Preprint of paper to appear in:
IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 2012

The Infinite-Order Conditional Random Field Model for Sequential Data Modeling

Sotirios P. Chatzis and Yiannis Demiris

Abstract—Sequential data labeling is a fundamental task in machine learning applications, with speech and natural language processing, activity recognition in video sequences, and biomedical data analysis being characteristic such examples, to name just a few. The conditional random field (CRF), a log-linear model representing the conditional distribution of the observation labels, is one of the most successful approaches for sequential data labeling and classification, and has lately received significant attention in machine learning, as it achieves superb prediction performance in a variety of scenarios. Nevertheless, existing CRF formulations can capture only one- or few-timestep interactions, and neglect higher-order dependencies, which are potentially useful in many real-life sequential data modeling applications. To resolve these issues, in this paper we introduce a novel CRF formulation, based on the postulation of an energy function which entails infinitely-long time-dependencies between the modeled data. Building blocks of our novel approach are: (i) the *sequence memoizer*, a recently proposed nonparametric Bayesian approach for modeling label sequences with infinitely-long time dependencies; and (b) a *mean-field-like approximation* of the model marginal likelihood, which allows for the derivation of computationally efficient inference algorithms for our model. The efficacy of the so-obtained infinite-order CRF (CRF$^\infty$) model is experimentally demonstrated.

Index Terms—Conditional random field, sequential data, sequence memoizer, mean-field principle.

1 Introduction

The problem of predicting from a set of observations a set of corresponding labels that are statistically correlated within some combinatorial structures like chains or lattices is of great importance, as it appears in a broad spectrum of application domains including annotating natural language sentences (e.g., parsing, chunking, named entity recognition), labeling biological sequences (e.g., protein secondary structure prediction), and classifying regions of images (e.g., image segmentation with object recognition), to name just a few.

Graphical models are a natural formalism for exploiting the dependence structure among entities. Traditionally, graphical models have been used to represent the joint probability distribution $p(\boldsymbol{y}, \boldsymbol{x})$, where the variables $\boldsymbol{y}$ represent the attributes of the entities that we wish to predict, and the variables $\boldsymbol{x}$ represent our observed knowledge about the entities. But modeling the joint distribution can lead to difficulties, because it requires modeling the distribution $p(\boldsymbol{x})$, which can include complex dependencies. Modeling these dependencies among inputs can lead to intractable models, but ignoring them can lead to reduced performance. A solution to this problem is to directly model the conditional distribution $p(\boldsymbol{y}|\boldsymbol{x})$, which is sufficient for classification. Indeed, this is the approach taken by conditional random fields (CRFs) [21].

A conditional random field is simply a log-linear model representing the conditional distribution $p(\boldsymbol{y}|\boldsymbol{x})$ with an associated graphical structure. Because the model is conditional, dependencies among the observed variables $\boldsymbol{x}$ do not need to be explicitly represented, affording the use of rich, global features of the input. For example, in natural language tasks, useful features include neighboring words and word bigrams, prefixes and suffixes, capitalization, membership in domain-specific lexicons, and semantic information from sources such as WordNet [36]. During the last years, we have witnessed an explosion of interest in CRFs, as it has managed to achieve superb prediction performance in a variety of scenarios, thus being one of the most successful approaches to the structured output prediction problem, with successful applications including text processing, bioinformatics, natural language processing, and computer vision [21], [13], [20], [27], [35], [45], [26], [29].

Despite their success, a significant issue that plagues current CRF formulations concerns their limited capability of capturing longer-term dynamics in the modeled datasets. Specifically, in order for CRFs to be computationally tractable, usual CRF formulations are limited to incorporate only pairwise potentials, giving rise to a first-order Markovianity assumption for the modeled data. Additionally, attempts to allow for higher-order potentials have been met with only limited success, due to the entailed computational burden incurred by their inference algorithms. Indeed, there are two main lines of research pertaining to making efficient inference possible in higher-order CRFs. The first involves developing new optimization algorithms through proposing new approximation techniques, such as the master-slave based decomposition process [19] and the compact

The authors are with the Department of Electrical and Electronic Engineering, Imperial College London. Exhibition Road, South Kensington Campus, SW7 2BT, London, UK. E-mail: {s.chatzis, y.demiris}@imperial.ac.uk.

transformation method [33], or generalizing widely used inference methods, such as belief propagation (BP) [22], [31], [9], and graph-cuts [3], [18]. The second line of research involves defining higher-order energies with some special restricted forms, which can be solved efficiently by application of popular optimization methods [16], [17].

In this paper, we focus on linear-chain CRFs; linear-chain CRFs, the basic probabilistic principle of which is illustrated in Fig. 1, are conditional probability distributions over label sequences which are conditioned on the observed sequences [21], [36]. Hence, in conventional linear-chain CRF formulations, an one-dimensional first-order Markov chain is assumed to represent the dependencies between the modeled data. In our work, we seek to provide a novel formulation of linear-chain CRF models that allows to capture dependencies in the modeled data over a theoretically infinitely-long time-window, that is essentially a *non-Markovian CRF model.* To achieve this goal, we introduce a novel form of energy functions for the linear-chain CRF model, based on the *sequence memoizer* (SM) [40], a nonparametric Bayesian method recently proposed for modeling sequential data with discrete values and dependencies over infinitely-long time-windows. As we show, inference for our model can be efficiently reduced to the forward-backward and Viterbi algorithms used in the case of simple first-order linear-chain CRF models, by utilizing an intricate approximation technique, based on the mean-field principle from statistical mechanics [4], [44]. We evaluate our novel approach in a number of sequential data modeling applications from diverse domains; as we show, our proposed approach offers considerable improvement over conventional first-order linear-chain CRFs, without any compromises in the entailed computational costs of the model inference algorithms.

The remainder of this paper is organized as follows: In Section 2, we provide the theoretical background of our method. Specifically, in Section 2.1, a brief introduction to CRFs is provided. In Section 2.2, we concisely present the sequence memoizer, a nonparametric Bayesian approach for modeling long-term dynamics in sequential data with discrete values. In Section 2.3, we present the mean-field principle from statistical mechanics, which comprises a key methodology we employ to derive our model. In Section 3, the proposed infinite-order conditional random field (CRF$^\infty$) model is introduced, and its inference algorithms are derived. In Section 4, we consider a number of applications of the CRF$^\infty$ model, with the aim to investigate whether coming up with a computationally tractable way of relaxing the Markovian assumption of linear-chain CRF models is of any significance for the sequential data classification algorithm when considering real-life datasets. Finally, in the concluding section of this paper, we summarize our work and discuss future possible directives in our line of research.

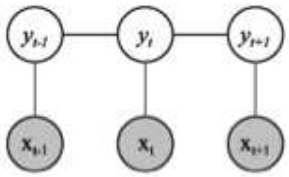

Figure 1. Linear-chain conditional random fields: An open node denotes a random variable, and a shaded node has been set to its observed value.

2 Theoretical Background

2.1 Conditional Random Fields

In the following, we provide a brief introduction to linear-chain CRF models, which constitute the main research theme of this paper. For a more detailed account of CRF models, the interested reader may refer to [36].

Linear-chain CRFs typically assume dependencies encoded in a left-to-right chain structure. Formally, linear-chain CRFs are defined in the following fashion: Let $\{\boldsymbol{x}_t\}_{t=1}^{T_X}$ be a sequence of observable random vectors, and $\{\boldsymbol{y}_t\}_{t=1}^{T_Y}$ be a sequence of random vectors that we wish to predict. Typically, the model is simplified by assuming that the lengths of the two sequences are equal, i.e. $T_X = T_Y = T$, and that the predictable variables are scalars defined on a vocabulary comprising K words, i.e. $y_t \in \mathcal{Y}$, with $\mathcal{Y} = \{1, \ldots, K\}$, whereas the observable variables are usually defined on a high-dimensional real space, $\boldsymbol{x}_t \in \mathcal{X}$, with $\mathcal{X} \subseteq \mathbb{R}^\zeta$. Then, introducing the notation $\boldsymbol{x} = \left([\boldsymbol{x}_t']_{t=1}^T\right)'$, and $\boldsymbol{y} = [y_t]_{t=1}^T$, a first-order linear-chain CRF defines the conditional probability for a label sequence $\boldsymbol{y}$ to be given by

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \exp\left[\sum_{t=2}^{T} \phi_t(y_t, y_{t-1}, \boldsymbol{x}_t) + \phi_1(y_1, \boldsymbol{x}_1)\right] \quad (1)$$

where $\phi_t(\cdot)$ is the local *potential* (or *score*) *function* of the model at time t, and $Z(\boldsymbol{x})$ is a partition function that ensures the conditional probability $p(\boldsymbol{y}|\boldsymbol{x})$ of a state sequence $\boldsymbol{y}$ will always sum to one

$$Z(\boldsymbol{x}) = \sum_{\boldsymbol{y}} \exp\left[\sum_{t=2}^{T} \phi_t(y_t, y_{t-1}, \boldsymbol{x}_t) + \phi_1(y_1, \boldsymbol{x}_1)\right] \quad (2)$$

In this work, we will be assuming that the potential functions of the postulated linear-chain CRFs can be written in the form

$$\phi_t(y_t, y_{t-1}, \boldsymbol{x}_t) = \phi_t^1(y_t, \boldsymbol{x}_t) + \phi_t^2(y_t, y_{t-1}, \boldsymbol{x}_t) \quad (3)$$

$$\phi_1(y_1, \boldsymbol{x}_1) = \phi_1^1(y_1, \boldsymbol{x}_1) + \phi_1^2(y_1, \boldsymbol{x}_1) \quad (4)$$

where the $\phi_t^1(y_t, \boldsymbol{x}_t)$ and the $\phi_t^2(y_t, y_{t-1}, \boldsymbol{x}_t)$ are the *unary* and *transition potentials* of the model, respectively, centered at the current time point.

Regarding the form of the unary and transition potentials usually selected in the literature, the most typical

selection consists in setting

$$\phi_t^1(y_t, \boldsymbol{x}_t) = \sum_{i=1}^{K} \delta(y_t - i)\boldsymbol{\omega}_i^1 \cdot \boldsymbol{x}_t \quad (5)$$

and

$$\phi_t^2(y_t, y_{t-1}, \boldsymbol{x}_t) = \sum_{i=1}^{K}\sum_{j=1}^{K} \delta(y_t - j)\delta(y_{t-1} - i)\boldsymbol{\omega}_{ij}^2 \cdot \boldsymbol{x}_t \quad (6)$$

with

$$\phi_1^2(y_1) = \sum_{i=1}^{K} \delta(y_t - i)\boldsymbol{\omega}_i^2 \cdot \boldsymbol{x}_t \quad (7)$$

where $\delta(\sigma)$ is the Dirac delta function, the parameters $\boldsymbol{\omega}_i^1$ are the prior weights of an observation emitted from state i, the parameters $\boldsymbol{\omega}_{ij}^2$ are related to the prior probabilities of the transition from state i to state j, and the parameters $\boldsymbol{\omega}_i^2$ are related to the prior probabilities of being at state i at the initial time point $t = 1$. Estimates of these parameters are obtained by means of model training, which consists in maximization of the log of the model likelihood, given by (1). For this purpose, usually quasi-Newton optimization methodologies are employed, such as the BFGS algorithm [2], or its limited memory variant (L-BFGS) [24]. Indeed, the likelihood function of this model is known to be of a convex form, which guarantees convergence to the global optimum [21], [36].

Note that computation of the model likelihood $p(\boldsymbol{y}|\boldsymbol{x})$ entails calculation of the sum $Z(\boldsymbol{x})$ defined in (2). This can be effected in a computationally efficient manner using the familiar forward-backward algorithm [32], [7], widely known from the HMM literature. Indeed, as discussed, e.g., in [36], it is easy to show that

$$Z(\boldsymbol{x}) = \sum_{j=1}^{K} \alpha_T(j) \quad (8)$$

where the $\alpha_T(j)$ is the forward probability of state j at the final time T. In the case of linear-chain CRF models, the forward probabilities $\alpha_t(j)$ are given by [36]

$$\alpha_t(j) = \sum_{i=1}^{K} \alpha_{t-1}(i)\exp\Big\{\phi_t(y_t = j, y_{t-1} = i, \boldsymbol{x}_t)\Big\},\ t \geq 2 \quad (9)$$

with initialization

$$\alpha_1(j) = \exp\Big\{\phi_1(y_1 = j, \boldsymbol{x}_1)\Big\} \quad (10)$$

Finally, inference under a linear-chain CRF model consists in determining the optimal sequence of labels $\hat{\boldsymbol{y}}$ given a sequence of observations $\boldsymbol{x}$, i.e.,

$$\hat{\boldsymbol{y}} = \underset{\boldsymbol{y}}{\operatorname{argmax}}\ \log p(\boldsymbol{y}|\boldsymbol{x}) \quad (11)$$

Solution of this problem can be again obtained in a computationally efficient fashion by employing a variant of the algorithms used to solve the familiar problem of sequence segmentation in the HMM literature, namely the Viterbi algorithm [32]. In the case of linear-chain CRFs, it can be shown that the Viterbi algorithm yields the following recursion [36]

$$\xi_t(j) \triangleq \max_{1\leq i\leq K} \exp\Big\{\phi_t(y_t = j, y_{t-1} = i, \boldsymbol{x}_t)\Big\}\xi_{t-1}(i) \quad (12)$$

with initialization

$$\xi_1(j) \triangleq \exp\Big\{\phi_1(y_1 = j, \boldsymbol{x}_1)\Big\} \quad (13)$$

based on which, output sequence optimization reads

$$\hat{y}_t = \underset{1\leq i\leq K}{\operatorname{argmax}}\ \xi_t(i) \quad (14)$$

2.2 The Sequence Memoizer

The sequence memoizer (SM) is a non-Markovian model for stationary discrete sequential data [40]. The model is non-Markovian in the sense that the next value in a sequence is modeled as being conditionally dependent on all the previous values in the same sequence. Formulation of the SM model is based on the provision of a specific parameterization of an unbounded-depth hierarchical Pitman-Yor process (HPYP) [38], [30]. In the following, we begin by briefly introducing the hierarchical Pitman-Yor process model. Further, we explain how the SM generalizes the HPYP model to allow for infinitely-long (or, better, infinitely-deep) model structures, and we provide a concise description of the inference algorithms of the sequence memoizer.

2.2.1 The hierarchical Pitman-Yor process

Let us consider a vocabulary $\mathcal{Y}$ comprising K words. For each word $y \in \mathcal{Y}$, let $G(y)$ be the (to be estimated) probability of y; let also $G = [G(y)]_{y\in\mathcal{Y}}$ be the vector of word probabilities. The Pitman-Yor process [30] is an appropriate prior that can be imposed over the vector of word probabilities G. We can write

$$G|d, \theta, G_0 \sim \mathrm{PY}(d, \theta, G_0) \quad (15)$$

where $d \in [0, 1)$ is the discount parameter of the process, $\theta > -d$ is its strength parameter, and $G_0 = [G_0(y)]_{y\in\mathcal{Y}}$ is its base distribution, expressing the a priori probability of a word y before any observation, usually set to $G_0(y) = \frac{1}{K}\ \forall y \in \mathcal{Y}$.

Now, let us consider a sequence of words $\{y_t\}_{t=1}^{T}$ drawn independently and identically (i.i.d.) from G

$$y_t|G \sim G, \quad t = 1, \ldots T \quad (16)$$

Integrating out G, the joint distribution of the variables $\{y_t\}_{t=1}^{T}$ can be shown to exhibit a clustering effect. Specifically, given the first $T - 1$ samples drawn i.i.d. from G, $\{y_t\}_{t=1}^{T-1}$, it can be shown that the new sample y_T is either (a) drawn from the base distribution G_0 with probability $\frac{\theta+dK}{\theta+T-1}$, or (b) is selected from the existing draws, according to a multinomial allocation, with

probabilities proportional to the number of the previous draws with the same allocation. In other words

$$p(y_T|\{y_t\}_{t=1}^{T-1}, d, \theta, G_0) = \frac{\theta + dK}{\theta + T - 1} G_0 + \sum_{c=1}^{K} \frac{f_c^{T-1} - d}{\theta + T - 1} \delta_c \quad (17)$$

where δ_c denotes the distribution concentrated at a single point (label) c, and f_c^{T-1} is the number of draws in the previous $T-1$ timesteps that turned out to be equal to c, that is

$$f_c^{T-1} = \{\#y_t, t = 1, \ldots, T-1 : \ y_t = c\} \quad (18)$$

The above generative procedure produces a sequence of words drawn i.i.d. from G, with G marginalized out. Notice the rich-gets-richer clustering property of the process: the more words have been assigned to a draw from G_0, the more likely subsequent words will be assigned to the draw. Further, the more we draw from G_0, the more likely a new word will again be assigned to a new draw from G_0. These two effects together produce a *power-law distribution* where many unique words are observed, most of them rarely [30]. In particular, for a vocabulary of unbounded size and for $d > 0$, the number of unique words scales as $\mathcal{O}(\theta T^d)$, where T is the total number of drawn words. Note also that, for $d = 0$, the Pitman-Yor process reduces to another famous method in the field of Bayesian nonparametrics, the Dirichlet process [10], in which case the number of unique words grows more slowly at $\mathcal{O}(\theta \log T)$ [38].

Despite the merits of the Pitman-Yor process, under the construction (17) the drawn words are always considered to be independent of each other. However, in practical applications, it is usually the case that a set of sequential observations are always closely interdependent, thus the i.i.d. assumption is clearly invalid. An nth order hierarchical Pitman-Yor process [38] resolves these issues by postulating a hierarchical model of the form

$$G_{\boldsymbol{u}} \sim \mathrm{PY}(d_{|\boldsymbol{u}|}, \theta_{|\boldsymbol{u}|}, G_{\pi(\boldsymbol{u})}) \quad (19)$$

where $\boldsymbol{u}$ is the context variable, denoting the set of the previously drawn (up to) n words, $G_{\boldsymbol{u}}(y)$ is the probability of the current word taking on the value y given that its context is $\boldsymbol{u}$, $G_{\boldsymbol{u}} = [G_{\boldsymbol{u}}(y)]_{y \in \mathcal{Y}}$ is the vector of probabilities of all the possible words $y \in \mathcal{Y}$ when the context is $\boldsymbol{u}$, and $\pi(\boldsymbol{u})$ is the prefix of $\boldsymbol{u}$ consisting of all but the latest word in $\boldsymbol{u}$. Note that the discount and strength parameters of the HPYP model are functions of the length $|\boldsymbol{u}|$ of the context (up to n), and not of the context $\boldsymbol{u}$ itself. Note also that the base distribution $G_{\pi(\boldsymbol{u})}$ in (19) is also unknown; for this reason, we recursively place a prior $G_{\pi(\boldsymbol{u})}$ over it using again the general expression (19), but now with parameters $\theta_{\pi(\boldsymbol{u})}$, $d_{\pi(\boldsymbol{u})}$, and $G_{\pi(\pi(\boldsymbol{u}))}$ instead. This recursion is repeated until we get to $G_\emptyset$, that is we reach an empty context, on which we place a simple Pitman-Yor process prior of the form

$$G_\emptyset \sim \mathrm{PY}(d_0, \theta_0, G_0) \quad (20)$$

where G_0 is a simple base distribution with $G_0(y) = \frac{1}{K} \ \forall y \in \mathcal{Y}$.

Based on this construction, the probability of drawing a word $y \in \mathcal{Y}$ when the (up to) nth order context is $\boldsymbol{u}$ yields [38]

$$G_{\boldsymbol{u}}(y) = \frac{c_{\boldsymbol{u}}(y) - d_{|\boldsymbol{u}|} t_{\boldsymbol{u}}(y)}{\theta_{|\boldsymbol{u}|} + c_{\boldsymbol{u}}} + \frac{\theta_{|\boldsymbol{u}|} + d_{|\boldsymbol{u}|} t_{\boldsymbol{u}}}{\theta_{|\boldsymbol{u}|} + c_{\boldsymbol{u}}} G_{\pi(\boldsymbol{u})}(y) \quad (21)$$

where $\mathcal{M}$ is the postulated HPYP model, $c_{\boldsymbol{u}}(y)$ is the *frequency of draws with the context being* $\boldsymbol{u}$ *that the drawn word was equal to* y, $t_{\boldsymbol{u}}(y)$ *is the frequency of draws with the context being* $\boldsymbol{u}$ *that the drawn word was equal to* y *and was drawn by recursion to the base distribution* $G_{\pi(\boldsymbol{u})}$, $c_{\boldsymbol{u}}$ *is the number of times the current context was* $\boldsymbol{u}$

$$c_{\boldsymbol{u}} = \sum_{y \in \mathcal{Y}} c_{\boldsymbol{u}}(y) \quad (22)$$

and, similar $t_{\boldsymbol{u}}$ *is the number of times the current context was* $\boldsymbol{u}$ *and recursion to the base distribution* $G_{\pi(\boldsymbol{u})}$ *was conducted*

$$t_{\boldsymbol{u}} = \sum_{y \in \mathcal{Y}} t_{\boldsymbol{u}}(y) \quad (23)$$

Inference for the HPYP model is conducted efficiently using a simple Gibbs sampling scheme, described, e.g., in [38]. The Gibbs sampling scheme for the HPYP model aims at obtaining *the posterior distributions over the* word arrangement *variables* $c_{\boldsymbol{u}}(y)$, $t_{\boldsymbol{u}}(y)$, $c_{\boldsymbol{u}}$, *and* $t_{\boldsymbol{u}}$, *as well as the model discount and strength parameters,* d_l *and* θ_l. Given a sample from the posterior of the arrangement variables and model parameters, the predictive distribution of the next symbol y given its context $\boldsymbol{u}$ is defined as the next draw from the distribution $G_{\boldsymbol{u}}(y)$, and is given by recursively applying (21).

2.2.2 The sequence memoizer as an unbounded-depth HPYP model

The sequence memoizer is basically an unbounded-depth HPYP model. Specifically, the sequence memoizer is based on the postulation of a HPYP model of the form (19), with the maximum length n of its context variables $\boldsymbol{u}$ taken as tending to infinity, i.e., $n \to \infty$. That is, we consider that the distribution of a word is dependent on all the previously drawn words, thus a non-Markovian model is obtained. The sequence memoizer is not Markovian in the sense that the next value in a sequence is modeled as being conditionally dependent on all previous values in the sequence.

As is obvious, inference in such an unbounded-depth HPYP model might entail a large number of recursions of the form (21), a fact that could possibly give rise to prohibitive computational costs for the model inference algorithms when the length of the drawn sequences increases considerably. To constrain the learning of these latent variables, a special hierarchical Bayesian prior based on Pitman-Yor processes can be employed, which

promotes sharing of statistical strength between subsequent symbol predictive distributions for equivalent contexts of different lengths [38]. Specifically, in this work, as a way of mitigating these issues, we exploit the following result [39]:

Theorem 1. *Let us consider a single path in a graphical model $G_1 \rightarrow G_2 \rightarrow G_3$ with G_2 having no children other than G_3. Then, if $G_2|G_1 \sim \mathrm{PY}(d_1, 0, G_1)$ and $G_3|G_2 \sim \mathrm{PY}(d_2, 0, G_2)$, it holds $G_3|G_1 \sim \mathrm{PY}(d_1 d_2, 0, G_1)$ with G_2 marginalized out.*

Thus, the sequence memoizers employed in this work comprise a HPYP model with its strength parameters θ_n set equal to zero, $\theta_n = 0,\ \forall n$ (note though that a wider family of distributions can be also considered for similar computational efficiency purposes [11]). Then, given a context $\boldsymbol{u}$, the sequence memoizer determines which of its prefixes i.e., $\pi(\boldsymbol{u}), \pi(\pi(\boldsymbol{u}))$, and so on, has no other children than the one appearing within $\boldsymbol{u}$, and removes them from the recursions in (19), based on Theorem 1. This way, the computational complexity of the SM model inference algorithms, *which are otherwise identical to those of the HPYP model*, can be considerably reduced in cases of long drawn sequences, without compromises in the model's efficacy.

Once the collapsed graphical model representation for the sequence memoizer has been built, inference proceeds as it would for any conventional hierarchical Pitman-Yor process model, and consists in determining *the posterior distributions over the model discount parameters d_l, and word arrangement variables.* For this purpose, in this paper we use the Gibbs sampler proposed in [38], as described in the previous section.

At test time t, inference consists in using the sequence memoizer to compute the probability $q(y_t|\{y_\tau\}_{\tau=1}^{t-1})$ of the modeled variable being equal to the symbol y_t, given a context $\boldsymbol{u} = \{y_\tau\}_{\tau=1}^{t-1}$. Similar to the discussions of the previous section, the predictive probability of the sequence memoizer is taken as the posterior expectation of the distribution $G_{\boldsymbol{u}}(y_t)$ of the current word taking on the value y_t, given that its context is $\boldsymbol{u}$, i.e.

$$q(y_t|\{y_\tau\}_{\tau=1}^{t-1}) \triangleq \mathbb{E}\left[G_{\boldsymbol{u}}(y_t)\right] \tag{24}$$

where the distribution of $G_{\boldsymbol{u}}(y_t)$ is given by (21), and $\boldsymbol{u} \triangleq \{y_\tau\}_{\tau=1}^{t-1}$.

2.3 The mean-field principle

The mean-field principle is originally a method of approximation for the computation of the mean of a Markov random field. It comes from statistical mechanics (e.g. [5]) where it has been used as an analysis tool to study phase transition phenomena. More recently, it has been used in computer vision applications (e.g. [12], [43], [42]), graphical models (e.g. [15], and references therein) and other areas (e.g. [14]). It can also be used to provide an approximation of the distribution of a Markov random field [6], [8]. The basic idea of the mean-field principle consists in neglecting the fluctuations of the variables interacting with a considered variable. As a result of this assumption, the resulting system behaves as one composed of independent variables for which computation gets tractable.

More specifically, let us consider a set of interdependent variables $\{y_t\}_{t=1}^T$ that define a Markov random field with a specified neighborhood system. For example, a neighborhood system of first-order sequential nature may be considered, in which case the postulated Markov random field reduces to a first-order Markov chain. Under the mean-field principle, the joint distribution $p(\{y_t\}_{t=1}^T)$ is approximated by the product

$$p(\{y_t\}_{t=1}^T) \approx \prod_{t=1}^{T} \hat{p}_t(y_t) \tag{25}$$

Here, the $\hat{p}_t(y_t)$ is an approximation of the marginal distribution $p(y_t)$ of the field at the site (e.g., time point) t. This latter quantity is expressed under the mean-field principle in the following conditional form

$$\hat{p}_t(y_t) \approx p(y_t|\{\hat{y}_\tau\}_{\tau \in \mathcal{N}(t)}) \tag{26}$$

where $\hat{y}_\tau$ is the expected value of y_τ, i.e.,

$$\hat{y}_\tau = \mathbb{E}[y_\tau] \tag{27}$$

and $\mathcal{N}(t)$ is the set of neighbors of site t. For example, in the case of a first-order sequential nature neighborhood system, (26) yields

$$\hat{p}_t(y_t) \approx p(y_t|\{\hat{y}_\tau\}_{\tau=1}^{t-1}) \tag{28}$$

In other words, under the mean-field principle, the marginal distribution of a Markov random field at a given site (e.g., time point) is expressed as the distribution of the observed variable at the considered site conditional on the expected values of the variables at all the sites interacting with the considered site. As such, we effectively neglect fluctuations from the mean in the neighborhood system of each site (e.g., time point).

More generally, we talk about *mean field-like approximations* when the value of a variable observed at a site t is considered independent of the fluctuations of the values at other sites in its neighborhood, which are all set to constants ***(not necessarily their means),*** independently of the value at site t. This idea is applied to alleviate the computational burden when dealing with complex joint distributions in a multitude of applications in computer science (e.g., [6], [8], [15]).

3 Proposed Approach

As already discussed, in this paper we aim to introduce an infinite-order (non-Markovian) linear-chain conditional random field model for sequential data modeling. That is, we seek to derive a discriminative model, associating a sequence of multivariate observations $\boldsymbol{x}$ with

a sequence of corresponding labels $\boldsymbol{y}$ by means of the distribution

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})}\exp\left[\sum_{t=2}^{T}\phi_t(y_t, \{y_\tau\}_{\tau=1}^{t-1}, \boldsymbol{x}_t) + \phi_1(y_1, \boldsymbol{x}_1)\right] \tag{29}$$

where the partition function is given by

$$Z(\boldsymbol{x}) = \sum_{\boldsymbol{y}}\exp\left[\sum_{t=2}^{T}\phi_t(y_t, \{y_\tau\}_{\tau=1}^{t-1}, \boldsymbol{x}_t) + \phi_1(y_1, \boldsymbol{x}_1)\right] \tag{30}$$

and the potential functions of the postulated linear-chain CRFs are of the form

$$\phi_t(y_t, \{y_\tau\}_{\tau=1}^{t-1}, \boldsymbol{x}_t) = \phi_t^1(y_t, \boldsymbol{x}_t) + \phi_t^2(y_t, \{y_\tau\}_{\tau=1}^{t-1}, \boldsymbol{x}_t) \tag{31}$$

$$\phi_1(y_1, \boldsymbol{x}_1) = \phi_1^1(y_1, \boldsymbol{x}_1) + \phi_1^2(y_1, \boldsymbol{x}_1) \tag{32}$$

To obtain such a model, we need to determine a suitable functional form for the *transition potentials* $\phi_t^2(y_t, \{y_\tau\}_{\tau=1}^{t-1}, \boldsymbol{x}_t)$.

Let us consider that the label variables y take values on a finite set $\mathcal{Y} = \{1, \ldots, K\}$. Based on the discussions of Section 2.2, the desired form of the transition potentials of our model can be obtained by "training" (obtaining the posterior distributions of) a sequence memoizer model. For this purpose, we use the Gibbs sampler of [38], and a dataset of observed training label sequences.

Subsequently, we let the transition potentials of the sought CRF model be given by the log predictive probability $q(y_t|\{y_\tau)_{\tau=1}^{t-1})$, obtained by the sequence memoizer as described in Section 2.2, multiplied by a term accounting for the value of the observed variable $\boldsymbol{x}_t$, i.e.

$$\phi_t^2(y_t, \{y_\tau\}_{\tau=1}^{t-1}, \boldsymbol{x}_t) \triangleq \left[\boldsymbol{\gamma}_{y_t} \cdot \boldsymbol{x}_t\right] \log q(y_t|\{y_\tau\}_{\tau=1}^{t-1}) \tag{33}$$

where $q(y_t|\{y_\tau\}_{\tau=1}^{t-1})$ is given by (24), and the $\{\boldsymbol{\gamma}_i\}_{i=1}^{K}$ are model parameters estimated through model training.

This selection allows for computationally efficient modeling of the dependencies between sequentially appearing label data over infinitely-long time-windows. Indeed, having used a sequence memoizer to learn the dependencies between successive observed labels y, we obtain a CRF model which takes into account the whole history of label ($y \in \mathcal{Y}$) observations in making predictions, thus of infinite-order nature.

Definition 2. *A linear-chain CRF with conditional probability $p(\boldsymbol{y}|\boldsymbol{x})$ given by (29), and associated transition potentials defined over infinite-length time-windows, expressed as the scaled log-predictive probabilities of a sequence memoizer, as in (33), shall be denoted as the infinite-order CRF (CRF$^\infty$) model for sequential data modeling.*

The CRF$^\infty$ model entails transition potential functions which take into account the whole history of past observed labels, thus giving rise to a linear-chain CRF model that postulates infinite-order time-dependencies. Note also that the unary potentials of our model are defined similar to the case of a first-order linear-chain CRF model, reading

$$\phi_t^1(y_t, \boldsymbol{x}_t) = \boldsymbol{\omega}_{y_t} \cdot \boldsymbol{x}_t \tag{34}$$

where the $\{\boldsymbol{\omega}_i\}_{i=1}^{K}$ are model parameters estimated through model training.

Having defined the proposed CRF$^\infty$ model, we can now proceed to the derivation of its training and sequence decoding (inference) algorithms.

3.1 Model Training

Training the CRF$^\infty$ model comprises two separate procedures:

1) *"Training" of the sequence memoizer* used to obtain the state transition potentials of our model. This procedure essentially consists in using the Gibbs sampler of [38], and a set of observed sequences of labels $y \in \mathcal{Y}$, to obtain the posterior distributions of the sequence memoizer used in the expression (33) of our model.
2) *Estimation of the log-linear parameters $\boldsymbol{\omega}_i$ (from the unary potentials), and $\boldsymbol{\gamma}_i$ (from the transition potentials)* of our model. This is performed by considering the sequence memoizer in (33) as a fixed distribution, and using a separate training set (sequences of pairs of observed inputs $\boldsymbol{x} \in \mathcal{X}$ and their corresponding labels $y \in \mathcal{Y}$) to estimate the log-linear parameters of the model. In the remainder of this section, we concentrate on this latter procedure.

Let us consider a training set $\mathcal{D} = \{(\boldsymbol{x}^n, \boldsymbol{y}^n)\}_{n=1}^{N}$, where $\boldsymbol{x}^n = \{\boldsymbol{x}_t^n\}_{t=1}^{T}$, $\boldsymbol{x}_t^n \in \mathcal{X}$, and $\boldsymbol{y}^n = \{\boldsymbol{y}_t^n\}_{t=1}^{T}$, $\boldsymbol{y}_t^n \in \mathcal{Y}$. To obtain point-estimates of the parameters $\boldsymbol{\omega}_i$ and $\boldsymbol{\gamma}_i$, $i \in \{1, \ldots, K\}$, given the dataset $\mathcal{D}$, we need to optimize the log-likelihood function of the CRF$^\infty$ model, reading

$$\log p(\boldsymbol{y}|\boldsymbol{x}) = \sum_{n=1}^{N}\Bigg\{\sum_{t=2}^{T}\phi_t(y_t^n, \{y_\tau^n\}_{\tau=1}^{t-1}, \boldsymbol{x}_t^n) + \phi_1(y_1^n, \boldsymbol{x}_1^n) - \log Z(\boldsymbol{x}^n)\Bigg\} \tag{35}$$

As we observe, estimation of the model log-likelihood (35) requires calculation of the quantities $Z(\boldsymbol{x}^n)$, reading

$$Z(\boldsymbol{x}^n) = \sum_{\boldsymbol{y}}\exp\left[\sum_{t=2}^{T}\phi_t(y_t, \{y_\tau\}_{\tau=1}^{t-1}, \boldsymbol{x}_t^n) + \phi_1(y_1, \boldsymbol{x}_1^n)\right] \tag{36}$$

It is clear that this quantity is not computationally tractable in most real-world scenarios. Therefore, an approximation to the expression of the partition function $Z(\boldsymbol{x}^n)$ is needed. For this purpose, we employ a mean-

field-like approximation: From (36), we have

$$\begin{aligned} Z(\boldsymbol{x}^n) &= \sum_{\boldsymbol{y}} \exp\left[\phi_1(y_1, \boldsymbol{x}_1^n)\right] \prod_{t=2}^{T} \exp\left[\phi_t(y_t, \{y_\tau\}_{\tau=1}^{t-1}, \boldsymbol{x}_t^n)\right] \\ &\approx \sum_{\boldsymbol{y}} \exp\left[\phi_1(y_1, \boldsymbol{x}_1^n)\right] \prod_{t=2}^{T} \exp\left[\phi_t(y_t, y_{t-1}, \{y_\tau^n\}_{\tau=1}^{t-2}, \boldsymbol{x}_t^n)\right] \\ &= \sum_{i=1}^{K} \sum_{j=1}^{K} \exp\Bigg[\sum_{t=2}^{T} \phi_t(y_t = i, y_{t-1} = j, \{y_\tau^n\}_{\tau=1}^{t-2}, \boldsymbol{x}_t^n) \\ &\quad + \phi_1(y_1 = i, \boldsymbol{x}_1^n) \Bigg] \end{aligned} \tag{37}$$

In other words, in calculating $Z(\boldsymbol{x}^n)$, we assume that in each term $\phi_t(y_t, \{y_\tau\}_{\tau=1}^{t-1}, \boldsymbol{x}_t^n)$, $\forall t \geq 2$, the variables $\{y_\tau\}_{\tau=1}^{t-2}$ do not fluctuate with y_t and y_{t-1}, but, rather, they are constants equal to some known (here, their observed) value $\{y_\tau^n\}_{\tau=1}^{t-2}$. This is in essence a mean-field-like approximation of $Z(\boldsymbol{x}^n)$, which allows for computing the partition function of the infinite-order CRF$^\infty$ model using a computationally efficient algorithm, very similar to the method used in the case of first-order CRFs, that is the *forward recursions algorithm*.

The forward probabilities in the case of the CRF$^\infty$ model, with partition function approximate in the form (37), read

$$\alpha_t^n(j) = \sum_{i=1}^{K} \alpha_{t-1}^n(i) \exp\Big\{ \phi_t(y_t = j, y_{t-1} = i, \{y_\tau^n\}_{\tau=1}^{t-2}, \boldsymbol{x}_t^n) \Big\} \qquad t \geq 2 \tag{38}$$

with initialization

$$\alpha_1^n(j) = \exp\Big\{ \phi_1(y_1 = j, \boldsymbol{x}_1^n) \Big\} \tag{39}$$

yielding

$$Z(\boldsymbol{x}^n) = \sum_{j=1}^{K} \alpha_T^n(j) \tag{40}$$

Note that the employed mean-field-like approximation does not constitute an assumption of the CRF$^\infty$ model itself, but is only applied to obtain a computationally tractable expression for the normalization constant $Z(\boldsymbol{x}^n)$ of the model, which is necessary in order to estimate the parameters $\boldsymbol{\omega}_i$ and $\boldsymbol{\gamma}_i$. In other words, the transition potentials of the CRF$^\infty$ model are still computed using the *whole history of observed labels* $\{y_\tau\}_{\tau=1}^{t-1}$, a computation made possible by exploiting the sequence memoizer. Hence, the proposed approximation definitely affects the eventual quality of the obtained estimates of $\boldsymbol{\omega}_i$ and $\boldsymbol{\gamma}_i$, but it does not by any means alter the assumptions of the model itself, which continues to postulate infinite-order transition potentials, taking into account the *whole history of observed labels* $\{y_\tau\}_{\tau=1}^{t-1}$, with no truncations imposed in that respect. The truncation only occurs in computing the terms $\phi_t(y_t, \{y_\tau\}_{\tau=1}^{t-1}, \boldsymbol{x}_t^n)$, and it consists in only truncating **the fluctuation** of the values $\{y_\tau\}_{\tau=1}^{t-2}$, by setting them to a constant.

Having obtained a computationally tractable expression for the partition functions $Z(\boldsymbol{x})$, we can now proceed to estimation of the parameters $\boldsymbol{\omega}_i$ and $\boldsymbol{\gamma}_i$, $i \in \{1, \ldots, K\}$, of the model. To effect this procedure, we resort to maximization of the log-likelihood $\log p(\boldsymbol{y}|\boldsymbol{x})$ of the model over each one of them, by means of an iterative maximization algorithm; the L-BFGS algorithm [24], and the scaled conjugate gradient (SCG) descent algorithm are two approaches suitable for this purpose. We shall evaluate both of them in the experimental section of this paper.

3.2 Sequence Decoding

Inference in linear-chain CRF models, also referred to as sequence decoding, consists in the optimization problem

$$\hat{\boldsymbol{y}} = \operatorname*{argmax}_{\boldsymbol{y}} \log p(\boldsymbol{y}|\boldsymbol{x}) \tag{41}$$

That is, we want to obtain a labeling $\boldsymbol{y}$ of the sequence of observed data $\boldsymbol{x}$ that maximizes model likelihood. In the case of the CRF$^\infty$ model, the log-likelihood of the model reads

$$\begin{aligned} \log p(\boldsymbol{y}|\boldsymbol{x}) \propto \sum_{t=1}^{T} \Bigg\{ & \phi_t^1(y_t, \boldsymbol{x}_t) \\ & + \left[\boldsymbol{\gamma}_{y_t} \cdot \boldsymbol{x}_t\right] \log q(y_t | \{y_\tau\}_{\tau=1}^{t-1}) \Bigg\} \end{aligned} \tag{42}$$

Therefore, in a fashion similar to first-order linear-chain CRF models, (41) turns out to be a dynamic programming problem, with backwards recursion

$$\hat{y}_t = \operatorname*{argmax}_{1 \leq i \leq K} \big\{ \xi_t(i) \big\} \tag{43}$$

where the cost function $\xi_t(i)$ is defined as

$$\begin{aligned} \xi_t(i) \triangleq \max_{\{y_\tau\}_{\tau=1}^{t-1}} \Bigg\{ & \xi_{t-1}(y_{t-1}) \\ & + \left[\boldsymbol{\gamma}_i \cdot \boldsymbol{x}_t\right] \log q(y_t = i | \{y_\tau\}_{\tau=1}^{t-1}) \Bigg\} \\ & + \phi_t^1(y_t = i, \boldsymbol{x}_t) \end{aligned} \tag{44}$$

with initialization

$$\xi_1(i) \triangleq \phi_1(y_1 = i, \boldsymbol{x}_1) \tag{45}$$

As we observe from the definition of the cost function (44), the obtained dynamic programming problem entails a large (theoretically infinite) number of variables over which $\xi_t(i)$ gets optimized. As such, the incurred computational costs might become prohibitive in most real-world scenarios. For this reason, we need to come up with an approximate solution with bounded (worst-case) computational costs. We resort to a mean-field-like approximation.

Specifically, we propose the following approximation: Let us begin with the second time step, $t = 2$. The cost function $\xi_t(i)$ reads

$$\begin{aligned} \xi_2(i) = \max_{y_1} \Big\{ & \xi_1(y_1) + \left[\boldsymbol{\gamma}_i \cdot \boldsymbol{x}_2\right] \log q(y_2 = i | y_1) \Big\} \\ & + \phi_2^1(y_2 = i, \boldsymbol{x}_2) \end{aligned} \tag{46}$$

Let us now continue to the next time-step, $t = 3$. The cost function $\xi_t(i)$ now reads

$$\xi_3(i) = \max_{\{y_\tau\}_{\tau=1}^{2}} \left\{ \xi_2(y_2) + [\boldsymbol{\gamma}_i \cdot \boldsymbol{x}_3] \log q(y_3 = i | \{y_\tau\}_{\tau=1}^{2}) \right\} + \phi_3^1(y_3 = i, \boldsymbol{x}_3) \tag{47}$$

At this point, we make the following key-hypothesis: We assume that, in $\xi_3(i)$, the variable y_1 does not fluctuate with y_2 and y_3, but, instead, it takes on a constant ("optimal") value $\hat{y}_1$. This way, the expression of $\xi_3(i)$ reduces to

$$\xi_3(i) \approx \max_{y_2} \left\{ \xi_2(y_2) + [\boldsymbol{\gamma}_i \cdot \boldsymbol{x}_3] \log q(y_3 = i | y_2, \hat{y}_1) \right\} + \phi_3^1(y_3 = i, \boldsymbol{x}_3) \tag{48}$$

This assumption is in essence a mean-field-like approximation of $\xi_3(i)$. Similar, for $t = 4$ we have

$$\xi_4(i) = \max_{\{y_\tau\}_{\tau=1}^{3}} \left\{ \xi_3(y_3) + [\boldsymbol{\gamma}_i \cdot \boldsymbol{x}_4] \log q(y_4 = i | \{y_\tau\}_{\tau=1}^{3}) \right\} + \phi_4^1(y_4 = i, \boldsymbol{x}_4) \tag{49}$$

which, under a mean-field-like approximation, neglecting the fluctuations of y_1 and y_2, yields

$$\xi_4(i) \approx \max_{y_3} \left\{ \xi_3(y_3) + [\boldsymbol{\gamma}_i \cdot \boldsymbol{x}_4] \log q(y_4 = i | y_3, \hat{y}_2, \hat{y}_1) \right\} + \phi_4^1(y_4 = i, \boldsymbol{x}_4) \tag{50}$$

This way, we eventually reduce the dynamic programming problem with cost function given by (44), to a simpler problem with bounded worst-case computational costs, where the cost function is defined as

$$\xi_t(i) \approx \max_{1 \leq j \leq K} \left\{ \xi_{t-1}(j) + \log q(y_t = i | y_{t-1} = j, \{\hat{y}_\tau\}_{\tau=1}^{t-2}) \right\} + \phi_t^1(y_t = i, \boldsymbol{x}_t) \tag{51}$$

with the same initialization (45), and backwards recursion (43). This construction gives, in turn, rise to another issue: what is the appropriate selection of the values $\{\hat{y}_\tau\}_{\tau=1}^{t-2}$? Following the relevant literature (e.g., [6], [8], [15], [43]), the values $\{\hat{y}_\tau\}_{\tau=1}^{t-2}$ may be selected as the values of $\{y_\tau\}_{\tau=1}^{t-2}$ that optimize some criterion.

In this work, the values of $\{\hat{y}_\tau\}_{\tau=1}^{t-2}$ are obtained in the following manner:

1) First, we postulate a simple first-order linear-chain CRF for the same problem, trained on the same data as the CRF$^\infty$ model. We use this model to obtain an initial optimal value $\hat{\boldsymbol{y}}$ by means of the Viterbi algorithm.
2) Using this initial optimizer $\hat{\boldsymbol{y}}$, we run the dynamic programing recursions (51) of the CRF$^\infty$ inference (Viterbi-like) algorithm. This way, a new sequence segmentation estimate $\hat{\boldsymbol{y}}$ is derived.
3) Further, we repeat the CRF$^\infty$ inference algorithm as many times as needed for the estimate of $\boldsymbol{y}$ to converge, each time using the latest obtained estimate $\hat{\boldsymbol{y}}$ for the purposes of the mean-field-like approximation.

Algorithm 1 Sequence Decoding Algorithm for the CRF$^\infty$ model.

Let us consider an observed sequence $\boldsymbol{x}$. We want to obtain the corresponding optimal state sequence $\hat{\boldsymbol{y}}$ with respect to a trained CRF$^\infty$ model. The proposed algorithm comprises the following steps:

1) Obtain an initial estimate of $\hat{\boldsymbol{y}}$ by means of a simple linear-chain CRF model.
2) Run the dynamic programming algorithm with backward recursion (43), and cost function (51) with initialization (45).
3) If the estimate of $\hat{\boldsymbol{y}}$ converges, or a maximum number of iterations has been reached, **exit;** otherwise, **return** to **step 2.**

At this point, we would like to emphasize that, again, the mean-field-like approximation is not applied to the core assumptions of the CRF$^\infty$ itself. As we observe in Eqs. (46)-(51), the proposed dynamic programming algorithm for CRF$^\infty$ model inference entails "full" transition potentials, that is potentials taking into account the *whole history of observed labels* $\{y_\tau\}_{\tau=1}^{t-1}$, with no truncations imposed in that respect. Therefore, the application of the mean-field-like approximation does not affect the infinite-order nature of its transition potentials; it only consists in considering in each term $\xi_t(i)$ that the variables $\{y_\tau\}_{\tau=1}^{t-2}$ do not fluctuate with the y_t and y_{t-1}, but, instead, they take on some constant values.

Note also that, additionally, and in an attempt to counterbalance the effect of this approximation, we repeat the CRF$^\infty$ model inference algorithm multiple times, with the estimates $\hat{\boldsymbol{y}}$ of $\boldsymbol{y}$ set equal to the outcome of the CRF$^\infty$ model inference algorithm on the previous execution, until $\boldsymbol{y}$ convergence. Apparently, we do not expect the segmentation $\boldsymbol{y}$ eventually obtained by this approximate procedure to be identical to the result of the original dynamic programming problem with iteration (44); we do expect though that a good trade-off between computational complexity and segmentation quality is indeed obtained, for a model which still retains its infinite-order nature (in the sense of how its transition potentials are computed).

An outline of the proposed sequence decoding algorithm for the CRF$^\infty$ model is provided in Alg. 1.

4 Experiments

In the following, we experimentally evaluate the performance of the CRF$^\infty$ model, considering a number of applications from diverse domains, with the aim to investigate the practical significance of coming up with a computationally tractable way of relaxing the Markovian assumptions of existing linear-chain CRF models.

To ensure the objectivity of our findings, we consider obtaining the estimates of the γ_i and ω_i parameters of the CRF$^\infty$ model using a number of alternative approaches to optimize the model likelihood, namely SCG algorithm, L-BFGS algorithm, and entropy maximization (EnM), as suggested in [34]. The sequence memoizers used to obtain the transition potentials of the model are "trained" by means of Gibbs sampling, as suggested in [40]. The implementation of this Gibbs sampler was taken from the sequence memoizer software available at: http://www.sequencememoizer.com/. Our source codes were developed in MATLAB, and made partial use of software provided by Neil Lawrence [23].

We compare our model's performance to standard first-order linear-chain CRF models, and the moderate order CRFs of [41]. As a baseline for comparisons in our experiments, we consider hidden Markov models (HMMs) with diagonal covariance Gaussian mixture observation densities, trained using the expectation-maximization (EM) algorithm [28]. To ensure the transparency of our results, in all our experiments we use publicly available benchmark datasets.

4.1 Video sequences segmentation

Here, we consider application of the CRF$^\infty$ model to segment video sequences from the CMU motion capture dataset [1]. As input variables to the evaluated algorithms we use the whole available set of joint angle values, while the output variables are the activity (class) labels assigned to each video frame. We consider three different experimental cases, namely *3-step climbing*, *skateboard: stop and go*, and *skateboard: push and turn*. In each one of these cases, multiple different videos of each movement are used in order to perform leave-one-out cross validation. We also provide the p-metric value of the Student's-t test run on the pairs of performances of the models (CRF, CRF$^\infty$), (moderate order CRF, CRF$^\infty$), and (HMM, CRF$^\infty$). For simplicity, to make these computations of the p-metric, we use the results obtained by the CRF and CRF$^\infty$ models trained by means of the L-BFGS algorithm. The Student's-t test allows to assess the statistical significance of the performance difference between two evaluated methods, given a set of performance measurements. Generated p-values of the Student's-t test below 0.05 strongly indicate that the means of the obtained performance statistics of the two methods provide a very good assessment of their actual performance difference.

4.1.1 3-step climbing

In this experimental case, we deal with videos depicting a human subject ascending a short ladder, stepping on a table, making a U-turn on the table, and descending the ladder. In Fig. 2, we provide few characteristic frames from one of the videos used in our experiments. The aim is to train the evaluated models so as to be capable of segmenting the videos into three subsequences: (i) ladder ascending; (ii) making U-turn; (iii) ladder descending. Four different videos, comprising 200-276 frames, from the same subject are used in our experiments to perform (4-fold) leave-one-out cross validation. The obtained performance statistics of the evaluated algorithms are provided in Table 1 (means and standard deviations over the conducted folds).

Table 1
Activity-based segmentation of 3-step climbing videos: Error rates obtained by the evaluated methods.

Method	Error Rate (%)	p-value
CRF (SCG)	29.18 $\pm$ 1.91	10^{-9}
CRF (L-BFGS)	29.12 $\pm$ 1.28	10^{-9}
CRF (EnM)	29.27 $\pm$ 1.22	10^{-9}
CRF$^\infty$ (SCG)	26.41 $\pm$ 1.82	
CRF$^\infty$ (L-BFGS)	26.38 $\pm$ 1.79	
CRF$^\infty$ (EnM)	25.93 $\pm$ 1.73	
Moderate Order CRF (3rd Order)	27.67 $\pm$ 2.08	10^{-9}
Moderate Order CRF (5th Order)	27.03 $\pm$ 2.15	10^{-9}
HMM (M=8)	30.49 $\pm$ 6.67	10^{-12}

Table 2
Activity-based segmentation of skateboard: stop and go videos: Error rates obtained by the evaluated methods.

Method	Error Rate (%)	p-value
CRF (SCG)	12.03 $\pm$ 0.27	10^{-6}
CRF (L-BFGS)	12.61 $\pm$ 0.29	10^{-6}
CRF (EnM)	11.97 $\pm$ 0.30	10^{-6}
CRF$^\infty$ (SCG)	8.70 $\pm$ 0.18	
CRF$^\infty$ (L-BFGS)	9.12 $\pm$ 0.16	
CRF$^\infty$ (EnM)	8.65 $\pm$ 0.20	
Moderate Order CRF (3rd Order)	11.51 $\pm$ 0.22	10^{-6}
Moderate Order CRF (5th Order)	11.48 $\pm$ 0.21	10^{-6}
HMM (M=8)	47.26 $\pm$ 2.43	10^{-9}

As we notice, the CRF$^\infty$ model improves considerably the obtained error rate over its first-order Markovian counterpart. Further, we also observe that the moderate order CRF models of [41] fail to offer a substantial improvement over first-order linear-chain CRFs. This is a rather expectable result, since the used training and test sequences are always at least 200 time points long, thus requiring much longer time dependencies to be modeled so as to obtain a clear improvement over first-order linear-chain CRFs. Finally, we also evaluated HMMs with the number of mixture components M selected so as to optimize model performance. We observe that HMMs yielded a significantly inferior result compared to CRF$^\infty$.

4.1.2 Skateboard: Stop and Go

Here, we consider videos depicting a human subject sliding on a skateboard, then stopping, and subsequently pushing the skateboard back to start sliding again. In Fig. 3, we provide few characteristic frames from one of the videos used in our experiments. The aim in this experiment is to train the evaluated models so as to

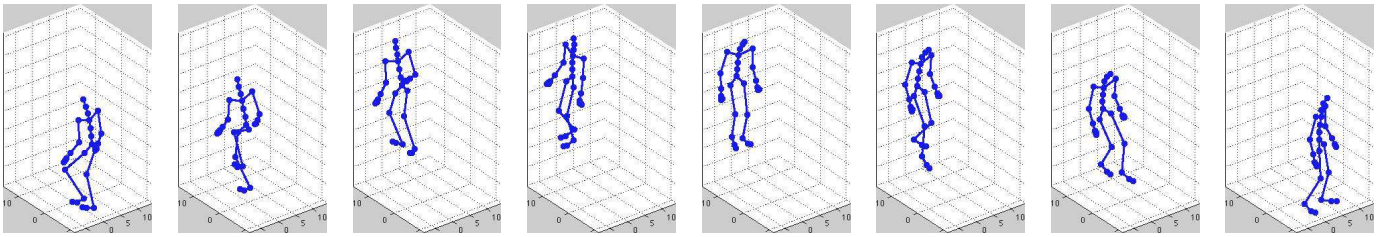

Figure 2. 3-step climbing: Few example frames from a sequence considered in our experiments.

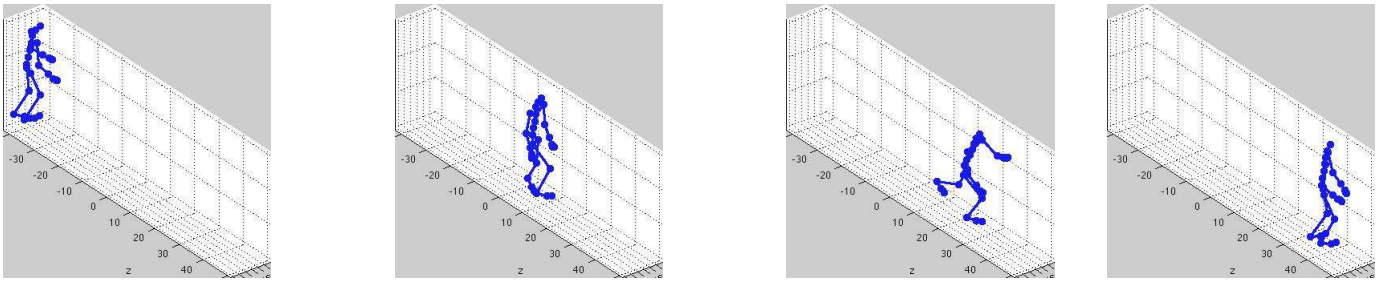

Figure 3. Skateboard: Stop and Go: Few example frames from a sequence considered in our experiments.

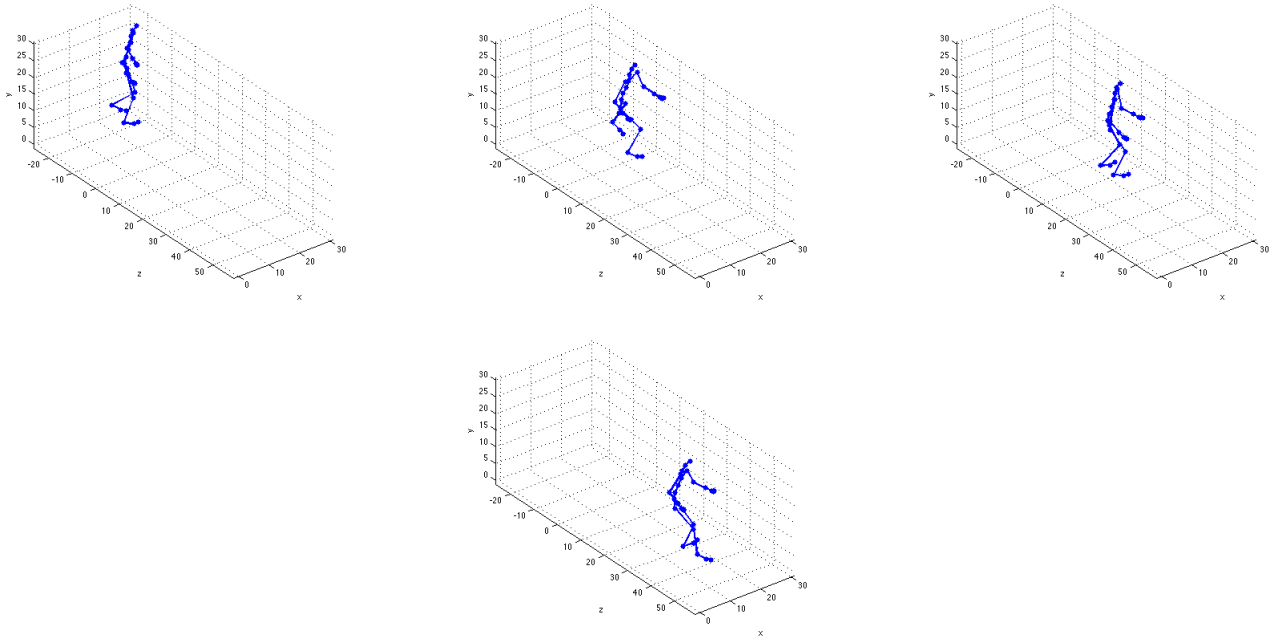

Figure 4. Skateboard: Push and Turn: Few example frames from a sequence considered in our experiments.

be capable of segmenting the videos into 3 parts: (i) sliding on the skateboard; (ii) stopping; (iii) pushing back to resume. Three different videos, comprising 368-474 frames, from the same subject are used in our experiments to perform (3-fold) leave-one-out cross validation. The obtained performance statistics of the evaluated algorithms are provided in Table 2 (means and standard deviations over the conducted 3 folds).

As we notice, the CRF$^{\infty}$ model improves considerably the obtained error rate over its first-order Markovian counterpart. Further, we again observe that the moderate order CRF models of [41] fail to offer a substantial improvement over first-order linear-chain CRFs. This is a rather expectable result, since the used training and test sequences are more than 368 time points long, thus requiring much longer time dependencies to be modeled so as to obtain a clear improvement over first-order linear-chain CRFs. Finally, we also evaluated HMMs with the number of mixture components M selected so as to optimize model performance. We observe that HMMs yielded a significantly inferior result compared to CRF-based approaches.

Table 3
Activity-based segmentation of skateboard: push and turn videos: Error rates obtained by the evaluated methods.

Method	Error Rate (%)	p-value
CRF (SCG)	33.29 ± 0.27	10^{-9}
CRF (L-BFGS)	33.08 ± 0.28	10^{-9}
CRF (EnM)	33.09 ± 0.27	10^{-9}
CRF$^\infty$ (SCG)	28.81 ± 0.23	
CRF$^\infty$ (L-BFGS)	28.63 ± 0.23	
CRF$^\infty$ (EnM)	28.65 ± 0.23	
Moderate Order CRF (3rd Order)	32.71 ± 0.24	10^{-9}
Moderate Order CRF (5th Order)	32.22 ± 0.22	10^{-9}
HMM (M=8)	37.42 ± 1.36	10^{-12}

4.1.3 Skateboard: Push and Turn

Finally, here we consider videos depicting a human subject sliding on a skateboard, subsequently pushing the skateboard back to increase speed, and then turning. In Fig. 4, we provide few characteristic frames from one of the videos used in our experiments. The aim in this experiment is to train the evaluated models so as to be capable of segmenting the videos into 3 parts: (i) sliding on the skateboard; (ii) pushing back to gain speed; (iii) turning. Four different videos, comprising 202-302 frames, from the same subject are used in our experiments to perform (4-fold) leave-one-out cross validation. The obtained performance statistics of the evaluated algorithms are provided in Table 3 (means and standard deviations over the conducted 4 folds).

As we notice, the CRF$^\infty$ model improves considerably the obtained error rate over its first-order Markovian counterpart. Further, we again observe that the moderate order CRF models of [41] fail to offer a substantial improvement over first-order linear-chain CRFs. This is a rather expectable result, since the used training and test sequences are more than 200 time points long, thus requiring much longer time dependencies to be modeled so as to obtain a clear improvement over first-order linear-chain CRFs. Finally, we also evaluated HMMs with the number of mixture components M selected so as to optimize model performance. We observe that HMMs yielded a significantly inferior result compared to CRF-based approaches.

4.2 Handwriting Recognition

In this experiment we use the handwriting recognition dataset from [37], which comprises 6877 handwritten words (i.e., time series), in which each word is represented as a series of handwritten characters (see, e.g., Fig. 5). The data comprises 55 unique words, and it consists of a total of 52152 characters (i.e., frames). Each character is a binary image of size 16 × 8 pixels, leading to a 128-dimensional binary feature vector. The dataset comprises a total of 26 unique characters (i.e., classes).

Our experimental setup is the following: the dataset is divided into 10 folds with each fold having approximately 6,000 training and 900 test examples, with the input variables $\boldsymbol{x}$ for a character being the corresponding pixel values. The obtained results are depicted in Table 4; we provide means, standard deviations, and the p-metric value of the Student's-t test run on the pairs of performances of the models (CRF, CRF$^\infty$), (moderate order CRF, CRF$^\infty$), and (HMM, CRF$^\infty$).

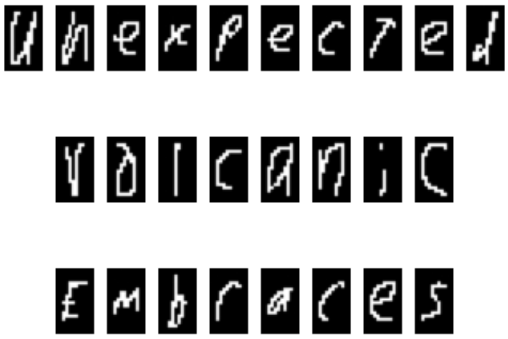

Figure 5. Handwriting recognition: Example words from the used dataset.

Table 4
Handwriting recognition: Error rates obtained by the evaluated methods.

Method	Error Rate (%)	p-value
CRF (SCG)	16.55 ± 0.74	10^{-6}
CRF (L-BFGS)	16.39 ± 0.70	10^{-6}
CRF (EnM)	16.38 ± 0.71	10^{-6}
CRF$^\infty$ (SCG)	13.15 ± 0.49	
CRF$^\infty$ (L-BFGS)	12.99 ± 0.44	
CRF$^\infty$ (EnM)	12.93 ± 0.50	
Moderate Order CRF (3rd Order)	14.03 ± 0.57	10^{-4}
Moderate Order CRF (5th Order)	13.38 ± 0.49	10^{-4}
HMM (M=6)	17.12 ± 0.89	10^{-6}

As we observe, the proposed approach offers a significant improvement over first-order linear-chain CRFs, as well as the rest of the considered alternatives. Therefore, we once again notice the practical significance of coming up with computationally efficient ways of relaxing the Markovian assumption in linear-chain CRF models applied to sequential data modeling. Note also that, in this experiment, the moderate order CRF models of [41] seem to yield a rather competitive result. This was expectable, since the average modeled sequence in this experiment is less than 10 time points long. Finally, regarding the HMM method, with the number of mixture components M selected so as to optimize model performance, we observe that the CRF$^\infty$ model yields a clear improvement, irrespectively of the employed likelihood optimization approach.

4.3 Part-of-speech tagging

Finally, here we consider an experiment with the Penn Treebank corpus [25], containing 74029 sentences with a total of 1637267 words. It comprises 49115 unique words,

Table 5
Part-of-speech tagging: Error rates obtained by the evaluated methods.

Method	Error Rate (%)
CRF (SCG)	4.45
CRF (EnM)	4.12
CRF$^\infty$ (SCG)	2.94
CRF$^\infty$ (EnM)	2.94
Moderate Order CRF (3rd Order)	3.43
Moderate Order CRF (5th Order)	3.35
HMM (M=10)	4.71

and each word in the corpus is labeled according to its part of speech; there are a total of 43 different part-of-speech labels. We use four types of features: (1) first-order word-presence features, (2) four-character prefix presence features, (3) four-character suffix presence features, and (4) four binary lexical features that indicate the presence of, respectively, a hyphen, a capital letter, a number, and an '-ing' suffix in a word. All features are measured in a window with width 3 around the current word, which leads to a total of 212610 features. We use a random 90% training / 10% test division for our experiments on the Penn Treebank corpus. No cross-validation was conducted in the case of these experiments, as the large size of the dataset rendered it rather prohibitive. Additionally, optimization by means of the L-BFGS algorithm wasn't feasible in this experiment, due to the incurred overwhelming memory requirements.

The obtained results are depicted in Table 5 for models estimated by means of likelihood optimization using the SCG method, as well as for models estimated using the maximum entropy approach. As we observe, relaxing the Markovian assumption of the CRF model yields a clear improvement over all the considered alternatives. Finally, we also evaluated diagonal covariance HMMs with the number of mixture components M selected so as to optimize model performance. We observe that HMMs yielded a rather competitive result in this experiment, clearly inferior though to the CRF$^\infty$ model.

5 Conclusions

In this paper, we presented a novel formulation of linear-chain CRF models, based on the postulation of an energy function which entails infinitely-long time-dependencies between the modeled data. This way, we circumvent the Markovian model assumption, thus allowing for better capturing the temporal dynamics in the modeled datasets, hence offering increased recognition performance compared to existing CRF-based approaches.

Building blocks of our novel approach are: (i) the *sequence memoizer*, a recently proposed nonparametric Bayesian approach for modeling label sequences with infinitely-long time dependencies; and (b) a *mean-field-like approximation* of the model marginal likelihood, which allows for the derivation of computationally efficient inference algorithms for our model. The efficacy of the so-obtained infinite-order CRF (CRF$^\infty$) model was demonstrated in several applications using benchmark datasets.

As we have shown, coming up with an efficient way of relaxing the Markovian assumption of conventional linear-chain CRF formulations allows one to obtain significantly increased discriminatory performance for the CRF model in sequential data modeling applications from diverse domains. Additionally, as our experimental results have illustrated, the benefits from relaxing the Markovian assumption of linear-chain CRF models remain significant irrespectively of the employed model training algorithm, thus vouching for the generality and the objectivity of our findings regarding the superiority of the CRF$^\infty$ model.

Acknowledgement

This work has been funded by the EU FP7 ALIZ-E project (grant 248116).

References

[1] "The CMU MoCap database. [Online]. Available: http://mocap.cs.cmu.edu/."
[2] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Athena Scientific, 1999.
[3] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, 2001.
[4] G. Celeux, F. Forbes, and N. Peyrard, "EM procedures using mean field-like approximations for Markov model-based image segmentation," *Pattern Recognition*, vol. 36, no. 1, pp. 131–144, 2003.
[5] D. Chandler, *Introduction to Modern Statistical Mechanics*. Oxford: Oxford University Press, 1987.
[6] S. P. Chatzis and T. A. Varvarigou, "A fuzzy clustering approach toward hidden Markov random field models for enhanced spatially constrained image segmentation," *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 5, pp. 1351–1361, October 2008.
[7] S. P. Chatzis, D. I. Kosmopoulos, and T. A. Varvarigou, "Robust sequential data modeling using an outlier tolerant hidden Markov model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 9, pp. 1657–1669, 2009.
[8] S. P. Chatzis and G. Tsechpenakis, "The infinite hidden Markov random field model," *IEEE Transactions on Neural Networks*, vol. 21, no. 6, pp. 1004–1014, 2010.
[9] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient belief propagation for early vision," *Int. J. Comput. Vis.*, vol. 70, no. 1, pp. 41–54, 2006.
[10] T. Ferguson, "A Bayesian analysis of some nonparametric problems," *The Annals of Statistics*, vol. 1, pp. 209–230, 1973.
[11] J. Gasthaus and Y. W. Teh, "Improvements to the Sequence Memoizer," in *Proc. NIPS*, 2011.
[12] D. Geiger and F. Girosi, "Parallel and deterministic algorithms from MRFs: surface reconstruction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 5, pp. 401–412, 1991.
[13] X. He, R. S. Zemel, and M. A. Carreira-Perpinan, "Multiscale conditional random fields for image labelling," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2004, pp. 692–705.
[14] T. Hofmann and J. Buhmann, "Pairwise data clustering by deterministic annealing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 1, pp. 1–14, 1997.
[15] T. Jaakkola and M. Jordan, "Improving the mean field approximation via the use of mixture distributions," in *Learning in Graphical Models*, M. Jordan, Ed. Dordrecht: Kluwer Academic Publishers, 1998, pp. 163–173.
[16] P. Kohli, M. Kumar, and P. Torr, "p^3 and beyond: Solving energies with higher order cliques," *Proc. IEEE Conf. CVPR*, pp. 1–8, 2007.

[17] P. Kohli, L. Ladicky, and P. Torr, "Robust higher order potentials for enforcing label consistency," *International Journal of Computer Vision*, vol. 82, pp. 302–324, 2009.
[18] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?" *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 147–159, 2004.
[19] N. Komodakis and N. Paragios, "Beyond pairwise energies: Efficient optimization for higher-order MRFs," in *Proc. IEEE Conf. CVPR*, 2009, pp. 2985–2992.
[20] S. Kumar and M. Hebert, "Discriminative random fields," *International Journal of Computer Vision*, vol. 68, no. 2, pp. 179–201, 2006.
[21] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. 18th International Conf. on Machine Learning*, 2001.
[22] X. Lan, S. Roth, D. Huttenlocher, and M. J. Black, "Efficient belief propagation with learned higher-order Markov random fields," in *Proc. ECCV*, vol. 2, 2006, pp. 269–282.
[23] N. Lawrence, "Gaussian process software: http://staffwww.dcs.shef.ac.uk/people/n.lawrence/software.html."
[24] D. Liu and J. Nocedal, "On the limited memory method for large scale optimization," *Mathematical Programming B*, vol. 45, no. 3, pp. 503–528, 1989.
[25] M. Marcus, B. Santorini, and M. Marcinkiewicz, "Building a large annotated corpus of English: the Penn Treebank," in *Corpus Linguistics: Readings in a Widening Discipline*, G. Sampson and D. McCarthy, Eds. Continuum, 2004.
[26] A. McCallum and W. Li, "Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons," in *In Seventh Conference on Natural Language Learning (CoNLL)*, 2003, pp. 188–191.
[27] R. McDonald and F. Pereira, "Identifying gene and protein mentions in text using conditional random fields," *BMC Bioinformatics*, vol. 6, no. Suppl 1, p. S6, 2005.
[28] G. McLachlan and D. Peel, *Finite Mixture Models*. Wiley Series in Probability and Statistics, 2000.
[29] F. Peng, F. Feng, and A. Mccallum, "Chinese segmentation and new word detection using conditional random fields," *Science*, pp. 562–es, 2004.
[30] J. Pitman and M. Yor, "The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator," in *Annals of Probability*, vol. 25, 1997, pp. 855–900.
[31] B. Potetz, "Efficient belief propagation for vision using linear constraint nodes," in *Proc. IEEE Conf. CVPR*, 2007, pp. 1–8.
[32] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, pp. 245–255, 1989.
[33] C. Rother, P. Kohli, W. Feng, and J. Jia, "Minimizing sparse higher order energy functions of discrete variables," in *Proc. IEEE Conf. CVPR*, 1382-1389, Ed., 2009.
[34] P. Sen and L. Getoor, "Cost-sensitive learning with conditional Markov networks," *Data Mining and Knowledge Discovery*, vol. 17, no. 2, pp. 136–163, 2008.
[35] F. Sha and F. Pereira, "Shallow parsing with conditional random fields," in *Proc. NAACL '03*, vol. 1, 2003, pp. 134–141.
[36] C. Sutton and A. McCallum, "An introduction to conditional random fields for relational learning," in *Introduction to Statistical Relational Learning*, L. Getoor and B. Taskar, Eds. MIT Press, 2006.
[37] B. Taskar, C. Guestrin, and D. Koller, "Max-margin Markov networks," in *Advances in Neural Information Processing Systems*, vol. 16, 2004.
[38] Y. W. Teh, "A hierarchical Bayesian language model based on Pitman-Yor processes," in *Proc. Association for Computational Linguistics*, 2006, pp. 985–992.
[39] F. Wood, C. Archambeau, J. Gasthaus, L. F. James, and Y. Teh, "A stochastic memoizer for sequence data," in *Proc. International Conference on Machine Learning (ICML)*, 2009.
[40] F. Wood, J. Gasthaus, C. Archambeau, L. James, and Y. W. Teh, "The sequence memoizer," *Communications of the ACM*, vol. 54, no. 2, pp. 91–98, 2011.
[41] N. Ye, W. S. Lee, H. L. Chieu, and D. Wu, "Conditional random fields with high-order features for sequence labeling," in *Proc. NIPS*, 2009.
[42] A. Yuille, "Generalized deformable models, statistical physics and matching problems," *Neural Comput.*, vol. 2, pp. 1–24, 1990.
[43] J. Zerubia and R. Chellappa, "Mean field approximation using compound Gauss-Markov random field for edge detection and image restoration," in *Proc. ICASSP*, 1990, pp. 2193–2196.
[44] J. Zhang, "The mean field theory in EM procedures for Markov random fields," *IEEE Transactions on Image Processing*, vol. 2, no. 1, pp. 27–40, 1993.
[45] J. Zhang and S. Gong, "Action categorization with modified hidden conditional random field," *Pattern Recognition*, vol. 43, no. 1, pp. 197–203, 2010.