A Conditional Random Field-Based Model for Joint Sequence Segmentation and Classification

Sotirios P. Chatzis[a], Dimitrios I. Kosmopoulos[b,c], Paul Doliotis[c,d]

[a] *Department of Electrical Engineering, Computer Engineering, and Informatics Cyprus University of Technology*
[b] *Department of Computer Science, Rutgers University, 08854, NJ-USA*
[c] *NCSR Demokritos, Inst. of Informatics and Telecommunications, GR- 15310, Greece*
[d] *University of Texas at Arlington, Computer Science and Engineering, 76013, TX*

Abstract

In this paper, we consider the problem of joint segmentation and classification of sequences in the framework of conditional random field (CRF) models. To effect this goal, we introduce a novel dual-functionality CRF model: on the first level, the proposed model conducts sequence segmentation, while, on the second level, the whole observed sequences are classified into one of the available learned classes. These two procedures are conducted in a joint, synergetic fashion, thus optimally exploiting the information contained in the used model training sequences. Model training is conducted by means of an efficient likelihood maximization algorithm, and inference is based on the familiar Viterbi algorithm. We evaluate the efficacy of our approach considering a real-world application, and we compare its performance to popular alternatives.

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

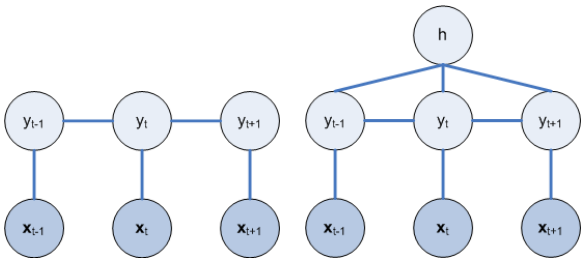
Email addresses: soteri0s@me.com (Sotirios P. Chatzis), dkosmo@ieee.org (Dimitrios I. Kosmopoulos), doliotis@iit.demokritos.gr (Paul Doliotis)

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) [1].

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 [2]. During the last years, we have witnessed an explosion of interest in CRFs, as they have 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 [1, 3, 4, 5, 6, 7, 8].

In this paper, we focus on linear-chain CRFs; linear-chain CRFs, the basic probabilistic principle of which is illustrated in Fig. 1(a), are conditional probability distributions over label sequences, which are conditioned on the observed sequences [1, 2]. 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 CRF-based model for joint segmentation and classification of observed sequences. Indeed, joint sequence segmentation and classification is a perennial problem that occurs in several application areas, such as object and behavior recognition in computer vision applications (e.g., [9, 10]), speech analysis (e.g., [11]), and bioinformatics [12]. By jointly treating the two tasks of sequence decoding (segmentation) and classification, one can more effectively exploit the available information, thus allowing for a potentially considerable increase in the obtained algorithm performance [10].

Towards this end, in this paper we propose a novel dual-functionality CRF (DF-CRF) model for joint sequence segmentation and classification. Our proposed model comprises two levels of functionality: In the first level, the observed sequences are segmented, using a variant of the familiar Viterbi algorithm. In the second level, classification of the observed sequences is performed. Both these procedures are conducted concurrently and in a synergetic fashion, thus optimally exploiting the information acquired from the available training data. Model training is effected by means of a computationally efficient likelihood maximization algorithm. We evaluate our novel approach in a real-world visual workflow segmentation and recognition application; as we show, our proposed approach offers considerable improvement over hidden Markov models (HMMs) [13], standard CRFs, as well as hidden conditional random field (HCRF) models [9], a method related to the DF-CRF, but considering that the sequence segment



(a) Conventional CRF (b) Proposed CRF

Figure 1: Linear-chain conditional random fields: the conventional and the proposed approach. The brightly colored nodes denote a random variable y_t, and the shaded nodes $\mathbf{x}_t$ have been set to the respective observed values. h denotes the sequence label.

labels are not observable and, hence, comprise a latent variable of the model.

The remainder of this paper is organized as follows: In Section 2, a brief introduction to CRFs is provided. In Section 3, the proposed DF-CRF model is introduced, its inference algorithms are derived, and we discuss the differences between our proposed approach and HCRFs. In Section 4, we apply our model to a real-world application dealing with visual workflow recognition and decoding, using challenging datasets obtained from the assembly lines of an automobile manufacturer. We compare our method's performance to HMMs, standard CRFs, and HCRF models. Finally, in the concluding section of this paper, we summarize our contribution and results.

2. 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 [2].

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 $\boldsymbol{x} = \left([\boldsymbol{x}_t^{\mathrm{T}}]_{t=1}^{T}\right)^{\mathrm{T}}$, 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

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

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

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

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

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

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

where the $\phi_t^1(y_t, \boldsymbol{x}_t)$ and the $\phi_t^2(y_t, y_{t-1})$ are the *unary* and *transition potentials* of the model, respectively, centered at the current time point. Note that, in the above definition, we have considered that the transition potentials $\phi_t^2(y_t, y_{t-1})$ do not depend on the observations $\boldsymbol{x}_t$, but, instead, a given transition, say from state i to state j, always receives the same transition potential function value regardless of the input. Such a model formulation is usually referred to as a hidden Markov model (HMM)-like linear-chain CRF [2]. We will be considering this form of transition potentials throughout this work; however, our results can be easily extended to any other formulation, where the transition potentials are assumed to also depend on the observed input variables $\boldsymbol{x}$.

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

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

and

$$\phi_t^2(y_t, y_{t-1}) = \sum_{i=1}^{K} \sum_{j=1}^{K} \delta(y_t - j) \delta(y_{t-1} - i) \xi_{ij} \tag{6}$$

with

$$\phi_1^2(y_1) = \sum_{i=1}^{K} \delta(y_t - i) \xi_i \tag{7}$$

where $\delta(\sigma)$ is the Dirac delta function, the parameters $\boldsymbol{\omega}_i$ are the prior weights of an observation emitted from state i, the parameters ξ_{ij} are related to the prior

probabilities of the transition from state i to state j, and the parameters ξ_i 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 [14], or its limited memory variant (L-BFGS) [15], which, indeed, is the most commonly used method in the CRF literature [1, 2].

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 [16, 13], widely known from the HMM literature. Indeed, as discussed, e.g., in [2], it is easy to show that

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

where the $\alpha_t(j)$ are the forward probabilities of state j at time t, which in the case of linear-chain CRF models yield [2]

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

with initialization

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

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

$$\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}} \log p(\boldsymbol{y}|\boldsymbol{x}) \tag{11}$$

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 decoding in the HMM literature, namely the Viterbi algorithm [16]. In the case of linear-chain CRFs, it can be shown that the Viterbi algorithm yields the following recursion [2]

$$\delta_t(j) = \max_{1 \leq i \leq K} \exp\Big\{ \phi_t(y_t = j, y_{t-1} = i, \boldsymbol{x}_t) \Big\} \delta_{t-1}(i) \tag{12}$$

with initialization

$$\delta_1(j) = \exp\Big\{ \phi_1(y_1 = j, \boldsymbol{x}_1) \Big\} \tag{13}$$

based on which, output sequence optimization reads

$$\hat{y}_t = \arg\max_{1 \leq i \leq K} \delta_t(i) \tag{14}$$

A graphical illustration of the considered linear-chain CRF models is presented in Fig. 1(a).

3. Proposed Approach

Let us consider a task where, given a sequence of observations $\boldsymbol{x} = \left([\boldsymbol{x}_t^{\mathrm{T}}]_{t=1}^{T}\right)^{\mathrm{T}}$, we wish to obtain a sequence of segment labels $\boldsymbol{y} = [y_t]_{t=1}^{T}$, as well as a class value $h \in \mathcal{H}$ for the whole sequence. For this purpose, we postulate a CRF-based model of the form

$$p(h, \boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \exp\left[\sum_{t=2}^{T} \phi_t(h, y_t, y_{t-1}, \boldsymbol{x}_t) + \phi_1(h, y_1, \boldsymbol{x}_1)\right] \tag{15}$$

where the partition function is given by

$$Z(\boldsymbol{x}) = \sum_{h} \sum_{\boldsymbol{y}} \exp\left[\sum_{t=2}^{T} \phi_t(h, y_t, y_{t-1}, \boldsymbol{x}_t) + \phi_1(h, y_1, \boldsymbol{x}_1)\right] \tag{16}$$

and we introduce a set of *class-conditional* potential functions of the form

$$\phi_t(h, y_t, y_{t-1}, \boldsymbol{x}_t) = \phi_t^1(h, y_t, \boldsymbol{x}_t) + \phi_t^2(h, y_t, y_{t-1}) \tag{17}$$

$$\phi_1(h, y_1, \boldsymbol{x}_1) = \phi_1^1(h, y_1, \boldsymbol{x}_1) + \phi_1^2(h, y_1) \tag{18}$$

comprising the *class-conditional* transition potential functions $\phi_t^2(h, y_t, y_{t-1})$, and the *class-conditional* unary potential functions $\phi_t^1(h, y_t, \boldsymbol{x}_t)$. For example, under the linear potential functions selection discussed in Section 2 (Eqs. (5)-(7)), we may write

$$\phi_t^1(h, y_t, \boldsymbol{x}_t) = \sum_{i=1}^{K} \delta(y_t - i) \boldsymbol{\omega}_i^h \cdot \boldsymbol{x}_t \tag{19}$$

and

$$\phi_t^2(h, y_t, y_{t-1}) = \sum_{i=1}^{K} \sum_{j=1}^{K} \delta(y_t - j)\delta(y_{t-1} - i)\xi_{ij}^h \tag{20}$$

with

$$\phi_1^2(h, y_1) = \sum_{i=1}^{K} \delta(y_t - i)\xi_i^h \tag{21}$$

Any other selection of the class-conditional potential functions of our model is also valid, as soon as their parameters do also depend on the value of the sequence classes $h \in \mathcal{H}$.

Definition 1. A discriminative linear-chain CRF-based model with conditional probability $p(h, \boldsymbol{y}|\boldsymbol{x})$ of the form (15) shall be dubbed the dual-functionality

CRF model.

Without any loss of generality, in the remainder of this work, we will be considering class-conditional potential functions of the forms (19)-(21) for the proposed DF-CRF model. A graphical illustration of the proposed method is presented in Fig. 1(b).

Having defined the proposed DF-CRF model, we can now proceed to the derivation of its training and inference algorithms.

3.1. Model Training

To begin with, training for the DF-CRF model comprises estimation of the parameters of the model potential functions. Considering class-conditional potential functions of the forms (19)-(21), this reduces to estimation of the parameters $\{\boldsymbol{\omega}_i^h\}_{h\in\mathcal{H},i\in\mathcal{Y}}$, $\{\xi_{ij}^h\}_{h\in\mathcal{H},i,j\in\mathcal{Y}}$, and $\{\xi_i^h\}_{h\in\mathcal{H},i\in\mathcal{Y}}$. To effect DF-CRF model training, we resort to optimization of the model's log-likelihood function, reading

$$\begin{aligned}\log p(h,\boldsymbol{y}|\boldsymbol{x}) = \sum_{t=2}^{T}\phi_t(h,y_t,y_{t-1},\boldsymbol{x}_t) + \phi_1(h,y_1,\boldsymbol{x}_1) \\ - \log Z(\boldsymbol{x})\end{aligned} \tag{22}$$

over each one of its parameters. As we observe, computation of the model log-likelihood (22) requires calculation of the quantity $Z(\boldsymbol{x})$ which entails summation over all possible $\boldsymbol{y}$ and h values. To make this computation tractable, we express $Z(\boldsymbol{x})$ in the following form

$$Z(\boldsymbol{x}) = \sum_{h\in\mathcal{H}} Z(\boldsymbol{x}|h) \tag{23}$$

where

$$Z(\boldsymbol{x}|h) = \sum_{\boldsymbol{y}} \exp\left[\sum_{t=2}^{T}\phi_t(h,y_t,y_{t-1},\boldsymbol{x}_t) + \phi_1(h,y_1,\boldsymbol{x}_1)\right] \tag{24}$$

Under this regard, the partition function $Z(\boldsymbol{x})$ of the DF-CRF model becomes tractable, as it arises as the mere summation of the terms $Z(\boldsymbol{x}|h)$, the number of which is equal to the number of modeled classes (i.e., the cardinality of $\mathcal{H}$), with each one of them being easily computable by means of the forward-backward algorithm. Indeed, based on the relevant discussions of Section 2, we will have

$$Z(\boldsymbol{x}|h) = \sum_{j=1}^{K}\alpha_T(h,j) \tag{25}$$

where the *class-conditional* forward probabilities $\alpha_t(h,j)$ are given by

$$\begin{aligned}\alpha_t(h,j) = \sum_{i=1}^{K}\alpha_{t-1}(h,i)\exp\Big\{\phi_t(h,y_t=j,y_{t-1}=i,\boldsymbol{x}_t)\Big\} \\ t\geq 2\end{aligned} \tag{26}$$

with initialization

$$\alpha_1(h, j) = \exp\Big\{\phi_1(h, y_1 = j, \boldsymbol{x}_1)\Big\} \tag{27}$$

Having computed the partition functions $Z(\boldsymbol{x})$, maximization of $\log p(h, \boldsymbol{y}|\boldsymbol{x})$ given a set of N training sequences $\{\boldsymbol{x}_n, \boldsymbol{y}_n, h_n\}_{n=1}^N$, is easily conducted by means of the L-BFGS algorithm, similar to the discussions of Section 2.

3.2. Inference Algorithm

The inference algorithm for the DF-CRF model comprises two separate procedures: (i) sequence classification; and, (ii) sequence segmentation.

3.2.1. Sequence classification

Sequence classification is the problem of finding the optimal class $\hat{h} \in \mathcal{H}$ for a given observed sequence $\boldsymbol{x}$. This problem can be formulated as follows:

$$\hat{h} = \arg\max_{h \in \mathcal{H}} p(h|\boldsymbol{x}) \tag{28}$$

Based on (22), we have

$$\begin{aligned} p(h|\boldsymbol{x}) &= \sum_{\boldsymbol{y} \in \mathcal{Y}} p(h, \boldsymbol{y}|\boldsymbol{x}) \\ &= \frac{\sum_{\boldsymbol{y}} \exp\left[\sum_{t=2}^{T} \phi_t(h, y_t, y_{t-1}, \boldsymbol{x}_t) + \phi_1(h, y_1, \boldsymbol{x}_1)\right]}{Z(\boldsymbol{x})} \\ &= \frac{Z(\boldsymbol{x}|h)}{Z(\boldsymbol{x})} \end{aligned} \tag{29}$$

Based on (29), criterion (28) eventually yields

$$\hat{h} = \arg\max_{h \in \mathcal{H}} Z(\boldsymbol{x}|h) \tag{30}$$

The latter quantities $Z(\boldsymbol{x}|h)$ can be easily computed by means of the forward iterations (26), (27), described in Section 3.1, given the observed sequence $\boldsymbol{x}$. Therefore, sequence classification in our model consists in merely computing the *class-conditional* partition functions $Z(\boldsymbol{x}|h)$, and picking the class $\hat{h} \in \mathcal{H}$ which maximizes them.

3.2.2. Sequence segmentation

Given an observed sequence $\boldsymbol{x}$, sequence segmentation, also referred to as sequence decoding or labeling, consists in the optimization problem

$$\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}} \log p(\boldsymbol{y}|\boldsymbol{x}) \tag{31}$$

That is, we want to assign an optimal segment label to each one of the observations comprising the sequence $\boldsymbol{x}$. Based on (22), and using Jensen's inequality, we have

$$\begin{aligned} \log p(\boldsymbol{y}|\boldsymbol{x}) =& \log \sum_{h \in \mathcal{H}} p(h, \boldsymbol{y}|\boldsymbol{x}) \\ & \geq \sum_{h \in \mathcal{H}} \left[\sum_{t=2}^{T} \phi_t(h, y_t, y_{t-1}, \boldsymbol{x}_t) + \phi_1(h, y_1, \boldsymbol{x}_1) \right] + const \\ & \propto \sum_{t=2}^{T} \psi_t(y_t, y_{t-1}, \boldsymbol{x}_t) + \psi_1(y_1, \boldsymbol{x}_1) \end{aligned} \tag{32}$$

where

$$\psi_t(y_t, y_{t-1}, \boldsymbol{x}_t) = \sum_{h \in \mathcal{H}} \phi_t(h, y_t, y_{t-1}, \boldsymbol{x}_t) \tag{33}$$

$$\psi_1(y_1, \boldsymbol{x}_1) = \sum_{h \in \mathcal{H}} \phi_1(h, y_1, \boldsymbol{x}_1) \tag{34}$$

Based on (32), we observe that $\log p(\boldsymbol{y}|\boldsymbol{x})$ is lower-bounded by a quantity analogous to

$$\mathcal{L}(\boldsymbol{y}|\boldsymbol{x}) = \sum_{t=2}^{T} \psi_t(y_t, y_{t-1}, \boldsymbol{x}_t) + \psi_1(y_1, \boldsymbol{x}_1) \tag{35}$$

Exploiting this result, we solve the sequence decoding problem (31) by reducing it to the following problem:

$$\begin{aligned} \hat{\boldsymbol{y}} &= \underset{\boldsymbol{y}}{\text{argmax}} \mathcal{L}(\boldsymbol{y}|\boldsymbol{x}) \\ &= \arg \max_{\boldsymbol{y}} \left\{ \sum_{t=2}^{T} \psi_t(y_t, y_{t-1}, \boldsymbol{x}_t) + \psi_1(y_1, \boldsymbol{x}_1) \right\} \end{aligned} \tag{36}$$

Following the relevant discussions of Section 2, it is easy to observe that this problem is essentially a dynamic programming problem with backwards recursion

$$\hat{y}_t = \arg \max_{1 \leq i \leq K} \big\{ \delta_t(i) \big\} \tag{37}$$

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

$$\delta_t(j) = \max_{1 \leq i \leq K} \exp \Big\{ \psi_t(y_t = j, y_{t-1} = i, \boldsymbol{x}_t) \Big\} \delta_{t-1}(i) \tag{38}$$

with initialization

$$\delta_1(j) = \exp \Big\{ \psi_1(y_1 = j, \boldsymbol{x}_1) \Big\} \tag{39}$$

which, in essence, is nothing more than the Viterbi algorithm itself, using the

$\psi_t(\cdot)$ functions instead of the potential functions of the model.

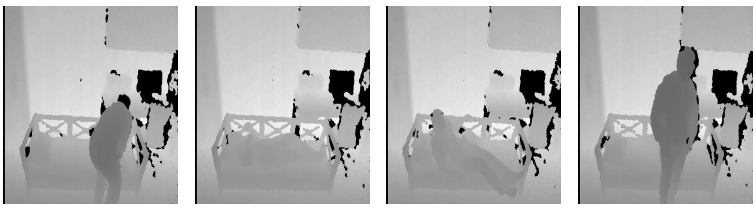
3.3. Relation to existing approaches

The hidden conditional random field model is the most closely related existing approach to the DF-CRF model. Indeed, in cases of sequential data modeling, the HCRF model does also postulate a conditional density expression of the form (15). However, in the case of the HCRF model, the segment labels $y \in \mathcal{Y}$ are considered latent variables during model training, and are marginalized out when making predictions (sequence classification). As such, HCRF model training cannot exploit existing segmentation information of the available training sequences, thus yielding less accurate classification models. Additionally, a trained HCRF model cannot be used for supervised sequence segmentation, but its functionality is limited to sequence classification tasks.

4. Experiments and Results

4.1. Action recognition from depth images

(a) Color images

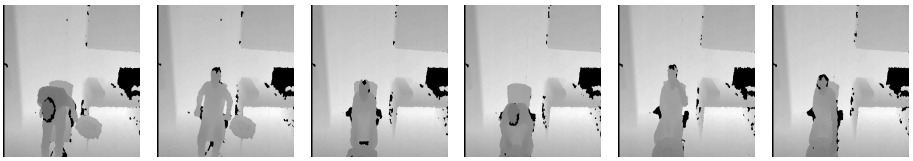


(b) Depth images

Figure 2: Key frames from activity 1: action 1 - go to bed (frames 1,2), and action 2 - get up from bed (frames 3,4))

In our first experiment, we evaluated our method in segmenting and classifying sequences using depth images, which depict humans performing actions in an assistive living environment. More specifically, we have used the dataset described in [17], which includes several actions from which we have selected

(a) Color images



(b) Depth images

Figure 3: Key frames from activity 2: action 3 - sit down (frames 1,2), action 4 - eat meal (frames 3,4)), action 5 - drink water (frames 5,6)

the following: (1) get up from bed, (2) go to bed, (3) sit down, (4) eat meal, and (5) drink water.

We sought to recognize two activities: activity 1 comprised actions (1)-(2) (see Fig. 2); activity 2 comprised actions (3),(4),(5) (see Fig. 3). The activity classes h that we sought to recognize were the activity classes 1 and 2, and the frame labels y were the five different actions (1)-(5). The observable input was the sequence of vectors $\boldsymbol{x}$, which was extracted as described next.

For each depth image we extracted features similar to [17] using a variation of Motion History Images (MHIs). MHIs are among the first holistic representation methods for behavior recognition [18]. In an MHI H_τ, pixel intensity is a function of the temporal history of motion at that point.

$$H_\tau^I(x,y,t) = \begin{cases} \tau, \text{if } |I(x,y,t) - I(x,y,t-1)| > \delta I_{th} \\ max(0, H_\tau^I(x,y,t-1) - 1), \text{otherwise.} \end{cases} \tag{40}$$

Here τ is the longest time window we want the system to consider and δI_{th} is the threshold value for generating the mask for the region of motion. The result is a scalar-valued image where more recently moving pixels are brighter.

Ni et al. [17] proposed the use of a depth sensor and they introduced the motion history along the depth changing directions. To encode the backward motion history (decrease of depth), they introduced the backward-DMHI (bDMHI):

$$H_\tau^{fD}(x,y,t) = \begin{cases} \tau, \text{if } D(x,y,t) - D(x,y,t-1) < -\delta I_{th} \\ max(0, H_\tau^{fD}(x,y,t-1) - 1), \text{otherwise.} \end{cases} \tag{41}$$

Here, H_t^{bD} denotes the backward motion history image and $D(x, y, t)$ denotes the depth sequence. δI_{th} is the threshold value for generating the mask for the region of backward motion. Similarly is defined the forward history image, which we don't use in our experiments, but is expected to give similar results.

In order to calculate the depth change induced motion history images, according to the above equations, we use depth maps captured by a Kinect$^{\text{TM}}$ device. Kinect depth maps however have the main disadvantage of the presence of a significant amount of noise. After frame differencing and thresholding we noticed that motion was encoded even in areas where there are only still objects. To tackle this problem, we used a median filtering at the spatial domain. In the temporal domain each pixel value was replaced by the minimum of its neighbors.

The MHI images are represented by means of the complex Zernike coefficients A_{00}, A_{11}, A_{20}, A_{22}, A_{31}, A_{33}, A_{40}, A_{42}, A_{44}, A_{51}, A_{53}, A_{55}, A_{60}, A_{62}, A_{64}, A_{66}, for each of which the norm and the angle were included in the provided descriptors. We used a total of 31 parameters (constant elements were removed), thus providing an acceptable scene reconstruction without a computationally prohibitive dimension.

In our experiments, we have used 35 action sets per type (these are the first 35 samples in the dataset for each action). We have used cross validation in the following fashion: in each cycle fifteen of these sets were randomly selected to perform training, and the rest twenty were used for testing. We run the same experiment 50 times to account for the effect of random selection of samples. We postulated models comprising 2+3 states to account for the five different actions.

We evaluated the decoding quality of our method by comparing it to the standard CRF and to an HMM with explicitly trained states, hereafter mentioned as pseudo-HMM. The training and testing procedures for the CRF and DF-CRF models were conducted as described in sections 2 and 3, respectively. Regarding the pseudo-HMM, we postulated a 5-state model with Gaussian mixture models (GMMs) used as their emission probability distributions; we experimentally found that three mixture component distributions per model state was the selection giving the best performance for all the postulated models. For the pseudo-HMM each state was trained using the expectation-maximization (EM) algorithm for GMMs and the transition matrix and priors were trivially computed by counting based on ground truth labels. The Viterbi algorithm for HMMs was used for sequence segmentation.

Additionally, apart from sequence segmentation, we also evaluated the sequence classification performance of our model. For this purpose, we used our model to classify the test workflows into two classes corresponding to actions 1 and 2, respectively. For comparison, we repeated the same experiment using standard HMMs in a maximum a posteriori classification fashion. For the same purpose, we also used HCRFs with 4 hidden states and potential functions similar to the DF-CRF, trained as described in [9].

The results for the segmentation (labeling) task are given in Table 1, and for the classification task are illustrated in Table 2. As we observe, DF-CRF

generally performs better than the competition. To examine the statistical significance of the differences between the evaluated methods, we made use of the Student-t test between the three evaluated methods. Our results verified with a 24% significance level that the DF-CRF and the standard CRF approach yield statistically significant performance differences. Concerning the pairs of DF-CRF/pseudo-HMM and of standard CRF/pseudo-HMM, the statistical significance of the results was verified at the 0.01% significance level in both cases.

From the above results we infer that the performance difference between DF-CRF and standard CRF is not very big and this can be largely attributed to the fact that the actions (1),(2) look quite different from actions (3),(4),(5), so a more elaborate method does not improve much the results. The pseudo-HMM is clearly inferior to both approaches in segmentation. This seems to be reasonable, due to the fact that the CRF approaches are trained to optimize the conditional probability of labels given an observation sequence, while for the pseudo-HMM we trained separately for each label y, thus yielding generative models for each of the five y values. Similarly, concerning the classification into activities h, the standard HMM was inferior to both CRF approaches, which can be explained by the fact that the HMM is generative, while the CRF-based models are discriminative and more appropriate for classification problems.

We would like to mention that the obtained segmentation error rates could have been reduced even more by employing some application constraints, such as the number of times that a task may appear in an activity, the sequence of tasks etc. Yet, we assumed no such constraints in our experiments, because their imposition is not in the scope of demonstrating the segmentation method.

4.2. Workflows from RGB images

(a) Workflow 1: task 1, task 2, task 3 (2 key frames for each task)

(b) Workflow 2: task 4, task 5, task 6 (2 key frames for each task)

Figure 4: Key frames from workflow 1 and workflow 2.

		Recognized Label				
		1	2	3	4	5
True Label	1	.61	.12	.07	.07	.12
	2	.51	.22	.01	.12	.12
	3	.20	.01	.37	.17	.22
	4	.21	.01	.01	.63	.12
	5	.19	.01	.04	.34	.40
						Total error = 50.01%

(a) pseudo HMM

		Recognized Label				
		1	2	3	4	5
True Label	1	.50	.33	.06	.06	.05
	2	.18	.68	.02	.07	.05
	3	.09	.03	.45	.19	.25
	4	.05	.06	.01	.72	.15
	5	.06	.05	.05	.38	.45
						Total error = 39.44%

(b) CRF

		Recognized Label				
		1	2	3	4	5
True Label	1	.53	.37	.02	.04	.03
	2	.20	.72	.00	.04	.02
	3	.07	.02	.46	.19	.25
	4	.04	.05	.01	.73	.16
	5	.05	.04	.06	.39	.46
						Total error = 37.57%

(c) DF-CRF

Table 1: Action recognition from depth images: Confusion matrix for the segmentation task. The results are normalized based on the total number of frames per activity, considering all cross-validation runs.

In our second experiment, we considered a visual workflow segmentation and classification application. We used the workflow recognition (WR) dataset, a real-world dataset captured in the assembly line (workcells) of a major automobile manufacturer [19]. In that dataset, two factory workers are depicted picking up car parts from various racks and placing them on a welding cell, where a robot performs welding.

This experiment differs from the previous one in the following ways: (a) The features are extracted from color images and not from depth data; (b) some of the subtasks y are similar; and (c) the same subtask may appear in

		Recognized Class	
		1	2
True Class	1	.69	.31
	2	.16	.84

Total error = 23.34%

(a) HMM

		Recognized Class	
		1	2
True Class	1	.836	.163
	2	.107	.893

Total error = 13.50%

(b) H-CRF

		Recognized Class	
		1	2
True Class	1	.925	.075
	2	.09	.91

Total error = 8.25%

(c) DF-CRF

Table 2: Action recognition from depth images: Confusion matrix for classification task. The results are normalized based on the total number of activities, considering all cross-validation runs.

both the considered workflow types h. Especially issues (b) and (c) posed great challenges to the classification algorithm, necessitating effective utilization of context information from all the modeled tasks.

We experimented with two workflows of the dataset, pertaining to car assembly, and used visual data captured from camera #1 (see [19] for more details). Briefly, the two considered workflows comprise the following three tasks each:

Workflow 1 (WF1):

- *Task 1:* One worker picks part #1 from rack #1 and places it on the welding cell.
- *Task 2:* Two workers pick part #2a from rack #2 and place it on the welding cell.
- *Task 3:* Two workers pick part #2b from rack #3 and place it on the welding cell.

Workflow 2 (WF2):

- *Task 4:* A worker picks up parts #3a and #3b from rack #4 and places them on the welding cell.
- *Task 5:* A worker picks up part #4 from rack #1 and places it on the welding cell.
- *Task 6:* Two workers pick up part #5 from rack #5 and place it on the welding cell.

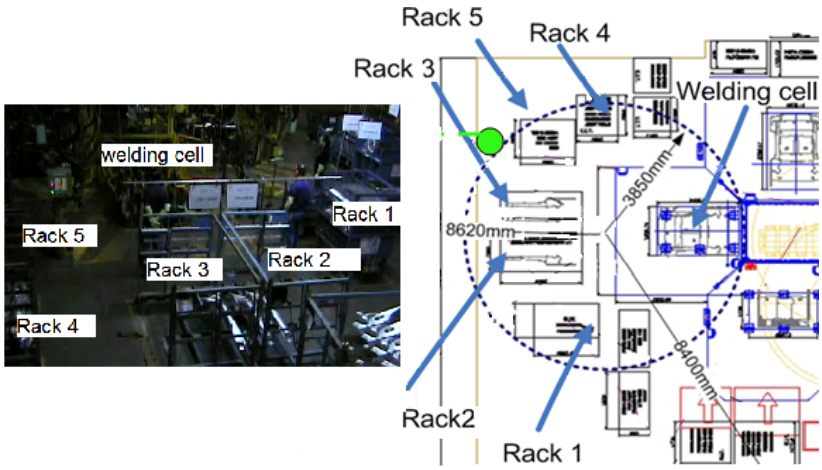


Figure 5: Depiction of a workcell along with the position of the used camera (camera 1) and the racks #1-5. Each of the actions (labels y) to identify are associated with transferring each rack from the respective pallet and putting it on the welding cell. The classes h are sequences of these actions.

Some intervals of inactivity (idle task) are possible in both WF1 (mainly at the beginning), and WF2 (mainly at the end). Therefore, each of these workflows can be considered as actually composed of four tasks, including the previously described three tasks, as well as the (common) idle task. Some key frames from an example case of the considered workflows are given in Fig. 4, while the workcell configuration is illustrated in Fig. 5.

In this application, the class h is the identified workflow class (WF1 or WF2), and the frame labels correspond to the six different tasks plus the idle task. The observable inputs $\boldsymbol{x}_t$ are the feature vectors for frame t, which are provided in the dataset [19]. These are complex Zernike moments of pixel change history images: in [20], Xiang et al. have shown that pixel change history (PCH) images are able to capture relevant duration information with better discrimination performance. The mathematical formulation for a PCH image is given by:

$$P_{\varsigma,\tau}(x,y,t) = \begin{cases} \min(P_{\varsigma,\tau}(x,y,t-1) + \frac{255}{\varsigma}, 255) \\ \text{if} D(x,y,t) = 1 \\ \max(P_{\varsigma,\tau}(x,y,t-1) - \frac{255}{\tau}, 0) \\ \text{otherwise} \end{cases} \tag{42}$$

where $P_{\varsigma,\tau}(x,y,t)$ is the PCH in time t for a pixel at (x,y), $D(x,y,t)$ is the binary image indicating the foreground region, ς is an accumulation factor, and τ is a decay factor. By setting appropriate values to ς and τ, we are able to capture pixel-level changes over time.

Similar to the previous experiment, we represented the PCH images by 31 parameters calculated as the norms and angles of the complex Zernike moments.

The moments were calculated in rectangular regions of interest of approximately 15000 pixels in each image to limit the processing burden and allow for real-time feature extraction (performed at a rate of approximately 50-60 fps). As mentioned earlier, the classification task here is very challenging because the idle task appears in both workflows, while tasks 1 and 5 may look similar depending on the point of view of the camera.

In our experiments, we used 20 sequences from each workflow. The total number of frames in each case was approximately 80000. We used the groundtruth annotation provided by the authors of the dataset [19]. We have used cross validation in the following fashion: in each cycle, ten of these workflows were randomly selected and the rest ten were used for testing. We run the same experiment 50 times to account for the effect of random selection of samples. For each one of the learned workflows, we postulated models comprising 4 states, to account for the three task types of each workflow plus the idle task, common to both workflows.

First, we compared the evaluated methods on the grounds of the obtained sequence decoding quality, by application of the Viterbi algorithm. Comparisons to the standard CRF, HMM and pseudo-HMM methods were done, similar to the previous experiment. We experimentally found that three mixture component distributions per model state was the selection giving the best performance for the postulated models, wherever applicable.

The obtained labeling results for WF1 and WF2 are given in Table 3 in the form of confusion matrices for each method. The related classification results are displayed in Table 4. We observe that in most cases the DF-CRF outperforms the competitors, with the HMM usually yielding the worst performance. To validate the statistical significance of the differences between the evaluated methods, we also run the Student-t test on the pairs of results from the methods: (HMM/CRF), (HMM/DF-CRF), and (CRF/DF-CRF). As we observed, in all cases the null hypothesis that the obtained differences are not statistically significant was rejected at the 0.01% significance level, a result strongly indicating a clear difference between the evaluated methods [21].

In addition, we would like to underline that, unlike the previous experiment, the HCRF does not cope well with the classification task, most probably due to the aforementioned challenges. In contrast to that, the DF-CRF offers a staggering improvement in sequence classification and segmentation performance. Indeed, from these experiments, it seems that sequence classification is the procedure which benefits the most from the adoption of a joint supervised sequence segmentation and classification scheme, as performed in the case of the DF-CRF model. This is probably due to the fact that our approach helps the algorithm by-pass the problem that pose the similarities between different tasks, as long as the remaining tasks in the workflows are clearly different.

5. Conclusions

In this paper, we presented the dual-functionality CRF model, a CRF-type model for joint segmentation (decoding) and classification of observed sequences.

		Recognized Label						
		1	2	3	4	5	6	7
True Label	1	.51	.18	.20	.00	.03	.01	.07
	2	.16	.22	.62	.00	.00	.00	.00
	3	.16	.01	.83	.00	.00	.00	.00
	4	.24	.03	.16	.35	.07	.08	.07
	5	.43	.06	.19	.05	.17	.04	.01
	6	.43	.04	.56	.00	.00	.04	.01
	7	.54	.06	.10	.00	.01	.01	.27

Total error = 65.42%

(a) pseudo HMM

		Recognized Label						
		1	2	3	4	5	6	7
True Label	1	.28	.16	.04	.07	.35	.04	.05
	2	.04	.43	.03	.03	.42	.02	.02
	3	.00	.06	.45	.10	.17	.21	.01
	4	.02	.12	.16	.49	.06	.11	.03
	5	.03	.19	.09	.21	.26	.17	.05
	6	.04	.06	.23	.13	.05	.48	.02
	7	.07	.08	.03	.25	.16	.16	.24

Total error = 60.89%

(b) standard CRF

		Recognized Label						
		1	2	3	4	5	6	7
True Label	1	.65	.12	.08	.00	.00	.00	.15
	2	.14	.71	.07	.00	.00	.00	.08
	3	.10	.06	.80	.00	.00	.00	.04
	4	.03	.03	.03	.58	.13	.14	.05
	5	.07	.01	.04	.14	.56	.13	.05
	6	.07	.02	.07	.07	.07	.66	.03
	7	.20	.07	.01	.02	.01	.02	.66

Total error = 33.10%

(c) DF-CRF

Table 3: Confusion matrices for segmentation into tasks in the WR dataset. The results are normalized based on the total number of frames per task, considering all cross-validation runs.

We provided efficient algorithms for model training and inference, and evaluated the efficacy of our method in a real-world application. As we showed, our novel approach outperforms popular related approaches, including simple CRFs, HCRFs, and HMMs, in both sequential data segmentation and classification tasks. It is also notable that this performance improvement is obtained with no sacrifices in terms of the imposed computational costs, since our model imposes computational complexity of the same order of magnitude compared to standard CRF and HCRF models. Therefore, we believe that our method emerges as a compelling alternative to the state-of-the-art in sequential data segmentation and classification, with vast potential in several application domains.

		Recognized Clas	
		1	2
True Class	1	1.00	.00
	2	.21	.79
Total error = 10.66%			

(a) HMM

		Recognized Class	
		1	2
True Class	1	.53	.47
	2	.33	.66
Total error = 40.17%			

(b) H-CRF

		Recognized Class	
		1	2
True Class	1	1.00	0
	2	.153	.846
Total error = 7.66%			

(c) DF-CRF

Table 4: Confusion matrix for classification of workflows in the WR dataset. The results are normalized based on the total number of workflows, considering all cross-validation runs.

Acknowledgement

This work was partially supported by the EU FP7 USEFIL project (Unobtrusive Smart Environments for Independent Living), grant no. 288532.

References

[1] J. Lafferty, A. McCallum, F. Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data, in: Proc. 18th International Conf. on Machine Learning, 2001.

[2] C. Sutton, A. McCallum, An introduction to conditional random fields for relational learning, in: L. Getoor, B. Taskar (Eds.), Introduction to Statistical Relational Learning, MIT Press, 2006.

[3] S. Kumar, M. Hebert, Discriminative random fields, International Journal of Computer Vision 68 (2) (2006) 179–201.

[4] R. McDonald, F. Pereira, Identifying gene and protein mentions in text using conditional random fields, BMC Bioinformatics 6 (Suppl 1) (2005) S6.

[5] F. Sha, F. Pereira, Shallow parsing with conditional random fields, in: Proc. NAACL '03, Vol. 1, 2003, pp. 134–141.

[6] J. Zhang, S. Gong, Action categorization with modified hidden conditional random field, Pattern Recognition 43 (1) (2010) 197–203.

[7] A. McCallum, 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.

[8] F. Peng, F. Feng, A. Mccallum, Chinese segmentation and new word detection using conditional random fields, Science (2004) 562–es.

[9] A. Quattoni, S. Wang, L.-P. Morency, M. Collins, T. Darrell, Hidden conditional random fields, IEEE Trans. Pattern Anal. Mach. Intell. 29 (10) (2007) 1848–1853.

[10] M. Hoai, Z.-Z. Lan, F. De la Torre, Joint segmentation and classification of human actions in video, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2011.

[11] M. Zimmermann, Y. Liu, E. Shriberg, A. Stolcke, A* based joint segmentation and classification of dialog acts in multipary meetings, in: In Proc. ASRU, 2005, pp. 215–219.

[12] F. Picard, E. Lebarbier, M. Hoebeke, G. Rigaill, B. Thiam, S. Robin, Joint segmentation, calling, and normalization of multiple cgh profiles, Biostatisticsdoi:10.1093/biostatistics/kxq076.

[13] S. P. Chatzis, D. I. Kosmopoulos, T. A. Varvarigou, Robust sequential data modeling using an outlier tolerant hidden Markov model, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (9) (2009) 1657–1669.

[14] D. P. Bertsekas, Nonlinear Programming, 2nd Edition, Athena Scientific, 1999.

[15] D. Liu, J. Nocedal, On the limited memory method for large scale optimization, Mathematical Programming B 45 (3) (1989) 503–528.

[16] L. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proceedings of the IEEE 77 (1989) 245–255.

[17] B. Ni, G. Wang, P. Moulin, Rgbd-hudaact: A color-depth video database for human daily activity recognition, in: ICCV Workshops, 2011, pp. 1147–1153.

[18] J. W. Davis, A. F. Bobick, The representation and recognition of human movement using temporal templates, in: CVPR, 1997, pp. 928–934.

[19] A. Voulodimos, D. Kosmopoulos, G. Vasileiou, E. Sardis, V. Anagnostopoulos, C. Lalos, A. Doulamis, T. Varvarigou, A threefold dataset for activity and workflow recognition in complex industrial environments, Multimedia, IEEE 19 (3) (2012) 42 –52.

[20] T. Xiang, S. Gong, Beyond tracking: modelling activity and understanding behaviour, International Journal of Computer Vision 67 (2006) 2006.

[21] G. McLachlan, D. Peel, Finite Mixture Models, Wiley Series in Probability and Statistics, 2000.