsymbol{\theta}^y)\Psi^e(y, \mathbf{s}; \boldsymbol{\theta}^e) \tag{4}$$

$$\Psi^x(\mathbf{s}, \mathbf{X}; \boldsymbol{\theta}^x) = \prod_{t=1}^{T}\prod_{\phi=1}^{d} \psi^x(f_t[\phi], s_t; \boldsymbol{\theta}^x)^{f_t[i]} \tag{5}$$

$$\Psi^y(y, \mathbf{s}; \boldsymbol{\theta}^y) = \prod_{t=1}^{T} \psi^y(y, s_t; \boldsymbol{\theta}^y) \tag{6}$$

$$\Psi^e(y, \mathbf{s}; \boldsymbol{\theta}^e) = \prod_{t=2}^{T} \psi^e(s_{t-1}, y, s_t; \boldsymbol{\theta}^e) \tag{7}$$

$$\psi^x(f_t[\phi], s_t; \boldsymbol{\theta}^x) = \exp\{\theta^x[\phi, s_t]\} \tag{8}$$

$$\psi^y(y, s_t; \boldsymbol{\theta}^y) = \exp\{\theta^y[y, s_t]\} \tag{9}$$

$$\psi^e(s_{t-1}, y, s_t; \boldsymbol{\theta}^e) = \exp\{\theta^e[s_\lambda, y, s_\kappa]\}. \tag{10}$$

In this brief, we use the notation $\theta^x[\phi, h_i]$ and $\psi^x[\phi, h_i]$[1] to refer to the weight or potential that measures the compatibility between the feature indexed by ϕ and state $h_i \in \mathcal{H}$. Similarly, $\theta^y[y, h_i]$ and $\psi^y[y, h_i]$ stand for weights or potentials that correspond to class y and state h_i, whereas $\theta^e[y, h_i, h']$ and $\psi^e[h_i, y, h']$ measure the compatibility of the label y with a transition from h_i to h'.

B. iHCRF Model

Allowing an infinite number of hidden states in $\mathcal{H}$ implies a more flexible model but also the need for an infinite number of weights and corresponding potentials. The key to our iHCRF is that the potentials $\boldsymbol{\psi}^{\boldsymbol{x}}$, $\boldsymbol{\psi}^{\boldsymbol{y}}$, and $\boldsymbol{\psi}^{\boldsymbol{e}}$ are sampled directly from a set of HDPs, a separate process for each of the three $\boldsymbol{\psi}$ groups, i.e., HDP^x, HDP^y, and HDP^e. The choice to use HDPs and not separate DPs was clear and in line with previous work (e.g., [4]) as for each kind of the iHCRF potentials, we want to introduce intraset dependencies that should be different for each kind. The latter was also precisely the reason for our choosing three distinct processes with different hyperparameters to derive our potentials. The iHCRF is therefore not parameterized by weights $\boldsymbol{\theta}$ but by the parameters of those processes that will allow our model to have a potentially infinite number of potentials. We

[1]We abuse here the notation ψ. It was used in Section II to represent the dish served by a specific table in HDPs with notation identical to [10]. It will be used to represent potentials for the rest of this brief.

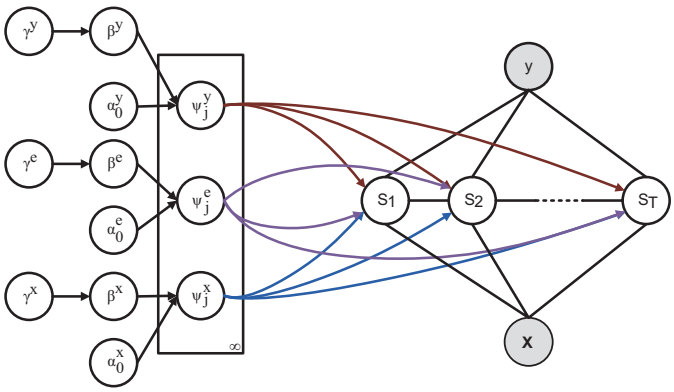

Fig. 1. Graphical representation of our iHCRF. The $\boldsymbol{\psi}^x, \boldsymbol{\psi}^y, \boldsymbol{\psi}^e$ weights are derived from three HDPs conditioned on their hyperparameters $\alpha_0^x, \gamma^x, \alpha_0^y, \gamma^y, \alpha_0^e, \gamma^e$.

define $\boldsymbol{\omega}^x, \boldsymbol{\omega}^y$, and $\boldsymbol{\omega}^e$ to be these sets of parameters for the processes we will use to derive a potentially infinite number of $\boldsymbol{\psi}^{\boldsymbol{x}}$, $\boldsymbol{\psi}^{\boldsymbol{y}}$, and $\boldsymbol{\psi}^{\boldsymbol{e}}$, respectively. The iHCRF, visualized in Fig. 1, is then parameterized only by the six hyperparameters–$\boldsymbol{\omega} = \{\alpha_0^x, \gamma^x, \alpha_0^y, \gamma^y, \alpha_0^e, \gamma^e\}$–and the model can be expressed as follows[2]:

$$\begin{aligned} P(y \mid \mathbf{X}; \boldsymbol{\omega}^x, \boldsymbol{\omega}^y, \boldsymbol{\omega}^e) &\propto \sum_{\mathbf{s}} \Psi^x(\mathbf{s}, \mathbf{X}; \boldsymbol{\omega}^x)\Psi^y(y, \mathbf{s}; \boldsymbol{\omega}^y) \\ &\quad \times \Psi^e(y, \mathbf{s}; \boldsymbol{\omega}^e) \\ \boldsymbol{\omega}^x &= \{\boldsymbol{\alpha_0^x}, \boldsymbol{\gamma^x}\}, \quad \boldsymbol{\omega}^y = \{\alpha_0^y, \gamma^y\}, \quad \boldsymbol{\omega}^e = \{\alpha_0^e, \gamma^e\} \\ \boldsymbol{\beta}^{\boldsymbol{x}} &\sim \mathrm{GEM}(\gamma^x), \quad \boldsymbol{\psi}^{\boldsymbol{x}}[j, :] \mid \boldsymbol{\beta}^{\boldsymbol{x}} \sim \mathrm{DP}(\alpha_0^x, \boldsymbol{\beta}^{\boldsymbol{x}}) \\ \boldsymbol{\beta}^{\boldsymbol{y}} &\sim \mathrm{GEM}(\gamma^y), \quad \boldsymbol{\psi}^{\boldsymbol{y}}[j, :] \mid \boldsymbol{\beta}^{\boldsymbol{y}} \sim \mathrm{DP}(\alpha_0^y, \boldsymbol{\beta}^{\boldsymbol{y}}) \\ \boldsymbol{\beta}^{\boldsymbol{e}} &\sim \mathrm{GEM}(\gamma^e), \quad \boldsymbol{\psi}^{\boldsymbol{e}}[j, :, :] \mid \boldsymbol{\beta}^{\boldsymbol{e}} \sim \mathrm{DP}(\alpha_0^e, \boldsymbol{\beta}^{\boldsymbol{e}}) \end{aligned} \tag{11}$$

where j is the index for the features in $\boldsymbol{\psi}^{\boldsymbol{x}}$, the labels in $\boldsymbol{\psi}^{\boldsymbol{y}}$, or the previous hidden states in $\boldsymbol{\psi}^{\boldsymbol{e}}$.

In iHCRFs, K, which is the same for all three underlying HDPs, is the number of visited states represented in the counts for each process $\mathbf{n}^x, \mathbf{n}^y, \mathbf{n}^e$ (see Section II), which are populated based on a latent variable sequence assignment. The infinitely many unvisited states are represented by an additional single "state," which we call h_{K+1}. Potentials $\boldsymbol{\psi}^{\boldsymbol{x}}$, $\boldsymbol{\psi}^{\boldsymbol{y}}$, and $\boldsymbol{\psi}^{\boldsymbol{e}}$ are also associated with each of these $K+1$ states, and as K changes, so does the length of the number of $\boldsymbol{\psi}^{\boldsymbol{x}}$, $\boldsymbol{\psi}^{\boldsymbol{y}}$, and $\boldsymbol{\psi}^{\boldsymbol{e}}$. The potentials associated with h_{K+1} are $\boldsymbol{\psi}^{\boldsymbol{x}}[:, h_{K+1}]$, $\boldsymbol{\psi}^{\boldsymbol{y}}[:, h_{K+1}]$, $\boldsymbol{\psi}^{\boldsymbol{e}}[:, :, h_{K+1}]$ and represent the compatibility of visiting a new state given the features, the label, or the previous latent variable assignment and the label, respectively. Although all HDPs have the same number of represented states K, each HDP has separate concentration hyperparameters and counts: $\{\alpha_0^x, \gamma^x, \mathbf{n}^x\}$ for HDP^x; $\{\alpha_0^y, \gamma^y, \mathbf{n}^y\}$ for HDP^y; and $\{\alpha_0^e, \gamma^e, \mathbf{n}^e\}$ for HDP^e. Separate $\boldsymbol{\beta}$ sticks are sampled for each HDP, based on which the equivalent proportions $\boldsymbol{\psi}^{\boldsymbol{x}}$, $\boldsymbol{\psi}^{\boldsymbol{y}}$, and $\boldsymbol{\psi}^{\boldsymbol{e}}$ are also sampled.

[2]In this brief, the normalization factor Z is estimated by the forward-filtering performed during beam sampling, as described in Section III-C.

C. Hyperparameter Learning—Beam Sampling for the iHCRF

In this section, we present our technique to automatically learn the hyperparameters based on beam sampling,[3] which is an MCMC sampling method that has successfully been used to sample whole trajectories for the iHMM [5].

Beam sampling achieves forward filtering and backwards sampling for a chain of latent variables by introducing an auxiliary variable u_t for each latent variable s_t. In iHMM, the "beam" u_t acts as a threshold for the transition probabilities; if a transition probability is below the "beam," it is not considered feasible. This means that we are able to sample whole trajectories effectively, by conditioning the transition probabilities on $\mathbf{u}$, making the number of possible trajectories finite. Since we have adopted in this brief a chain structure for our model, the beam sampler can easily be adapted for the iHCRF. Unlike the iHMM, the iHCRF is an undirected model, and the equivalent of transition probabilities are the values $\mathbf{M}_t$, a matrix of the product of node and edge potentials for a specific node-latent variable s_t [1]. In iHCRFs, the node potentials $\mathbf{v}_t$ are

$$\mathbf{v}_t = \left[v_{(t,1)}, \ldots, v_{(t,K)}, v_{(t,K+1)}\right]$$

where

$$v_{(t,k)} = \psi^y[y, h_k] \prod_{i=1}^{d} (\psi^x[h_k])^{f_t[i]}$$

and the edge potentials $\mathbf{B}_t$ are

$$\mathbf{B}_t = \begin{pmatrix} B_{t,(1,1)} & \cdots & B_{t,(1,K)} & B_{t,(1,K+1)} \\ \vdots & \ddots & \vdots & \vdots \\ B_{t,(K,1)} & \cdots & B_{t,(K,K)} & B_{t,(K,K+1)} \end{pmatrix}$$

where

$$B_{t,(k,j)} = \psi^e[y, h_k, h_j]. \tag{12}$$

By letting $\mathbf{1}_K$ be a column vector of K 1s and $\mathbf{A}_t = \mathbf{1}_K \cdot \mathbf{v}_t$, in iHCRF a u_t acts as a threshold for $\mathbf{M}_t$

$$\mathbf{M}_t = \mathbf{A}_t \odot \mathbf{B}_t$$

$$M_t(y, s_{t-1}, s_t, \mathbf{X}; \boldsymbol{\psi}) = \psi^y[y, s_t] \prod_{i=1}^{d} (\psi^x[s_t])^{f_t[i]} \times \psi^e[y, s_{t-1}, s_t]$$

where $\odot$ symbolizes the Hadamard product, i.e., entry-wise matrix multiplication. For example, if $s_t = h_2$ and $s_{t-1} = h_4$, then $M_t(y, h_4, h_2, \mathbf{X}; \boldsymbol{\psi}) = \psi^y[y, h_2] \prod_{i=1}^{d} (\psi^x[h_2])^{f_t[i]} \psi^e[y, h_4, h_2]$, and for the special case of s_t being assigned to one of the infinite unrepresented states, $M_t(y, h_4, h_{K+1}, \mathbf{X}; \boldsymbol{\psi}) = \psi^y[y, h_{K+1}] \prod_{i=1}^{d} (\psi^x[h_{K+1}])^{f_t[i]} \psi^e[y, h_4, h_{K+1}]$.

Given a set of L training sequences $\{\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \ldots, \mathbf{X}^{(L)}\}$, the iHCRF hyperaparameters are learned using the beam sampler as follows.

1) *Initialization:* We choose an initial number for K. For each training sequence, we randomly create a latent variable sequence $\mathbf{s} = \{\mathbf{s}^{(1)}, \mathbf{s}^{(2)}, \ldots, \mathbf{s}^{(L)}\}$.

[3]The code for the beam sampler for iHCRF is available at www.doc.ic.ac.uk/~kb709/.

2) *Sample* $\alpha_0^x, \gamma^x, \alpha_0^y, \gamma^y, \alpha_0^e, \gamma^e, \boldsymbol{\beta^x}, \boldsymbol{\beta^y}, \boldsymbol{\beta^e}, \boldsymbol{\psi^x}, \boldsymbol{\psi^y}, \boldsymbol{\psi^e} | \mathbf{s}$*:* Based on $\mathbf{s}$, we populate the counts $\mathbf{n}^x, \mathbf{n}^y, \mathbf{n}^e$ for each of the three HDPs. We then sample, based on these counts, $\boldsymbol{\beta^x}, \boldsymbol{\beta^y}, \boldsymbol{\beta^e}$, the six hyperparameters, and the potentials $\boldsymbol{\psi^x}, \boldsymbol{\psi^y}, \boldsymbol{\psi^e}$, as shown in the equations and figure that describe our model in the main text. These follow directly from the theory of hyperparameter sampling for HDPs, for details of which the interested reader is referred to [10].
3) *Sample* $\mathbf{u}$*:* For each t in each sequence $\mathbf{X}^{(i)}$ of our training set, we sample $u_t^{(i)}$ from a beta distribution $p(u_t^{(i)} | \mathbf{s}^{(i)}, y^{(i)}, \mathbf{X}^{(i)}; \boldsymbol{\psi}) : u^{(i)} \sim \text{Beta}_{a,b}(0, M_t^{(i)}(y^{(i)}, s_{t-1}^{(i)}, s_t^{(i)}, \mathbf{X}^{(i)}; \boldsymbol{\psi}))$. For our experiments we used $a = 1$ and $b = 2$. Note that the distribution of our beam values does not necessarily have to be a beta one–a uniform distribution was used in the beam sampler of [5]. If there is a $s_{t-1}^{(i)}$ for which $u_t^{(i)} < M_t^{(i)}(y^{(i)}, s_{t-1}^{(i)}, s_t^{(i)} = h_{K+1}, \mathbf{X}^{(i)}; \boldsymbol{\psi})$, i.e., if the $M_t^{(i)}$-value for an unrepresented state is higher than $u_t^{(i)}$, we increment the number of represented states K. Note that by increasing K, the $M_t^{(i)}$-values for an unrepresented state decrease. We repeat this step until $\forall i, t\ u_t^{(i)} > M_t^{(i)}(y^{(i)}, s_{t-1}^{(i)}, s_t^{(i)} = h_{K+1}, \mathbf{X}_t^{(i)}; \boldsymbol{\psi})$.
4) *Sample* $\mathbf{s}$*:* We sample whole trajectories for all sequences in our training set by applying "beam"-assisted forward filtering–backward sampling.

Forward Filtering: We calculate forward probabilities $\mathbf{p}_w$ along the undirected chain for each training sequence as follows (the sequence indicators (i) are omitted for clarity):

$$\mathbf{p}_{w_t} = \frac{1}{\sum \mathbf{p}_{w_t}} \mathbf{v}_t \odot [q(1), q(2), \ldots, q(K+1)]$$

where $q(j)$ is the sum of the jth column of the pointwise product of forward probabilities and edge potentials

$$q(j) = \sum_{k=1}^{K} \mathbf{C}_t(j,k) \qquad \mathbf{C}_t = (\mathbf{1}_K \cdot \mathbf{p}_{w_{t-1}}) \odot \mathbf{B}_t$$

$$\mathbf{p}_{w_1} = \mathbf{v}_1 \cdot \frac{1}{\sum_{k=1}^{K} \mathbf{v}_1(k)}.$$

That is

$$p_{w_t}(s_t | y, \mathbf{f}_{1:t}, u_{1:t}; \boldsymbol{\psi^x}, \boldsymbol{\psi^y}, \boldsymbol{\psi^e}) \propto \psi^y[y, s_t] \prod_{i=1}^{d} (\psi^x[s_t])^{f_t[i]} \cdot \sum_{s_{t-1}} p_{w_t}(s_{t-1} | y, \mathbf{f}_{1:t-1}, u_{1:t-1}) \quad \psi^e[y, s_{t-1}, s_t]. \tag{13}$$

If any values in $\mathbf{M}_t$ are below the "beam" value, we set the corresponding elements in $\mathbf{p}_{w_t}$ to 0, rejecting trajectories that pass through the corresponding hidden state at time t.

Backwards Sampling: Once all forward probabilities are computed, a hidden state is sampled for the last latent variable in the sequence, $s_T \sim \mathbf{p}_{w_T}$. Conditioned on

Algorithm 1 Beam Sampler for iHCRFs

Initialize hidden states $\mathbf{s}$
while number of sampling iterations is not reached **do**
  Sample $\boldsymbol{\psi}^x, \boldsymbol{\psi}^y, \boldsymbol{\psi}^e \mid \mathbf{s}$
  Sample $p(\mathbf{u}|\mathbf{s}, y, \mathbf{X}, \boldsymbol{\psi}) : u_t^{(i)} \sim \text{Beta}_{a,b}(0, M_t^{(i)})$
  for $i = 1 \rightarrow L$ **do**
    for $t = 1 \rightarrow T^{(i)}$ **do**

$$p_{w_t}^{(i)}(s_t^{(i)}|y^{(i)}, \mathbf{f}_{1:t}^{(i)}, u_{1:t}^{(i)}) \propto \psi^y[y, s_t] \prod_{i=1}^{d} (\psi^x[s_t])^{f_t[i]} \times \sum_{s_{t-1}^{(i)}: u_t^{(i)} \leq M_t^{(i)}} \left[p_{w_t}^{(i)}(s_{t-1}^{(i)}|y^{(i)}, \mathbf{f}_{1:t-1}^{(i)}, u_{1:t-1}^{(i)}) \cdot \psi^e[y, s_{t-1}, s_t] \right]$$

    end for
  end for
  for $i = 1 \rightarrow L$ **do**
    Sample $s_T^{(i)} \sim \mathbf{p}_{w_T}^{(i)}$
    for $t = T^{(i)} \rightarrow 1$ **do**
      Sample $p(s_t|s_{t+1}, y) \propto p_{w_t}(s_t)\psi^e[y, s_t, s_{t+1}]$
    end for
  end for
end while

s_T, we sample s_{T-1}, and subsequently the entire chain backwards to s_1

$$p(s_t|s_{t+1}, y, \mathbf{f}_{1:t}, u_{1:t}; \boldsymbol{\psi}) \propto p_{w_t}(s_t|y, \mathbf{f}_{1:t}, u_{1:t}; \boldsymbol{\psi})\psi^e[y, s_t, s_{t+1}].$$

If one of the K represented states is not represented anymore after sampling the entire set of our training sequences, the state and any potentials associated with it are removed and K is decremented.

5) Repeat from Step 2 for a set number of iterations.

The beam sampler for iHCRF is summarized in Algorithm 1.

D. Restaurant Franchise Rating Agencies Analogy

In order to further explain (11) and beam sampling for iHCRFs, we present an analogy in the spirit of the Chinese Restaurant Franchise Analogy, which is widely used to explain generalizations of HDPs (see Section II). In our analogy, we have a number of rating agencies (e.g., Zagat and TimeOut) reviewing restaurant franchises (e.g., Pizza Hut, Gaby's, Strada). Restaurant franchises represent hidden states, and there is one rating agency for each of HDP^x, HDP^y, and HDP^e. Each agency has different criteria with which they rate each restaurant franchise. Each franchisee h_k has a number of branches, and each agency may rate a different number $m_{\cdot k}$ of branches[4] for each h_k, depending on their criteria. A customer represents a single sample of a latent variable $s_t^{(i)}$. The following is our analogy for beam sampling an iHCRF.

[4]The equivalent quantity for tables from the Restaurant Franchise Analogy.

1) A customer $s_t^{(i)}$ sends a query to each rating agency with a number of requirements for her dining experience. These are the variables our sampling is conditioned to. Since we use beam sampling, these are $\{\mathbf{x}_t^{(i)}, y^{(i)}, s_{t-1}^{(i)}\}$.
2) Each restaurant-franchise rating agency HDP^x, HDP^y, and HDP^e rates each franchise based on the user's requirements.
3) The customer chooses, based on the different suggestions from the rating agencies, to dine at a branch of franchise h_k that is listed in the ratings with a probability proportionate to the product of the number of branches rated by the agencies, $m_{\cdot k}^x m_{\cdot k}^y m_{\cdot k}^e$. The customer will choose to dine at a new franchise not listed in any of the agencies with probability proportionate to $\gamma^x \gamma^y \gamma^e$.
4) Our customer notifies the agencies regarding which franchise and branch he chose to dine at—the value assigned to $s_t^{(i)}$—and each rating agency updates their databases $\mathbf{n}^x, \mathbf{n}^y, \mathbf{n}^e$ to improve future suggestions to similar customers.

E. Inference

A learned iHCRF model can be described as the collection of hyperparameters and count tables $\Lambda = \{\alpha_0^x, \gamma^x, \mathbf{n}^x, \alpha_0^y, \gamma^y, \mathbf{n}^y, \alpha_0^e, \gamma^e, \mathbf{n}^e\}$ learned from a training set. Equation (3) can be efficiently evaluated via forward filtering. Since our parameters are not fixed, inference can be achieved by sampling new $\boldsymbol{\psi}^*$ conditioned on Λ. Given a new testing sequence $\bar{\mathbf{X}}$ and one such sampling of $\boldsymbol{\psi}^*$, we will estimate the label of the new sequence to be

$$\arg\max_{y \in Y} P(y \mid \bar{\mathbf{X}}; \Lambda).$$

The iHCRF model will assign the mode of these predictions as the label of the new sequence. We found that for our experiments, the number of samplings for accurate estimates was not higher than 100.

IV. Experiments

The problem of classifying episodes of high-level emotional states, such as pain and agreement and disagreement, based on nonverbal cues in audiovisual sequences of spontaneous human behavior is rather complex. In this brief, we used an audiovisual dataset of spontaneous agreement and disagreement and a visual dataset of pain to evaluate the performance of the proposed iHCRF on four classification problems: 1) agreement and disagreement recognition with two labels (agreement versus disagreement); 2) agreement and disagreement recognition with three labels (agreement versus disagreement versus neutral); 3) pain recognition with two labels (strong pain versus no pain); and 4) pain recognition with three labels (strong pain versus moderate pain versus no pain). We show that: 1) our model is capable of quickly converging to a correct number K of represented states and 2) iHCRFs perform better than the best performing finite HCRF in each of these problems in terms of recognition rates.

The audiovisual dataset of spontaneous agreement and disagreement comprises 53 episodes of agreement, 94 episodes of

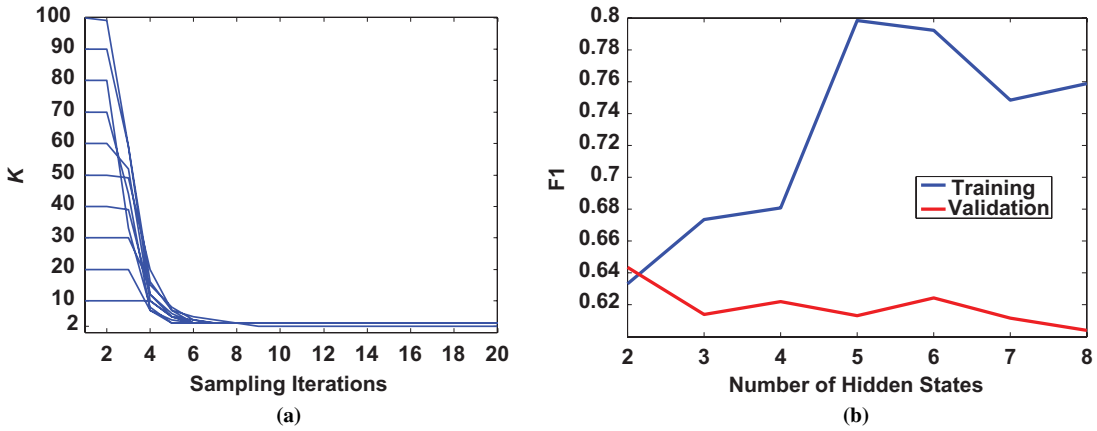

Fig. 2. Convergence analysis of iHCRF number of represented states K for the Canal9 dataset on the three-label problem (ADA3). We ran iHCRF experiments with initial $K = 10, 20, \ldots, 100$ and all of them converged to $K = 2{-}3$, which seems to be the correct value for K for the given problem, as HCRFs with a higher number of hidden states start showing signs of overfitting. (a) Number of represented iHCRF hidden states K for the Canal9 dataset with three labels. (b) Total finite HCRF F1 measures for training and validation sets of the Canal9 dataset with three labels.

disagreement, and 130 neutral episodes of neither agreement nor disagreement. These episodes feature 28 participants and they occur over a total of 11 real political debates from The Canal9 Database of Political Debates[5] [12]. We used automatically extracted prosodic features, based on previous work on agreement and disagreement classification, and manually annotated visual features, the hand and head gestures hypothesized relevant according to the literature [13]. The two prosodic features used were F0 and Energy, and the eight gestures used in our experiments are the "Head Nod," "Head Shake," "Forefinger Raise," "Forefinger Raise-Like," "Forefinger Wag," "Hand Wag," "Hands Scissor," and "Shoulder Shrug" (see [13] for details). We encoded each gesture in a binary manner, based on its presence at each of the 5700 total number of video frames, with each sequence ranging from 30 to 120 frames. The prosodic features were extracted with the publicly available software package OpenEar [14].

The database of pain we used was the UNBC-McMaster Shoulder Pain Expression Database[6] [15], which features 25 subjects/patients spontaneously expressing various levels of elicited pain in a total of 200 video sequences. The database was coded for, among others, pain level per sequence by expert observers on a 6-point scale from 0 (no pain) to 5 (extreme pain). Furthermore, each of the 48 398 video frames in the database was coded for each of the observable facial muscle movements–action units (AUs) according to the Facial Action Coding System (FACS) [16] by expert FACS coders. In our experiments, we encoded each of the possible 45 AUs in a binary manner, based on their presence. We labeled sequences coded with 0 as "no pain," sequences coded with 1–2 as "moderate pain," and those coded as 3–5 as "strong pain."

For our experiments, we compared the finite HCRFs to our iHCRF based on the F1 measure they achieved in each of the classification problems at hand. We evaluated the performance of the models on five different folds in the case of the Canal9 dataset (leave–2–debates–out for testing) and on 25 different folds in the case of the UNBC dataset (leave–1–subject–out for testing). In each case, we concatenated the predictions for every test sequence of each fold and calculated the F1 measure for each label. The measure we used was the average F1 over all labels. We ran both HCRF and iHCRF experiments with 10 random initializations, selecting the best trained model each time by examining the F1 measure achieved on a validation set consisting of the sequences from three debates in the case of the Canal9 dataset and from seven subjects in the case of the UNBC dataset. In every fold, our training, validation, and testing sets comprised not only unique sequences but also unique debates or subjects. In addition to the random initializations, the best HCRF model was also selected by experimenting with different number of hidden states and different values for the HCRF L2 regularization coefficient. Specifically, for each random initialization, we considered models with two, three, four, and five hidden states and a coefficient of 1, 10, and 100. This set of values for the hidden states was selected after preliminary results showed that a larger number of hidden states was unnecessary for all the problems considered. All our iHCRF models had their initial number of represented hidden states set to $K = 10$ and their beam numbers were sampled from a beta distribution with parameters $a = 1$ and $b = 2$. These were also chosen based on preliminary experiments, which showed that different initial values for K did not have a big impact on our model—see discussion on convergence of K below—and that these parameters for the beta distribution seemed the best for all four of our problems. Finally, our HCRF models were trained with a maximum of 300 iterations of the gradient ascent method used [1], whereas our iHCRF models were trained with 100 sampling iterations, and tested by considering 100 samples of optimal $\boldsymbol{\psi}^*$ values.

As one can see in Fig. 2, the beam sampler for iHCRF is able to quickly converge to a stable number of represented states within only 10 sampling iterations, regardless of the initialization for K. The final K ranged from two to three hidden states in the case of the three-label agreement and disagreement recognition task. This seems to be a correct choice, as evident from Fig. 2, which shows the average—over all labels—F1 achieved on the training and validation

[5]Publicly available at http://canal9-db.sspnet.eu/.

[6]Publicly available at http://www.pitt.edu/~jeffcohn/PainArchive/.

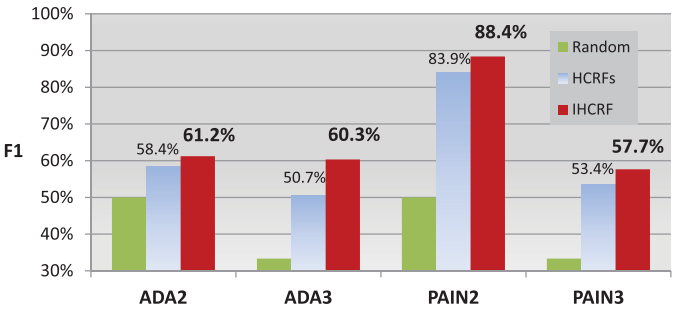

Fig. 3. F1 measure achieved by our iHCRF versus the best, in each fold of each problem, finite HCRF versus a random classifier. ADA2: Two-label classification for the Canal9 dataset of agreement and disagreement. ADA3: Three-label classification for the Canal9 dataset. PAIN2: Two-label classification for the UNBC dataset of shoulder pain. PAIN3: Three-label classification for the UNBC dataset.

TABLE I

DISTRIBUTION OF EXECUTION TIME. THE FIRST TWO COLUMNS SHOW THE AVERAGE TIME IT TOOK A MODEL TO TRAIN AND TEST FOR A GIVEN SET OF PARAMETERS. THE LAST TWO COLUMNS SHOW THE TOTAL TIME IT TOOK ALL HCRFS VERSUS ALL iHCRFS TO COMPLETE ALL THE EXPERIMENTS, INCLUDING CROSS-VALIDATION, AS OUTLINED IN THE TEXT EXCLUDING PRELIMINARY TESTS

Dataset	Mean HCRF	Mean iHCRF	All HCRF	All iHCRF
ADA2	71 s ($\sigma = 4$ s)	419 s ($\sigma = 103$ s)	11.1 h	3.5 h
ADA3	160 s ($\sigma = 98$ s)	399 s ($\sigma = 92$ s)	22.2 h	15.4 h
PAIN2	357 s ($\sigma = 143$ s)	864 s ($\sigma = 108$ s)	143.5 h	48.6 h
PAIN3	714 s ($\sigma = 236$ s)	1985 s ($\sigma = 209$ s)	273.7 h	95.5 h
Total	1302 s	3667 s	450.5 h	163 h

sets by the best HCRF models selected via cross-validation.

Fig. 3 shows the average—over all labels—F1 measure on the test sets for all four of our problems. Evidently, the iHCRF managed to achieve better results than the best HCRF models we could find in each case. Since the infinite model structure is not specified *a priori* but is instead determined from our data, the iHCRF model is more flexible and is able to achieve better performance. Another reason for the better performance may be the fact that the finite HCRF optimization function is not convex and the parameter learning process is therefore prone to get stuck into local optima.

Beam sampling for iHCRFs was able to train a typical iHCRF model for the two-label agreement versus disagreement classification task in 84 s. Under the same conditions, an HCRF model took 211 s to train. Testing can be slower for iHCRFs, making the average amount of combined training and testing time required of a single iHCRF model higher than the equivalent one of a single HCRF model, as evident from Table I. Note that this is the time reported for training one HCRF model; however, selecting an HCRF model requires training 12 different models, according to our experimental setup, in order to select the best one. As an indication of running times, we compared the time required for HCRF and iHCRF models to train and test for a run of all the experiments reported above. The experiments were all timed on the same computing cluster of identical machines, and the time reported by each experiment involved only one node processor and included the tasks of training on a given training set, choosing parameters based on performance on the given validation set, and testing on the respective test set. As one can see in Table I, the time spent by iHCRF to achieve performance higher than the HCRF was a staggering 287.5 h less, just 27% of the total experimentation time. Naturally, this would only check a very limited HCRF parameter subspace. On the contrary, the iHCRF is able to achieve better F1 performance in a more efficient way each time even in such difficult classification problems where the data to be analyzed is spontaneous human behavior as is manifested in audiovisual sequences. It is worth noting that the implementation we used for the HCRF was optimized and compiled code written in C++, whereas our iHCRF implementation was in MATLAB.

V. CONCLUSION

In this brief, we introduced the first discriminative non-parametric sequential model with latent variables, namely, the iHCRF. We also presented an efficient sampling technique that allows us to not only learn the model's hyperparameters and correct number of hidden states, but also to predict the label of a new observation sequence. We conducted experiments with four challenging tasks of classification of naturalistic human behavior. iHCRFs were able to quickly converge to the exact number of states, and to perform well in all problems. Our next step entails further experimentation with a variety of datasets and the examination of different approaches to learning for iHCRFs, e.g., variational inference [17].

REFERENCES

[1] A. Quattoni, S. Wang, L. Morency, M. Collins, and T. Darrell, "Hidden conditional random fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 10, pp. 1848–1852, Oct. 2007.

[2] A. Gunawardana, M. Mahajan, A. Acero, and J. Platt, "Hidden conditional random fields for phone classification," in *Proc. Interspeech*, vol. 2. 2005, pp. 1117–1120.

[3] K. Bousmalis, L.-P. Morency, and M. Pantic, "Modeling hidden dynamics of multimodal cues for spontaneous agreement and disagreement recognition," in *Proc. IEEE Int. Autom.Face Gesture Recognit. Workshop Conf.*, Mar. 2011, pp. 746–752.

[4] M. Beal, Z. Ghahramani, and C. Rasmussen, "The infinite hidden Markov model," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2002, pp. 577–584.

[5] J. V. Gael, Y. Saatci, Y. W. Teh, and Z. Ghahramani, "Beam sampling for the infinite hidden Markov model," in *Proc. Int. Conf. Mach. Learn.*, 2008, pp. 1088–1095.

[6] C. Rasmussen, "The infinite Gaussian mixture model," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2000.

[7] J. V. Gael, Y. Teh, and Z. Ghahramani, "The infinite factorial hidden Markov model," in *Advances in Neural Information Processing Systems*, vol. 21. Cambridge, MA: MIT Press, Mar. 2009, pp. 1697–1704.

[8] P. Orbanz and J. Buhmann, "Nonparametric Bayes image segmentation," *Int. J. Comput. Vis.*, vol. 77, nos. 1–3, pp. 25–45, 2008.

[9] S. Chatzis and G. Tsechpenakis, "The infinite hidden Markov random field model," *IEEE Trans. Neural Netw.*, vol. 21, no. 6, pp. 1004–1014, Jun. 2010.

[10] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Hierarchical dirichlet processes," *J. Amer. Stat. Assoc.*, vol. 101, no. 476, pp. 1566–1581, 2006.

[11] J. Sethuraman, "A constructive definition of dirichlet priors," *Stat. Sinica*, vol. 4, pp. 639–650, Oct. 1994.

[12] A. Vinciarelli, A. Dielmann, S. Favre, and H. Salamin, "Canal9: A database of political debates for analysis of social interactions," in *Proc. IEEE 3rd Int. Affect. Comput. Intell. Interact. Workshop Conf.*, Sep. 2009, pp. 96–99.

[13] K. Bousmalis, M. Mehu, and M. Pantic, "Spotting agreement and disagreement: A survey of nonverbal audiovisual cues and tools," in *Proc. IEEE 3rd Int. Affect. Comput. Intell. Interact. Workshop Conf.*, Sep. 2009, pp. 1–9.

[14] F. Eyben, M. Wöllmer, and B. Schuller, "OpenEAR—Introducing the Munich open-source emotion and affect recognition toolkit," in *Proc. IEEE 3rd Int. Affect. Comput. Intell. Interact. Workshop Conf.*, Sep. 2009, pp. 1–6.

[15] P. Lucey, J. Cohn, K. Prkachin, P. Solomon, and I. Matthews, "Painful data: The UNBC-Mcmaster shoulder pain expression archive database," in *Proc. IEEE Autom. Face Gesture Recognit. Workshop Conf.*, Mar. 2011, pp. 57–64.

[16] P. Ekman, W. V. Friesen, and J. C. Hager, *Facial Action Coding System*. Salt Lake City, UT: Research Nexus, 2002.

[17] D. Blei and M. Jordan, "Variational inference for Dirichlet process mixtures," *Bayesian Anal.*, vol. 1, no. 1, pp. 121–144, 2006.