Advanced Structured Prediction

Editors:

Sebastian Nowozin Sebastian.Nowozin@microsoft.com
Microsoft Research
Cambridge, CB1 2FB, United Kingdom

Peter V. Gehler pgehler@tuebingen.mpg.de
Max Planck Insitute for Intelligent Systems
72076 Tübingen, Germany

Jeremy Jancsary jermyj@microsoft.com
Microsoft Research
Cambridge, CB1 2FB, United Kingdom

Christoph Lampert chl@ist.ac.at
IST Austria
A-3400 Klosterneuburg, Austria

This is a draft version of the author chapter.

The MIT Press
Cambridge, Massachusetts
London, England

1 Structured Prediction for Event Detection

Minh Hoai minhhoai@robots.ox.ac.uk
University of Oxford
Oxford, UK

Fernando De la Torre ftorre@cs.cmu.edu
Carnegie Mellon University
Pittsburgh, PA, USA

This chapter describes Segment-based SVMs (SegSVMs), a framework for event detection. SegSVMs combine energy-based structured prediction, maximum margin learning, and Bag-of-Words (BoWs) representation. Unlike traditional approaches for event detection based on Dynamic Bayesian Networks, the learning formulation of SegSVMs is convex, and the inference over multiple events can be efficiently done in linear time. Beyond detecting a single event, SegSVMs can be extended to solve two relatively unexplored problems in computer vision: early event detection and sequence labeling of multiple events. We illustrate the benefits of SegSVMs in several computer vision applications namely facial action unit detection, early recognition of hand gestures, early detection of facial expressions, and sequence labeling of human actions.

1.1 Introduction

Event detection (ED) is a cornerstone in many important applications, from video surveillance (Piciarelli et al., 2008) to motion analysis (Aggarwal and Cai, 1999) and psychopathology assessment (Cohn et al., 2009). ED refers to the task of localizing and recognizing the occurrences of temporal patterns that belong to some predefined target classes. Examples of target event classes are human actions (Ke et al., 2005), sport events (Efros

et al., 2003; Xu et al., 2003), and facial expressions (Lucey et al., 2006; Bartlett et al., 2006; Zhu et al., 2009; Valstar and Pantic, 2007). ED is different from and harder than event recognition. ED in continuous time series involves both localization and recognition. Event recognition systems, such as those from Yamato et al. (1992), Brand et al. (1997), Gorelick et al. (2007), Sminchisescu et al. (2005), and Laptev et al. (2008), only need to classify pre-segmented subsequences that correspond to coherent events.

ED in video is a challenging problem. Several highly important challenges are to: (1) accommodate large variability of human behavior across subjects; (2) train classifiers when relatively few examples for each event are present; (3) recognize events with subtle human motion; (4) model the temporal dynamics of events, which can be highly variable; and (5) determine the beginnings and the ends of the events.

Existing approaches for ED are typically based on segment classification or Dynamic Bayesian Networks (DBNs). Segment classification works by classifying candidate temporal segments (e.g., Piciarelli et al. (2008); Vassilakis et al. (2002); Nowozin et al. (2007); Shechtman and Irani (2007)). Although segment classification has been widely used for ED, it has several limitations. First, this approach classifies each candidate segment independently; it makes myopic decisions (Wang et al., 2006) and requires post-processing (e.g., to handle overlapping detections). Second, the segment classification approach often has difficulties for accurate localization of event boundaries (Wang et al., 2006), due to the ineffective use of negative examples in training. Negative examples are segments that misalign with target events, and they are either ignored (e.g., (Shechtman and Irani, 2007; Bobick and Wilson, 1997)) or required to be disjoint from the positive training examples (e.g., (Ke et al., 2005; Laptev and Perez, 2007)). In both cases, segments that partially overlap with positive examples are not used in training; those segments, however, are candidates for inaccurate localization at test time. Another popular approach for ED is to use a variant of DBNs. However, DBNs typically lead to a high-dimensional optimization problem with multiple local minima. Furthermore, generative models such as HMMs and variants, have limited ability to model the *null* class (no event or unseen events) due to the large variability of the null class.

In this chapter, we propose Segment-based SVMs (SegSVMs) to address the limitations of existing ED methods. SegSVMs combine structured prediction, maximum margin learning, and Bag-of-Words (BoW) representation. SegSVMs have several benefits for ED. First, SegSVMs use energy-based structured prediction because detecting semantic events in continuous time series is inherently a structured prediction task. Given a time series, the desired output is more than a binary label indicating the presence or absence

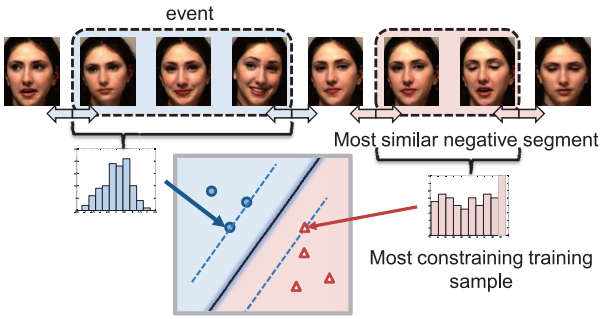


Figure 1.1: During testing, the events are found by efficiently searching over the segments (position and length) that maximize the SVM score. During training, the algorithm searches over all possible negative segments to identify those hardest to classify, which improves classification of subtle events.

of target events. It must predict the locations of target events and their associated class labels, and energy-based structured prediction provides a principled mechanism for concurrent top-down recognition and bottom-up temporal localization (see Fig. 1.1). Second, SegSVMs model temporal events using the BoW representation (Lewis, 1998; Sivic and Zisserman, 2003). The BoW representation requires no state transition model, eliminating the need for detailed annotation and manual definition of event dynamics. This representation can model and detect events of different lengths, removing the necessity of multi-size templates or multi-scale processing. BoW representation is not as rigid as template matching or dynamic time warping; it tolerates errors in misalignment, and it is robust to the impreciseness in human annotation. Finally, SegSVMs are based on the maximum margin training (Taskar et al., 2003; Tsochantaridis et al., 2005), which learns a discriminative model that maximizes the separating margin between different event classes. Maximizing the separating margin yields classifiers that are less prone to over-fitting. Furthermore, the learning formulation of SegSVMs is convex and extendable.

Beyond ED, SegSVMs can be extended to address the problems of early event detection and sequence labeling of multiple events. A temporal event has a duration, and by early detection, we mean to detect the event as soon as possible, *after it starts but before it ends*. Figure 1.2 illustrates the problem of early detection of an smile facial event. While ED has been studied extensively, little attention has been paid to early detection, even in the broader literature of computer vision. In Section 1.3, we will describe an extension of SegSVMs for early event detection, by training them to recognize partial events.

The last section of this chapter presents another extension of SegSVMs

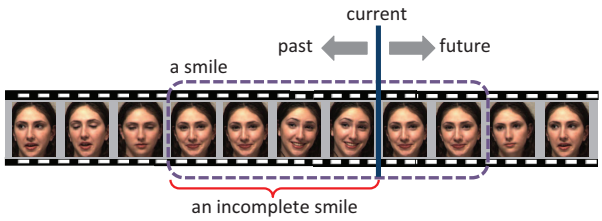


Figure 1.2: How many frames do we need to detect a smile reliably? Can we even detect a smile before it finishes? Existing event detectors are trained to recognize complete events only; they require seeing the entire event for a reliable decision, preventing early detection. We propose a learning formulation to recognize partial events, enabling early detection.

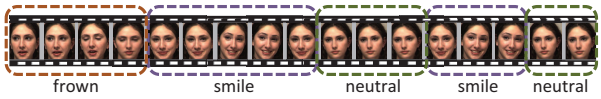


Figure 1.3: Sequence labeling factorizes a time series into a set of non-overlapping segments and recognizes their classes. In this figure, a facial video is labeled as a sequence of expressions.

for sequence labeling of multiple events. Sequence labeling factorizes a time series into a set of non-overlapping segments and assigns a class label to each segment. Recall that sequence labeling system assigns a unique semantic label to each frame, while an ED system may assign none or multiple labels. Figure 1.3 shows an example of sequence labeling. While the problems are slightly different, SegSVMs can be extended to solve the sequence labeling problem too.

1.2 Structured prediction for event detection

This section formulates ED as a structured prediction problem.

1.2.1 Event detection as a structured prediction problem

Consider a time series $\mathbf{X}$, and suppose that we need to detect a target event of which the length is bounded by l_{min} and l_{max}. We denote $\mathcal{Z}(t)$ be the set of length-bounded time intervals from the 1^{st} to the t^{th} frame:

$$\mathcal{Z}(t) = \{[s, e] \in \mathbb{N}^2 | 1 \leq s \leq e \leq t, l_{min} \leq e - s + 1 \leq l_{max}\} \cup \{\emptyset\}.$$

Here $|\cdot|$ is the length function. For a time series $\mathbf{X}$ of length l, $\mathcal{Z}(l)$ (or $\mathcal{Z}$ for brevity) is the set of all possible locations of an event. The empty segment, $\mathbf{z} = \emptyset$, indicates no event occurrence. For an interval $\mathbf{z} = [s, e] \in \mathcal{Z}$, let $\mathbf{X}_\mathbf{z}$ denote the subsegment of $\mathbf{X}$ from frame s to e inclusive.

Let $g(\mathbf{X})$ denote the output of the detector. We will learn the mapping g as in the structured prediction framework (Tsochantaridis et al., 2005; Bakir et al., 2007; Blaschko and Lampert, 2008) as:

$$g(\mathbf{X}) = \underset{\mathbf{z}\in\mathcal{Z}(l)}{\operatorname{argmax}} f(\mathbf{X}_\mathbf{z}; \boldsymbol{\theta}). \tag{1.1}$$

Here, $f(\mathbf{X}_\mathbf{z}; \boldsymbol{\theta})$ is the detection score of segment $\mathbf{X}_\mathbf{z}$, and $\boldsymbol{\theta}$ is the parameter vector of the score function. The output of the detector is defined as the segment that maximizes the detection score. We assume here that each sequence contains at most one occurrence of the event to be detected. This can be extended to k-or-fewer occurrences (Nguyen et al., 2010). The detector searches over all locations and temporal scales from l_{min} to l_{max}. The output of the detector may be the empty segment, and if it is, we report no detection.

1.2.2 Learning and inference

Let $(\mathbf{X}^1, \mathbf{z}^1), \cdots, (\mathbf{X}^n, \mathbf{z}^n)$ be the set of training time series and their associated ground truth annotations for the events of interest. We assume each training sequence contains at most one event of interest, as a training sequence containing several events can always be divided into smaller subsequences of single events. Thus $\mathbf{z}^i = [s^i, e^i]$ consists of two numbers indicating the start and the end of the event in time series $\mathbf{X}^i$.

We consider a linear detection score function, where the detection score is a linear combination of the features:

$$f(\mathbf{X}_\mathbf{z}; \boldsymbol{\theta}) = \begin{cases} \mathbf{w}^T \varphi(\mathbf{X}_\mathbf{z}) + b & \text{if } \mathbf{z} \neq \emptyset, \\ 0 & \text{otherwise.} \end{cases} \tag{1.2}$$

Here, $\varphi(\mathbf{X}_\mathbf{z})$ is the feature vector for segment $\mathbf{X}_\mathbf{z}$ and $\boldsymbol{\theta} = [\mathbf{w}^T, b]$. For brevity, hereafter we use $f(\mathbf{X}_\mathbf{z})$ instead of $f(\mathbf{X}_\mathbf{z}; \boldsymbol{\theta})$ to denote the score of segment $\mathbf{X}_\mathbf{z}$. The function parameters can be learned using Structured Output SVM (SOSVM) (Taskar et al., 2003; Tsochantaridis et al., 2005):

$$\begin{aligned} &\min_{\mathbf{w}, \{\xi^i\}} \frac{1}{2}||\mathbf{w}||^2 + \frac{C}{n}\sum_{i=1}^{n} \xi^i, \\ &\text{s.t. } f(\mathbf{X}^i_{\mathbf{z}^i}) \geq f(\mathbf{X}^i_\mathbf{z}) + \Delta(\mathbf{z}^i, \mathbf{z}) - \xi^i \ \forall \mathbf{z} \in \mathcal{Z} \text{ and } \xi^i \geq 0 \ \forall i. \end{aligned} \tag{1.3}$$

Here, $\Delta(\mathbf{z}^i, \mathbf{z})$ is a loss function that decreases as a label $\mathbf{z}$ approaches the ground truth label $\mathbf{z}^i$. Intuitively, the constraints in Eq. (1.3) force the score of $f(\cdot)$ to be higher for the ground truth label $\mathbf{z}^i$ than for any other value of $\mathbf{z}$, and moreover, to exceed this value by a margin equal to the loss associated with labeling $\mathbf{z}$.

This optimization problem is convex, but it has an exponentially large number of constraints. A typical optimization strategy is *constraint generation* (Tsochantaridis et al., 2005), which is theoretically guaranteed to produce a global optimal solution. Constraint generation is an iterative procedure that optimizes the objective w.r.t. a smaller set of constraints. The constraint set is expanded at every iteration by adding the most violated constraint. Thus at each iteration of constraint generation, given the current value of $\mathbf{w}$, we need to solve:

$$\hat{\mathbf{z}} = \underset{\mathbf{z}\in\mathcal{Z}}{\operatorname{argmax}}\{\Delta(\mathbf{z}^i, \mathbf{z}) + f(\mathbf{X}^i_{\mathbf{z}})\}. \tag{1.4}$$

Thus, for the feasibility of the training phase, it is necessary that (1.4) can be solved effectively and efficiently at every iteration. It is worth noting that this inference problem is different from the one for localizing an event:

$$\hat{\mathbf{z}} = \underset{\mathbf{z}\in\mathcal{Z}}{\operatorname{argmax}}\, f(\mathbf{X}^i_{\mathbf{z}}). \tag{1.5}$$

The optimization of (1.4) & (1.5) depends on the feature representation $\varphi(\mathbf{X}_{\mathbf{z}})$. In the next section, we describe two types of signal representation that render fast optimization.

1.2.3 Segment features using Bag-of-Words representation

We consider the feature mapping $\varphi(\mathbf{X}_{\mathbf{z}})$ as the histogram of temporal words (Nguyen et al., 2009). A temporal dictionary is built by applying a clustering algorithm to a set of feature vectors sampled from the training data (Sivic and Zisserman, 2003). Subsequently, each feature vector is represented by the ID of the corresponding vocabulary entry. Finally, the feature mapping $\varphi(\mathbf{X}_{\mathbf{z}})$ is taken as the histogram of IDs associated with the frames inside the interval $\mathbf{z}$. Let $\mathbf{x}_i$ be the feature vector associated with the i^{th} frame of signal $\mathbf{X}$, and let $\mathcal{C}_j$ denote the cluster j of the temporal dictionary. The feature mapping is defined as:

$$\varphi(\mathbf{X}_{\mathbf{z}}) = [\varphi_1, \cdots, \varphi_d, \text{len}(\mathbf{z})]^T;\ \varphi_j = \sum_{i\in\mathbf{z}} \varphi_{ji};\ \varphi_{ji} = \delta(\mathbf{x}_i \in \mathcal{C}_j). \tag{1.6}$$

Here d is the number of clusters, and $[\varphi_1, \cdots, \varphi_d]^T$ is the histogram of temporal words located within segment $[s, e]$ of signal $\mathbf{X}$.

In this work, instead of using hard quantization where each frame is associated with only one cluster, we propose to use *soft quantization* instead:

$$\varphi(\mathbf{X}_\mathbf{z}) = [\varphi_1, \cdots, \varphi_d, \text{len}(\mathbf{z})]^T;\ \varphi_j = \sum_{i \in \mathbf{z}} \varphi_{ji};\ \varphi_{ji} = k(\mathbf{x}_i, \mathbf{c}_j). \tag{1.7}$$

Here $\{\mathbf{c}_j\}$ are cluster centers, and $k(\cdot, \cdot)$ is the kernel function that measures the similarity between the frame $\mathbf{x}_i$ to the cluster center $\mathbf{c}_j$. φ_j measures the total similarity of the frames inside the segment $\mathbf{z}$ to the cluster center $\mathbf{c}_j$.

Notably, the vectors $\{\mathbf{c}_j\}$ do not need to be the cluster centers. They can be chosen to be any set of representative vectors. For example, $\{\mathbf{c}_j\}$ can be taken as the support vectors of a frame-based SVM trained to distinguish between individual positive and negative frames. In this case, our method directly improves the performance of frame-based SVM by relearning the weights to incorporate temporal constraints. To see this, consider the score function of frame-based SVM. For a frame $\mathbf{x}_i$ of a given signal $\mathbf{X}$, the SVM score is of the form $\mathbf{v}^T\varphi(\mathbf{x}_i) + b$. It has been shown that $\mathbf{v}$ can be expressed as a linear combination of the support vectors: $\mathbf{v} = \sum_{j=1}^{d} \alpha_j \varphi(\mathbf{c}_j)$. Thus the SVM score for frame $\mathbf{x}_i$ is: $\mathbf{v}^T\varphi(\mathbf{x}_i) + b = \sum_{j=1}^{d} \alpha_j k(\mathbf{x}_i, \mathbf{c}_j) + b$. Meanwhile, the decision function of structured learning is: $\mathbf{w}^T\varphi(\mathbf{X}_\mathbf{z}) + b = \sum_{i=s}^{e}\sum_{j=1}^{d} w_j k(\mathbf{x}_i, \mathbf{c}_j) + w_{d+1} \cdot \text{len}(\mathbf{z}) + b$.

For both feature mappings defined in Eq. (1.6) and Eq. (1.7), let a_i denote $\sum_{j=1}^{d} w_j \varphi_{ji} + w_{d+1}$. Thus $\mathbf{w}^T\varphi(\mathbf{X}_\mathbf{z}) = \sum_{i=s}^{e} a_i$. The label $\hat{\mathbf{z}}$ that maximizes $\mathbf{w}^T\varphi(\mathbf{X}_\mathbf{z})$ is: $\hat{\mathbf{z}} = [\hat{s}, \hat{e}] = \text{argmax}_{1 \le s \le e} \sum_{i=s}^{e} a_i$. There exists a linear time algorithm (Nguyen et al., 2009) for this optimization problem. Similarly, the label $\hat{\mathbf{z}}$ that maximizes $\Delta(\mathbf{z}^i, \mathbf{z}) + \mathbf{w}^T\varphi(\mathbf{X}^i_\mathbf{z})$ can be found as:

$$\hat{\mathbf{z}} = [\hat{s}, \hat{e}] = \underset{1 \le s \le e}{\text{argmax}} \left\{ \Delta(\mathbf{z}^i, [s, e]) + \sum_{t=s}^{e} a_t \right\}. \tag{1.8}$$

This can be conveniently solved using exhaustive search, or it can be efficiently optimized by means of a branch-and-bound algorithm (Lampert et al., 2008; Chu et al., 2012).

1.3 Early event detection

The ability to make reliable early detection of temporal events has many potential applications in a wide range of fields, ranging from security (e.g., pandemic attack detection), environmental science (e.g., tsunami warning) to health care (e.g., risk-of-falling detection) and robotics (e.g., affective computing). While temporal ED has been extensively studied, early detec-

tion is a relatively unexplored problem. By early detection, we mean to detect the event as soon as possible, *after it starts but before it ends*, as illustrated in Fig. 1.2. To see why it is important to detect events before they finish, consider a concrete example of building a robot that can affectively interact with humans. Arguably, a key requirement for such a robot is its ability to accurately and rapidly detect human emotional states from facial expressions so that appropriate responses can be made in a timely manner. More often than not, a socially acceptable response is to imitate the current human behavior. This requires facial events such as smiling or frowning to be detected even before they are complete; otherwise, the imitation response would be out of synchronization. However, the learning formulation provided in Sec. 1.2 does not train detectors to recognize partial events. Consequently, using this formulation for Early Event Detection (EED) would lead to unreliable decisions as we will illustrate in the experimental section.

This section proposes Max-Margin Early Event Detectors (MMED), a novel formulation for training event detectors that recognize partial events, enabling early detection. MMED is based on SOSVM (Taskar et al., 2003; Tsochantaridis et al., 2005), but extends it to accommodate the nature of sequential data. In particular, we simulate the sequential frame-by-frame data arrival for training time series and learn an event detector that correctly classifies partially observed sequences. Fig. 1.4 illustrates the key idea behind MMED: partial events are simulated and used as positive training examples. It is important to emphasize that we train a *single* event detector to recognize *all* partial events. But MMED does more than augment the set of training examples; it trains a detector to localize the temporal extent of a target event, even when the target event has not yet finished. This requires monotonicity of the detection function with respect to the inclusion relationship between partial events—the detection score (confidence) of a partial event cannot exceed the score of an encompassing partial event. MMED provides a principled mechanism to achieve this monotonicity, which cannot be assured by a naive solution that simply augments the set of training examples.

1.3.1 Learning with sequential data

To support early detection of events in time series data, we propose to use partial events as positive training examples (Fig. 1.4). In particular, we simulate the sequential arrival of training data as follows. Suppose the length of $\mathbf{X}^i$ is l^i. For each time $t = 1, \cdots, l^i$, let $\mathbf{z}_t^i$ be the part of event $\mathbf{z}^i$ that has already happened, i.e., $\mathbf{z}_t^i = \mathbf{z}^i \cap [1, t]$, which is possibly empty. Ideally, we want the output of the detector on time series $\mathbf{X}^i$ at time t to

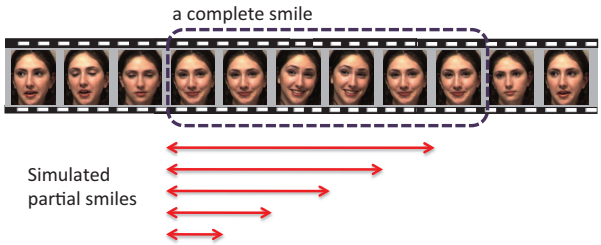


Figure 1.4: Given a training time series that contains a complete event, we simulate the sequential arrival of training data and use partial events as positive training examples. The red segments indicate the temporal extents of the partial events. We train a *single* event detector to recognize *all* partial events, but our method does more than augment the set of training examples.

be the partial event, i.e., $g(\mathbf{X}^i_{[1,t]}) = \mathbf{z}^i_t$. Note that $g(\mathbf{X}^i_{[1,t]})$ is not the output of the detector running on the entire time series $\mathbf{X}^i$. It is the output of the detector on the subsequence of time series $\mathbf{X}^i$ from the first frame to the t^{th} frame only, i.e.,

$$g(\mathbf{X}^i_{[1,t]}) = \underset{\mathbf{z}\in\mathcal{Z}(t)}{\operatorname{argmax}} f(\mathbf{X}^i_{\mathbf{z}}). \tag{1.9}$$

The desired property of the score function is: $f(\mathbf{X}^i_{\mathbf{z}^i_t}) \geq f(\mathbf{X}^i_{\mathbf{z}})\ \forall \mathbf{z} \in \mathcal{Z}(t)$. This constraint requires the score of the partial event $\mathbf{z}^i_t$ to be higher than the score of any other time series segment $\mathbf{z}$ that has been seen in the past, $\mathbf{z} \subset [1, t]$. This is illustrated in Fig. 1.5. Note that the score of the partial event is not required to be higher than the score of a future segment.

As in the case of SOSVM, the previous constraint can be required to be well satisfied by an adaptive margin. This margin is $\Delta(\mathbf{z}^i_t, \mathbf{z})$, the loss of the detector for outputting $\mathbf{z}$ when the desired output is $\mathbf{z}^i_t$ (in our case $\Delta(\mathbf{z}^i_t, \mathbf{z}) = 1 - \frac{2|\mathbf{z}^i_t \cap \mathbf{z}|}{|\mathbf{z}^i_t|+|\mathbf{z}|}$). The desired constraint is: $f(\mathbf{X}^i_{\mathbf{z}^i_t}) \geq f(\mathbf{X}^i_{\mathbf{z}}) + \Delta(\mathbf{z}^i_t, \mathbf{z})\ \forall \mathbf{z} \in \mathcal{Z}(t)$. This constraint should be enforced for all $t = 1, \cdots, l^i$. As in the formulations of SVM, constraints are allowed to be violated by introducing slack variables, and we obtain the following learning formulation:

$$\underset{\mathbf{w},b,\xi^i\geq 0}{\text{minimize}}\ \frac{1}{2}||\mathbf{w}||^2 + \frac{C}{n}\sum_{i=1}^{n}\xi^i, \tag{1.10}$$

$$\text{s.t. } f(\mathbf{X}^i_{\mathbf{z}^i_t}) \geq f(\mathbf{X}^i_{\mathbf{z}}) + \Delta(\mathbf{z}^i_t, \mathbf{z}) - \frac{\xi^i}{\mu\left(\frac{|\mathbf{z}^i_t|}{|\mathbf{z}^i|}\right)}\ \forall i, \forall t = 1 \cdots l^i, \forall \mathbf{z} \in \mathcal{Z}(t). \tag{1.11}$$

Here $|\cdot|$ denotes the length function, and $\mu\left(\frac{|\mathbf{z}^i_t|}{|\mathbf{z}^i|}\right)$ is a function of the proportion of the event that has occurred at time t. $\mu\left(\frac{|\mathbf{z}^i_t|}{|\mathbf{z}^i|}\right)$ is a slack vari-

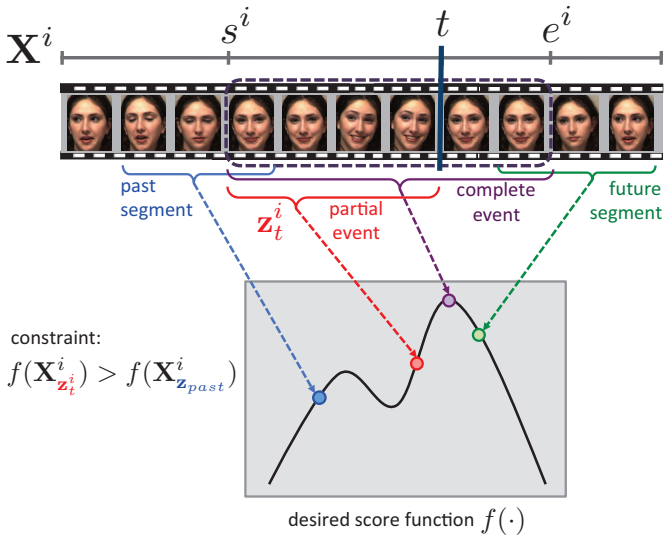


Figure 1.5: The desired score function for early event detection: the complete event must have the highest detection score, and the detection score of a partial event must be higher than that of any segment that ends before the partial event. To learn this function, we explicitly consider partial events during training. At time t, the score of the partial event is required to be higher than the score of any past segment; however, it is not required to be higher than the score of any future segment.

able rescaling factor and should correlate with the importance of correctly detecting at time t whether the event $\mathbf{z}^i$ has happened. $\mu(\cdot)$ can be any arbitrary non-negative function, and in general, it should be a non-decreasing function in $(0, 1]$. In our experiments, we found the following piece-wise linear function a reasonable choice: $\mu(0) = 1$; $\mu(x) = 0$ for $0 < x \leq \alpha$; $\mu(x) = (x - \alpha)/(\beta - \alpha)$ for $\alpha < x \leq \beta$; and $\mu(x) = 1$ for $\beta < x \leq 1$. Here, α and β are tunable parameters. $\mu(0) = \mu(1)$ emphasizes that true rejection is as important as true detection of the complete event.

This learning formulation is an extension of SOSVM. From this formulation, we obtain SOSVM by not simulating the sequential arrival of training data, i.e., to set $t = l^i$ instead of $t = 1, \cdots, l^i$ in Constraint (1.11). Notably, our method does more than augment the set of training examples; it enforces the monotonicity of the detector function, as shown in Fig. 1.6.

For a better understanding of Constraint (1.11), let us analyze the constraint without the slack variable term and break it into three cases: i) $t < s^i$ (event has not started); ii) $t \geq s^i$, $\mathbf{z} = \emptyset$ (event has started; compare the partial event against the detection threshold); and iii) $t \geq s^i$, $\mathbf{z} \neq \emptyset$ (event has started; compare the partial event against any non-empty segment). Recall $f(\mathbf{X}_\emptyset) = 0$ and $\mathbf{z}_t^i = \emptyset$ for $t < s^i$, cases (i), (ii), (iii) lead to

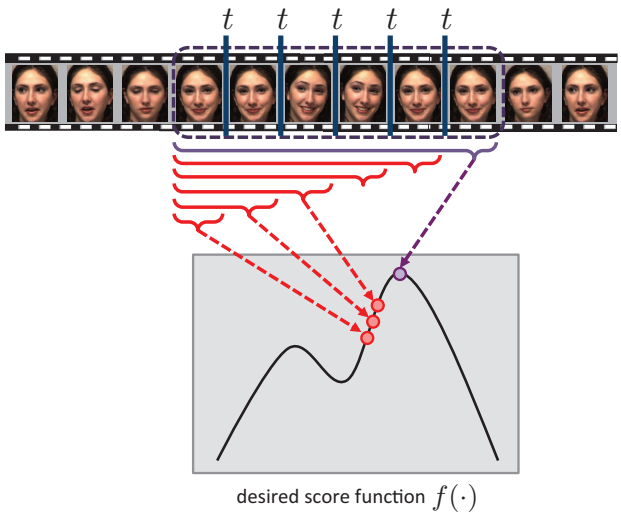


Figure 1.6: Monotonicity requirement – the detection score of a partial event cannot exceed the score of an encompassing partial event. MMED provides a principled mechanism to achieve this monotonicity, which cannot be assured by a naive solution that simply augments the set of training examples.

Constraints (1.12), (1.13), (1.14), respectively:

$$f(\mathbf{X}^i_{\mathbf{z}}) \leq -1 \ \forall \mathbf{z} \in \mathcal{Z}(s^i - 1) \setminus \{\emptyset\}, \tag{1.12}$$

$$f(\mathbf{X}^i_{\mathbf{z}^i_t}) \geq 1 \ \forall t \geq s^i, \tag{1.13}$$

$$f(\mathbf{X}^i_{\mathbf{z}^i_t}) \geq f(\mathbf{X}^i_{\mathbf{z}}) + \Delta(\mathbf{z}^i_t, \mathbf{z}) \ \forall t \geq s^i, \mathbf{z} \in \mathcal{Z}(t) \setminus \{\emptyset\}. \tag{1.14}$$

Constraint (1.12) prevents false detection when the event has not started. Constraint (1.13) requires successful recognition of partial events. Constraint (1.14) trains the detector to accurately localize the temporal extent of the partial events.

The proposed learning formulation Eq. (1.10) is convex, but it contains a large number of constraints. As in Sec. 1.2.2, we propose to use constraint generation in optimization (Tsochantaridis et al., 2005). In our experiments described in Sec. 1.5, constraint generation usually converges within 20 iterations. Each iteration requires minimizing a convex quadratic objective. This objective is optimized using Cplex[1] in our implementation.

1.3.2 Loss function and empirical risk minimization

In Sec. 1.3.1, we have proposed a formulation for training early event detectors. This section provides further discussion on what exactly is being optimized. First, we briefly review the loss of SOSVM and its surrogate empirical risk. We then

1. www-01.ibm.com/software/integration/optimization/cplex-optimizer/

describe two general approaches for quantifying the loss of a detector on sequential data. In both cases, what Eq. (1.10) minimizes is an upper bound on the loss.

As previously explained, $\Delta(\mathbf{z}, \hat{\mathbf{z}})$ is the function that quantifies the loss associated with a prediction $\hat{\mathbf{z}}$, if the true output value is $\mathbf{z}$. Thus, in the setting of offline detection, the loss of a detector $g(\cdot)$ on a sequence-event pair $(\mathbf{X}, \mathbf{z})$ is quantified as $\Delta(\mathbf{z}, g(\mathbf{X}))$. Suppose the sequence-event pairs $(\mathbf{X}, \mathbf{z})$ are generated according to some distribution $P(\mathbf{X}, \mathbf{z})$, the loss of the detector g is

$$\mathcal{R}^{\Delta}_{true}(g) = \int_{\mathcal{X}\times\mathcal{Z}} \Delta(\mathbf{z}, g(\mathbf{X}))dP(\mathbf{X}, \mathbf{z}). \tag{1.15}$$

However, P is unknown so the performance of $g(.)$ is described by the empirical risk on the training data $\{(\mathbf{X}^i, \mathbf{z}^i)\}$, assuming they are generated i.i.d according to P. The empirical risk is $\mathcal{R}^{\Delta}_{emp}(g) = \frac{1}{n}\sum_{i=1}^{n} \Delta(\mathbf{z}^i, g(\mathbf{X}^i))$. It has been shown that SOSVM minimizes an upper bound on the empirical risk $\mathcal{R}^{\Delta}_{emp}$ (Tsochantaridis et al., 2005).

Due to the nature of continual evaluation, quantifying the loss of an online detector on streaming data requires aggregating the losses evaluated throughout the course of the data sequence. Let us consider the loss associated with a prediction $\mathbf{z} = g(\mathbf{X}^i_{[1,t]})$ for time series $\mathbf{X}^i$ at time t as $\Delta(\mathbf{z}^i_t, \mathbf{z})\mu\left(\frac{|\mathbf{z}^i_t|}{|\mathbf{z}^i|}\right)$. Here $\Delta(\mathbf{z}^i_t, \mathbf{z})$ accounts for the difference between the output $\mathbf{z}$ and true truncated event $\mathbf{z}^i_t$. $\mu\left(\frac{|\mathbf{z}^i_t|}{|\mathbf{z}^i|}\right)$ is the scaling factor; it depends on how much the temporal event $\mathbf{z}^i$ has happened. Two possible ways for aggregating these loss quantities is to use their maximum or average. They lead to two different empirical risks for a set of training time series:

$$\mathcal{R}^{\Delta,\mu}_{max}(g) = \frac{1}{n}\sum_{i=1}^{n} \max_t \left\{ \Delta(\mathbf{z}^i_t, g(\mathbf{X}^i_{[1,t]}))\mu\left(\frac{|\mathbf{z}^i_t|}{|\mathbf{z}^i|}\right)\right\},$$
$$\mathcal{R}^{\Delta,\mu}_{mean}(g) = \frac{1}{n}\sum_{i=1}^{n} \operatorname*{mean}_t \left\{ \Delta(\mathbf{z}^i_t, g(\mathbf{X}^i_{[1,t]}))\mu\left(\frac{|\mathbf{z}^i_t|}{|\mathbf{z}^i|}\right)\right\}.$$

In the following, we state and prove a proposition that establishes that the learning formulation given in Eq. (1.10) minimizes an upper bound of the above two empirical risks.

Proposition: Denote by $\boldsymbol{\xi}^*(g)$ the optimal solution of the slack variables in Eq. (1.10) for a given detector g, then $\frac{1}{n}\sum_{i=1}^{n}\xi^{i*}$ is an upper bound on the empirical risks $\mathcal{R}^{\Delta,\mu}_{max}(g)$ and $\mathcal{R}^{\Delta,\mu}_{mean}(g)$.

Proof: Consider Constraint (1.11) with $\mathbf{z} = g(\mathbf{X}^i_{[1,t]})$ and together with the fact that $f(\mathbf{X}^i_{g(\mathbf{X}^i_{[1,t]})}) \geq f(\mathbf{X}^i_{\mathbf{z}^i_t})$, we have $\xi^{i*} \geq \Delta(\mathbf{z}^i_t, g(\mathbf{X}^i_{[1,t]}))\mu\left(\frac{|\mathbf{z}^i_t|}{|\mathbf{z}^i|}\right)\ \forall t$. Thus $\xi^{i*} \geq \max_t\{\Delta(\mathbf{z}^i_t, g(\mathbf{X}^i_{[1,t]}))\mu\left(\frac{|\mathbf{z}^i_t|}{|\mathbf{z}^i|}\right)\}$. Hence $\frac{1}{n}\sum_{i=1}^{n}\xi^{i*} \geq \mathcal{R}^{\Delta,\mu}_{max}(g) \geq \mathcal{R}^{\Delta,\mu}_{mean}(g)$. This completes the proof of the proposition. This proposition justifies the objective of the learning formulation.

1.4 Sequence Labeling

Another important problem in time series analysis is sequence labeling, which factorizes a time series into a set of non-overlapping segments and assigns a class label to each segment. Sequence labeling is related to ED and it is often used for ED. But these two problems are different. A sequence labeling system assigns a unique semantic label to each frame, while an ED system may assign no or multiple labels. Sequence labeling has been shown to be useful in a wide range of applications, from natural language processing (Rabiner, 1989) to office activity understanding (Brand and Kettnaker, 2000) and animal behavior analysis (Oh et al., 2008).

Most existing techniques for sequence labeling are based on probabilistic hidden-state models, and labeling a time series is equivalent to finding the sequence of event labels that yields the highest probability. Brand and Kettnaker (2000) use Hidden Markov Models (HMMs) (Rabiner, 1989) for understanding office activities. Xu et al. (2003) use multi-layer HMMs (Rabiner, 1989) to analyze baseball and volleyball videos. Oh et al. (2008) and Fox et al. (2009) use variants of Switching Linear Dynamical Systems (SLDS) (Pavlovic et al., 2000; Pavlovic and Rehg, 2000) to analyze human and animal behavior. Valstar and Pantic (2007); Koelstra and Pantic (2008); Tong et al. (2007); Shang and Chan (2009); Chang et al. (2009) use Dynamic Bayesian Networks (DBNs) for detecting facial events, while Laxton et al. (2007) design a hierarchical structure based on DBNs to decompose complex activities. Although these generative methods have been shown to be effective in their respective scenarios, they have limited ability to model the null class (i.e., no event, unseen event, or anything that we do not have a label for) due to the large variability of the null class. Conditional Random Fields (CRFs) (Lafferty et al., 2001) are the discriminative alternatives to HMMs, and they have been successfully used for a number of applications such as detection of highlight events in soccer videos (Wang et al., 2006). CRFs, however, cannot model long-range dependencies between labels (Sarawagi and Cohen, 2005), disabling the use of segment-level features. CRFs can be extended to account for higher-order dependencies, but the computational cost increases exponentially with the clique size. Semi-Markov CRFs (Sarawagi and Cohen, 2005) have lower computational cost, but they also require short segment lengths (Okanohara et al., 2006). Nevertheless, CRF-based models, like HMMs or any other hidden-state model, suffer the drawbacks of needing either an explicit definition of the latent state of all frames, or the need to simultaneously learn a state sequence and state transition model that fits the data, resulting in a high-dimensional minimization problem with typically many local minima. This section develops a multi-class extension of Seg-SVMs for sequence labeling, which simultaneously performs temporal segmentation and event recognition in time series.

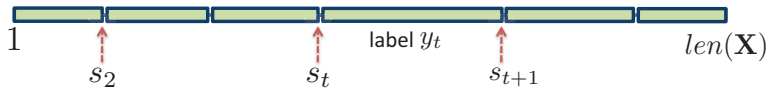


Figure 1.7: Joint segmentation and recognition process – we need to find the events' boundary points $s_1, \cdots, s_{k+1}$ and the class labels $y_1, \cdots, y_k$.

1.4.1 Structured prediction for sequence labeling

Our goal is to factorize a time series into a sequence of events and recognize their classes. Suppose there are m classes of events. We will discuss how to learn the detectors in Section 1.4.2, but assume for now that the detectors $\{\mathbf{w}_j\}_{j=1}^m$ have been learned. These detectors can be used independently to detect each class of target events in turn. This works well for many applications such as facial Action Unit (AU) detection. In many other applications, however, knowledge about the presence or absence of a particular event constrains on those of any other events, just like drinking and kissing do not occur together. This constraint can be incorporated in the joint segmentation and recognition process by finding a set of change points $s_1, \cdots, s_{k+1}$ (see Fig. 1.7) that:

$$\begin{aligned} \underset{k,s_t,y_t,\xi_t \geq 0}{\text{minimize}} \ & \sum_{t=1}^{k} \xi_t, \\ \text{s.t. } & l_{min} \leq \ s_{t+1} - s_t \leq l_{max} \ \forall t, \ s_1 = 0, s_{k+1} = len(\mathbf{X}), \\ & (\mathbf{w}_{y_t} - \mathbf{w}_y)^T \varphi(\mathbf{X}_{(s_t, s_{t+1}]}) \geq 1 - \xi_t \ \forall t, y \neq y_t. \end{aligned} \tag{1.16}$$

Observe that the number of segments k is not known in advance and, therefore, needs to be optimized over. In the above formulation, l_{min} and l_{max} are the minimum and maximum lengths of segments, which can be inferred from training data. Here $\mathbf{X}_{(s_t, s_{t+1}]}$ denotes the segment of time series $\mathbf{X}$, taken from frame $s_t + 1$ to frame s_{t+1} inclusive. $len(\mathbf{X})$ denotes the length of time series $\mathbf{X}$. $\mathbf{w}_y^T \varphi(\mathbf{X}_{(s_t, s_{t+1}]})$ is the SVM score for assigning segment $\mathbf{X}_{(s_t, s_{t+1}]}$ to class y. What we propose is to maximize the difference between the SVM score of the winning class y_t and that of any other class $y \neq y_t$, filtering through the Hinge loss. The idea is to seek a segmentation in which each resulting segment is assigned a class label with high confidence. This is different from what was proposed by Shi et al. (2008), who maximize the total SVM scores:

$$\begin{aligned} \underset{k,s_t,y_t}{\text{maximize}} & \sum_{t=1}^{k} \mathbf{w}_{y_t}^T \varphi(\mathbf{X}_{(s_t, s_{t+1}]}), \text{ s.t.} \\ l_{min} \leq \ & s_{t+1} - s_t \leq l_{max} \ \forall t, \ s_1 = 0, s_{k+1} = len(\mathbf{X}), \end{aligned} \tag{1.17}$$

Different from the above formulation, our segmentation criterion, Eq. (1.16), requires suppressing the non-maximum classes. To see the difference between these two criteria, consider breaking a time series AB in Fig. 1.8 at either M or N. For

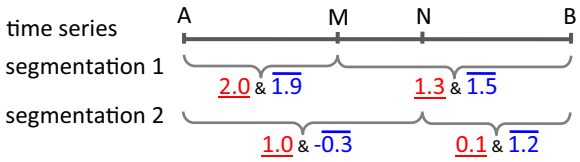


Figure 1.8: Which segmentation is preferred, breaking time series AB at M or N? Suppose there are only two classes, SVM scores of the first and second class for corresponding segments are printed in red and blue, respectively. Our segmentation criterion prefers to cut at N because the resulting segments can be confidently classified.

simplicity, suppose there are only two classes, and the SVM scores of the first and second class for some segments in Figure 1.8 are in printed in underlined and overlined, respectively. The segmentation criterion of Eq. (1.17) would prefer to divide AB at M because it leads to higher total SVM scores of the winning classes (total score of $3.5 = \underline{2.0} + \overline{1.5}$, $\underline{2.0}$ from segment AM and $\overline{1.5}$ from MB). On the other hand, our segmentation criterion does not prefer to cut at M because it cannot confidently classify the resulting segments. To see this, consider the segment AM, even though the SVM score of the winning class, class 1, is high, the SVM score of the alternative, class 2, is also similarly high. Our proposed criterion seeks the optimal segmentation that maximizes the difference between the SVM scores of the winning class and the next best alternative, filtering through the robust Hinge loss. As we will show in Subsection 1.4.2, our segmentation criterion optimizes the same objective as that of the training formulation.

1.4.2 Maximum-margin learning for sequence labeling

We now describe how to learn $\mathbf{w}_1, \cdots, \mathbf{w}_m$ from a collection of training time series $\mathbf{X}^1, \cdots, \mathbf{X}^n$ with known segmentation and class labels, i.e., the change points between actions $0 = s_1^i < \cdots < s_{k_i+1}^i = len(\mathbf{X}^i)$ and the associated class labels $y_1^i, \cdots, y_{k_i}^i \in \{1, \cdots, m\}$ are provided (see Fig. 1.7). We can use multi-class SVM Crammer and Singer (2001) to train a model for temporal actions:

$$\underset{\mathbf{w}_j, \xi_t^i \geq 0}{\text{minimize}} \ \frac{1}{2m} \sum_{j=1}^{m} ||\mathbf{w}_j||^2 + \frac{C}{n} \sum_{i=1}^{n} \sum_{t=1}^{k_i} \xi_t^i, \tag{1.18}$$

$$\text{s.t. } (\mathbf{w}_{y_t^i} - \mathbf{w}_y)^T \varphi(\mathbf{X}^i_{(s_t^i, s_{t+1}^i]}) \geq 1 - \xi_t^i \ \forall i, t, y \neq y_t^i. \tag{1.19}$$

Constraint (1.19) requires segment $\mathbf{X}^i_{(s_t^i, s_{t+1}^i]}$ to belong to class y_t^i with high confidence; in other words, the SVM score for class y_t^i should be relatively higher than that of any other class by a large margin. $\{\xi_t^i\}$ are slack variables which allow for penalized constraint violation. C is the parameter controlling the trade-off between a large margin and less constrained violation.

1.4.3 Dynamic programming algorithm for sequence labeling

Given the parameters $\{\mathbf{w}_j\}_{j=1}^m$, the inference for Eq. (1.16) can be solved using a dynamic programming algorithm, which makes two passes over the time series $\mathbf{X}$. In the forward pass, at frame u ($1 \leq u \leq len(\mathbf{X})$), it computes the best objective value for segmenting and labeling truncated time series $\mathbf{X}_{(0,u]}$ (ignoring frames from $u+1$ onward), i.e.

$$\begin{aligned} h(u) = \min_{k,s_t,y_t,\xi_t \geq 0} & \sum_{t=1}^{k} \xi_t, \\ \text{s.t. } & l_{min} \leq \ s_{t+1} - s_t \leq l_{max} \ \forall t, \ s_1 = 0, s_{k+1} = u, \\ & (\mathbf{w}_{y_t} - \mathbf{w}_y)^T \varphi(\mathbf{X}_{(s_t,s_{t+1}]}) \geq 1 - \xi_t \ \forall t, y \neq y_t. \end{aligned} \tag{1.20}$$

The forward pass computes $h(u)$, as well as $l(u)$, for $u = 1, \cdots, len(\mathbf{X})$ using the recursive formulas:

$$h(u) = \min_{l_{min} \leq l \leq l_{max}} \{\xi(u,l) + h(u-l)\}; \quad l(u) = \operatorname*{argmin}_{l_{min} \leq l \leq l_{max}} \{\xi(u,l) + h(u-l)\}.$$

Here $\xi(u,l)$ denotes the slack value of segment $\mathbf{X}_{(u-l,u]}$, i.e.

$$\xi(u,l) = \max\{0, 1 - (\mathbf{w}_{\hat{y}} - \mathbf{w}_{\tilde{y}})^T \varphi(\mathbf{X}_{(u-l,u]})\}, \tag{1.21}$$

where

$$\hat{y} = \operatorname*{argmax}_y \mathbf{w}_y^T \varphi(\mathbf{X}_{(u-l,u]}), \text{and } \tilde{y} = \operatorname*{argmax}_{y \neq \hat{y}} \mathbf{w}_y^T \varphi(\mathbf{X}_{(u-l,u]}). \tag{1.22}$$

The backward pass of the algorithm finds the best segmentation for $\mathbf{X}$, starting with $s_{k+1} = len(\mathbf{X})$ and using the backward-recursive formula: $s_t = s_{t+1} - l(s_{t+1})$. Once the optimal segmentation has been determined, the optimal assignment of class labels can be found using: $y_t = \operatorname{argmax}_y \mathbf{w}_y^T \varphi(\mathbf{X}_{(s_t,s_{t+1}]})$. The total complexity for the forward and backward passes of this dynamic programming algorithm is $O(m(l_{max} - l_{min} + 1)len(\mathbf{X}))$. This is linear in the length of the time series.

1.5 Experiments

This section describes experimental results on detection of facial Action Units (AUs) from video, early detection of facial expressions and sign language, and sequence labeling of human actions from video.

1.5.1 Detection of facial AUs

This section describes the experiments on detecting AUs in video. The experiments were performed on RU-FACS-1 dataset (Bartlett et al., 2006), a relatively large

corpus of FACS coded videos. Recorded at Rutgers University, subjects were asked to either lie or tell the truth under a false opinion paradigm in interviews conducted by police and FBI members who posed around 13 questions. These interviews resulted in 2.5 minute long continuous 30-fps video sequences containing spontaneous AUs of people of varying ethnicity and sex. Ground truth FACS coding was provided by expert coders. Data from 28 of the subjects was available for our experiments. In particular, we divided this dataset into 17 subjects for training (97000 frames) and 11 subjects for testing (67000 frames).

The AUs for which we present results were selected by requiring at least 100 event occurrences in the available RU-FACS-1 data, resulting in the following set of AUs: $1, 2, 12, 14, 15, 17, 24$. Additionally, to test performance on AU combinations, AU1+2 and AU6+12 were selected due to the large number of occurrences.

Following Zhu et al. (2009), we extracted fixed-scale-and-orientation SIFT descriptors (Lowe, 1999) anchored at several points of interest at the tracked landmarks for frame-level feature representation. Intuitively, the histogram of gradient orientations calculated in SIFT has the potential to capture much of the information that is described in FACS (e.g., the markedness of the naso-labial furrows, the direction and distribution of wrinkles, the slope of the eyebrows). At the same time, the SIFT descriptor has been shown to be robust to illumination changes and small errors in localization.

After the facial components have been tracked in each frame, a normalization step registers each image with respect to an average face (Zhu et al., 2009). An affine texture transformation is applied to each image so as to warp the texture into this canonical reference frame. This normalization provides further robustness to the effects of head motion. Once the texture is warped into this fixed reference, SIFT descriptors are computed around the outer outline of the mouth (11 points for lower face AU) and on the eyebrows (5 for upper face AU). Due to the large number of resulting features (128 by number of points), the dimensionality of the resulting feature vector was reduced using PCA to keep 95% of the energy, obtaining 261 and 126 features for lower face and upper face AU respectively.

We compared our method against a frame-based SVM and dynamic methods using HMM (Rabiner, 1989). The frame-based SVM (Bartlett et al., 2006) (referred to as SVM) is trained to distinguish between positive (AU) negative (non-AU) frames and uses a radial basis kernel $k(\mathbf{x}, \mathbf{z}) = \exp(-\gamma||\mathbf{x} - \mathbf{z}||^2)$. Our method (*SegSVM*) is based on soft-clustering, with the cluster centers are chosen to be the support vectors (SVs) of frame-based SVMs with a radial basis kernel. Because for several AUs the number of SVs can be quite large $(2000 - 4000)$, we apply the idea proposed by Avidan (2003) to reduce the number of SVs for faster training time and better generalization. However, instead of using a greedy algorithm for subset selection, we used LASSO regression (Tibshirani, 1996). In our experiments, the sizes of the reduced SV sets ranges from 100 to 500 SVs.

We also compared the performance of our method with dynamic approaches using HMMs which have been used with success in the facial expression literature (Valstar and Pantic, 2007). In this experiment, we will limit ourselves to a basic generative

HMM model where the observations for each state are modeled as a Gaussian distribution using a full covariance matrix with ridge regularization (i.e., $\hat{\mathbf{\Sigma}} = \mathbf{\Sigma} + \lambda \mathbf{I}$ where $\mathbf{I}$ is the identity matrix), and consider the same feature set used for all other experiments. Two different state mappings where tried resulting in HMM2 and HMM4. HMM2 is a 2-state model, where state-0 corresponds to a neutral face (no AU present) and state-1 corresponds to frames where the AU is present. HMM4 is a 4-state model, where state-0 is mapped to neutral face frames, state-1 corresponds to AU onset frames, state-2 corresponds to peak frames, and state-3 corresponds to offset frames.

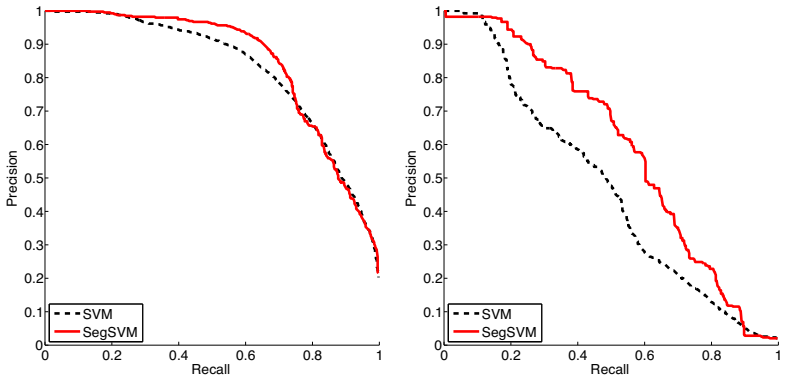
Following Bartlett et al. (2005), positive samples were taken to be frames where the AU was present, and negative samples where it was not. To evaluate performance, we used the precision-recall values and the maximum $F1$ score. The precision and recall measures were computed on a frame-by-frame basis by varying the bias or threshold of the corresponding classifier. The $F1$ score is defined as: $F1 = \frac{2 \cdot Recall \cdot Precision}{Recall + Precision}$, summarizing the trade-off between high recall rates and accuracy among the predictions. $F1$ score is a better performance measure than the more common ROC metric because the latter is designed for balanced binary classification rather than detection tasks, and fails to reflect the effect of the proportion of positive to negative samples on classification performance.

Parameter tuning is done using 3-fold subject-wise cross-validation on the training data. For the frame-based SVM, we need to tune C and γ, the scale parameter of the radial basis kernel. For SegSVM, we need to tune C only. The kernel parameter γ of SegSVM could also potentially be tuned, but for simplicity it was set to the same γ used for frame-based SVM.

Tab. 1.1 shows the experimental results on the RU-FACS-1 dataset. As can be seen, SegSVM, based on structured prediction, consistently outperforms frame-based SVM and HMM, achieving highest $F1$ score on 7 out of 10 test cases. Fig. 1.9 depicts the precision-recall curves of AU12 and AU15. These curves clearly show superior performance for SegSVM. For example, at 70% recall, the precision of SVM and SegSVM are 0.79 and 0.87, respectively. At 50% recall for AU15, the precision of SVM is 0.48 compared to 0.67, roughly $\frac{2}{3}$ that of SegSVM.

	Action Units									
Methods	1	2	6	12	14	15	17	24	1+2	6+12
SVM	0.48	0.42	0.50	0.74	0.20	0.50	0.55	0.15	0.36	0.55
HMM2	0.43	0.42	0.62	0.76	0.18	0.26	0.38	**0.18**	0.31	**0.64**
HMM4	0.39	0.18	**0.63**	0.77	0.12	0.25	0.28	0.05	0.31	0.63
SegSVM	**0.59**	**0.56**	0.59	**0.78**	**0.27**	**0.59**	**0.56**	0.08	**0.56**	0.62

Table 1.1: Max F1-score on the RU-FACS-1 dataset. Higher numbers indicate better performance, and best results are printed in bold.



(a) AU 12 (b) AU 15

Figure 1.9: Precision-recall curves for AU 12 and AU 15. Our method significantly outperforms Frm-SVM.

1.5.2 Early detection of facial expression

The experiment for early detection of facial expression was performed on CK+, the Extended Cohn-Kanade dataset (Lucey et al., 2010). This dataset contains 327 facial image sequences from 123 subjects performing one of seven discrete emotions: anger, contempt, disgust, fear, happiness, sadness, and surprise. Each of the sequences contains images from onset (neutral frame) to peak expression (last frame). We considered the task of detecting negative emotions: anger, disgust, fear, and sadness.

We used the canonical normalized appearance feature, CAPP (Lucey et al., 2010). For comparison purposes, we implemented two frame-based SVMs: *Frm-peak* was trained on peak frames of the training sequences while *Frm-all* was trained using all frames between the onset and offset of the facial action. Frame-based SVMs can be used for detection by classifying individual frames. In contrast, SOSVM and MMED are segment-based. Since a facial expression is a deviation of the neutral face, we represented each segment of an emotion sequence by the difference between the end frame and the start frame. Even though the start frame was not necessarily a neutral face, this representation led to good recognition results.

We used the area under the ROC curve for accuracy comparison and Normalized Time to Detection (NTtoD) for benchmarking the timeliness of detection. The ROC and AMOC curves are defined below.

ROC area: Consider testing a detector on a set of time series. The False Positive Rate (FPR) of the detector is defined as the fraction of time series that the detector fires before the event of interest starts. The True Positive Rate (TPR) is defined as the fraction of time series that the detector fires during the event of interest. A detector typically has a detection threshold that can be adjusted to trade off high TPR for low FPR and vise versa. By varying this detection threshold, we can generate a ROC curve, which is a function of TPR against FPR. We used the area

Figure 1.10: Disgust (a) and fear (b) detection on CK+ dataset. From left to right of each sequence are the onset frame, the frame at which MMED fires, the frame at which SOSVM fires, and the peak frame. The number in each image is the corresponding NTtoD.

under the ROC for evaluating the detector accuracy.

AMOC curve: To evaluate the timeliness of detection we use Normalized Time to Detection (NTtoD) which is defined as follows. Given a testing time series where the event of interest occurs from s to e, suppose the detector starts to fire at time t. For a successful detection, $s \leq t \leq e$, we define the NTtoD as the fraction of event that has occurred, i.e., $\frac{t-s+1}{e-s+1}$. NTtoD is defined as 0 for a false detection ($t < s$) and ∞ for a false rejection ($t > e$). By adjusting the detection threshold, one can achieve smaller NTtoD at the cost of higher FPR and vice versa. For a complete characteristic picture, we vary the detection threshold and plot the curve of NToD versus FPR. This is referred as the Activity Monitoring Operating Curve (AMOC) (Fawcett and Provost, 1999).

We randomly divided the data into disjoint training and testing subsets. The training set contained 200 sequences with equal numbers of positive and negative examples. For reliable results, we repeated our experiment 20 times and recorded the average performance. Regarding the detection accuracy, segment-based SVMs outperformed frame-based SVMs. The ROC areas (mean and standard deviation) for Frm-peak, Frm-all, SOSVM, MMED are 0.82 ± 0.02, 0.84 ± 0.03, 0.96 ± 0.01, and 0.97 ± 0.01, respectively. Comparing the timeliness of detection, our method was significantly better than the others, especially at low false positive rate which is what we care about. For example, at 10% false positive rate, Frm-peak, Frm-all, SOSVM, and MMED can detect the expression when it completes 71%, 64%, 55%, and 47% respectively. Fig. 1.11a plots the AMOC curves, and Fig. 1.10 displays some qualitative results. We used a linear SVM with $C = 1000, \alpha = 0, \beta = 0.5$.

1.5.3 Early detection of sign language

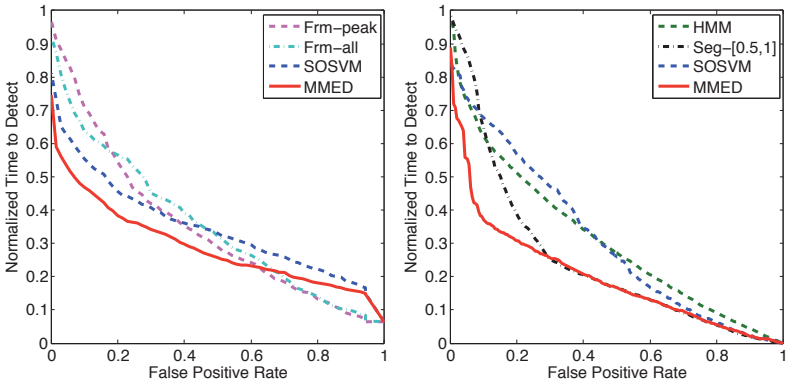
This section describes our experiments on a publicly available dataset (Kadous, 2002) that contains 95 Auslan signs, each with 27 examples. The signs were captured from a native signer using position trackers and instrumented gloves; the location of the two hands, the orientation of the palms, and the bending of the fingers were recorded. We considered detecting the sentence "I love you" in monologues obtained by concatenating multiple signs. In particular, each monologue contained an I-love-you sentence which was preceded and succeeded by 15 random signs. The I-love-you sentence was ordered concatenation of random samples of three signs: "I", "love", and "you". We created 100 training and 200 testing monologues from disjoint sets of sign samples; the first 15 examples of each sign were used to create training monologues while the last 12 examples were used for testing monologues. The average lengths and standard deviations of the monologues and the I-love-you sentences were 1836 ± 38 and 158 ± 6 respectively.

Previous work (Kadous, 2002) reported high recognition performance on this dataset using Hidden Markov Models (HMMs) (Rabiner, 1989). Following their success, we implemented a continuous density HMM for I-love-you sentences. Our HMM implementation consisted of 10 states, each was a mixture of 4 Gaussians. To use the HMM for detection, we adopted a sliding window approach; the window size was fixed to the average length of the I-love-you sentences.

Inspired by the high recognition rate of HMM, we constructed feature representation for SVM-based detectors (SOSVM and MMED) as follows. We first trained a Gaussian Mixture Model of 20 Gaussians for the frames extracted from the I-love-you sentences. Each frame was then associated with a 20×1 log-likelihood vector. We retained the top three values of this vector, zeroing out the other values, to create a frame-level feature representation. This is the soft quantization approach. To compute the feature vector for a given window, we divided the window into two roughly equal halves, the mean feature vector of each half was then calculated, and the concatenation of these mean vectors was then used as the feature representation of the window.

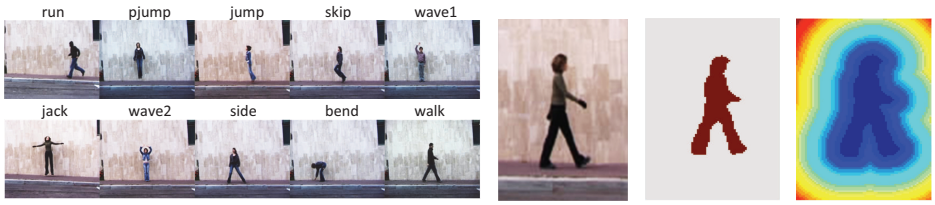
A naive strategy for early detection is to use truncated events as positive examples. For comparison, we implemented *Seg-[0.5,1]*, a binary SVM that used the first halves of the I-love-you sentences in addition to the full sentences as positive training examples. Negative training examples were random segments that had no overlapping with the I-love-you sentences.

We repeated our experiment 10 times and recorded the average performance. Regarding the detection accuracy, all methods except SVM-[0.5,1] performed similarly well. The ROC areas for HMM, SVM-[0.5,1], SOSVM, and MMED were 0.97, 0.92, 0.99, and 0.99, respectively. However, when comparing the timeliness of detection, MMED outperformed the others by a large margin. For example, at 10% false positive rate, our method detected the I-love-you sentence when it observed the first 37% of the sentence. At the same false positive rate, the best alternative method required seeing 62% of the sentence. The full AMOC curves are depicted



(a) CK+, AMOC (b) Auslan, AMOC

Figure 1.11: AMOC curves on Auslan and CK+ datasets; at the same false positive rate, MMED detects target events sooner than the other methods.



(a) Typical frames (b) (c) (d)

Figure 1.12: Weizmann dataset. (b)-(d): how frame-level features are computed; (b): original frame, (c): binary mask, and (d): Euclidean distance transform.

in Fig. 1.11b. In this experiment, we used linear SVM with $C = 1, \alpha = 0.25, \beta = 1$.

1.5.4 Sequence labeling of human actions

The experiments on sequence labeling of human actions were performed on the Weizmann dataset (Gorelick et al., 2007). This dataset contains 90 video sequences (180 × 144 pixels, deinterlaced 50fps) of 9 people, each performing 10 actions. Figure 1.12(a) displays several typical frames extracted from the dataset. Each video sequence in this dataset only consists of a single action.

To evaluate the segmentation and recognition performance of our method, we performed experiments on longer video sequences that were created by concatenating existing single-action sequences. Specifically, we created 9 long sequences, each composed of 10 videos for 10 different actions (each original video sample was used only once). To evaluate the performance of the proposed method in the presence of the null class, background clutter with large variability, we considered the last five classes of actions (side, skip, walk, wave1, and wave2) as the null class. Following Gorelick et al. (2007), we extracted binary masks (Figure 1.12c) and computed

	bend	jack	jump	pjump	run	Null
bend	.96	.01	.01	.00	.00	.01
jack	.00	.97	.00	.01	.00	.02
jump	.00	.00	.88	.06	.04	.02
pjump	.00	.00	.01	.98	.00	.01
run	.00	.00	.01	.00	.91	.08
Null	.01	.03	.00	.03	.03	.90

Table 1.2: Results on Weizmann dataset – Confusion matrix for segmentation and recognition of five different actions: bend, jack, jump, pjump, and run. The null class is the combination of all other classes. The average accuracy is 93.3%.

Euclidean distance transform (Figure 1.12d) for frame-level features. We built a codebook of temporal words with 100 clusters using k-means.

We measured the leave-one-out joint segmentation and recognition performance as follows. We ran our algorithm on long video sequences to find the optimal segmentation and class labels. At that point, each frame was associated with a particular class, and the overall frame-level accuracy against the ground truth labels was calculated as the ratio between the number of agreements over the total number of frames. This evaluation criterion is different from recognition accuracy of algorithms that require pre-segmented video clips (Gorelick et al., 2007).

Table 1.2 shows the confusion matrix for five actions and the null class. Our method yielded the average accuracy of 93.3%. The variant of our method, MaxScoreSeg (Shi et al., 2008), which performed temporal segmentation by maximizing the total SVM scores (Eq. 1.17), obtained an average accuracy of 77.9%. This relatively low accuracy is due to the mismatch between the segmentation criterion and the training objective, as explained in Section 1.4.1. Figure 1.13 displays side-by-side comparison of the prediction result and the human-labeled ground truth. Except for several cases, the majority of error occurs at the boundaries between actions. Error at the boundaries does not necessarily indicate the flaw of our method as human labels are often imperfect (Satkin and Hebert, 2010).

1.6 Summary

This chapter proposed SegSVMs, a structured prediction framework for ED, early ED, and sequence labeling. SegSVMs have convex learning formulations and efficient inference algorithms. We illustrated the benefits of our approaches in a number of existing and new problems in computer vision.

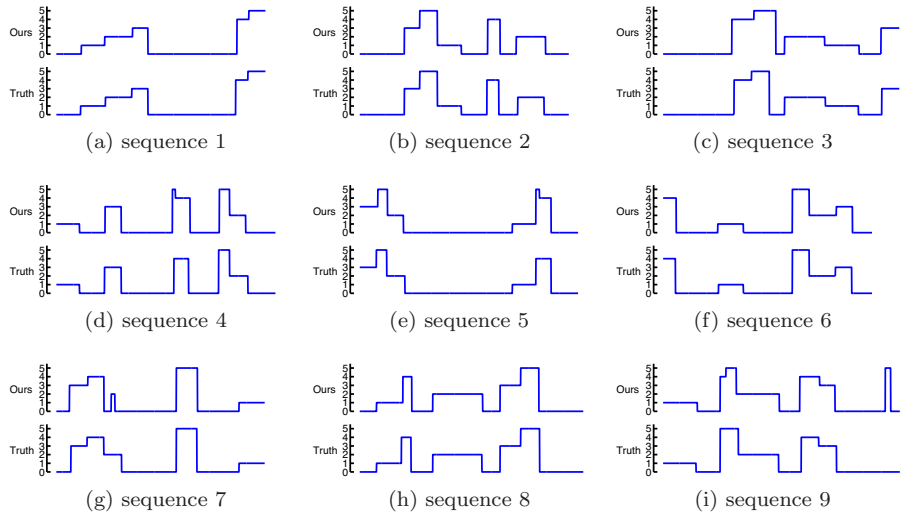


Figure 1.13: Automatic segmentation-recognition versus human-labeled ground truth for Weizmann dataset. The segments at values 0, 1, 2, 3, 4, 5 correspond to null, bend, jack, pjump, jump, run, respectively.

In this chapter, we have addressed the problems of ED, early ED, and sequence labeling using supervised learning. However, other important problems arise in the context of weakly-supervised and unsupervised settings. For instance, in weakly supervised learning, we need to localize the discriminative events from a set of time series annotated with binary labels indicating the presence of the event, but not its location (Nguyen et al., 2009). This has many important applications, e.g., for analyzing times series with or without a particular medical condition. Similarly, unsupervised clustering of time series is important for learning taxonomies of human behavior (Hoai and De la Torre, 2012a). These tasks can also be formulated as extensions of SegSVMs, and we refer the reader to (Nguyen et al., 2009; Hoai et al., 2011; Hoai and De la Torre, 2012a,b) for more details.

Acknowledgements

This work was supported by the National Science Foundation (NSF) under Grant No. RI-1116583. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF. The authors would like to thank Jeffrey Cohn and Tomas Simon for their contribution on the experiment 1.5.1. and many helpful discussions.

1.7 References

J. Aggarwal and Q. Cai. Human motion analysis: A review. *Computer Vision and Image Understanding*, 73(3):428–440, 1999.

S. Avidan. Subset selection for efficient SVM tracking. In *Proc. CVPR*, 2003.

G. Bakir, T. Hofmann, B. Schölkopf, A. Smola, B. Taskar, and S. Vishwanathan, editors. *Predicting Structured Data*. MIT Press, Cambridge, MA, 2007.

M. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, and J. Movellan. Recognizing facial expression: Machine learning and application to spontaneous behavior. In *Computer Vision and Pattern Recognition*, 2005.

M. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, and J. Movellan. Automatic recognition of facial actions in spontaneous expressions. *Journal of Multimedia*, 1(6):22–35, 2006.

M. B. Blaschko and C. H. Lampert. Learning to localize objects with structured output regression. In *Proc. ECCV*, 2008.

A. F. Bobick and A. D. Wilson. A state-based technique for the summarization and recognition of gesture. *IEEE PAMI*, 19(12):1325–1337, 1997.

M. Brand and V. Kettnaker. Discovery and segmentation of activities in video. *IEEE PAMI*, 22(8):844–851, 2000.

M. Brand, N. Oliver, and A. Pentland. Coupled hidden Markov models for complex action recognition. In *Proc. CVPR*, 1997.

K. Chang, T. Liu, and S. Lai. Learning partially-observed hidden conditional random fields for facial expression recognition. In *Computer Vision and Pattern Recognition*, 2009.

W.-S. Chu, F. Zhou, and F. D. la Torre. Unsupervised temporal commonality discovery. In *Proc. ECCV*, 2012.

J. Cohn, T. Simon, I. Matthews, Y. Yang, M. H. Nguyen, M. Tejera, F. Zhou, and F. De la Torre. Detecting depression from facial actions and vocal prosody. In *Proceedings of International Conference on Affective Computing and Intelligent Interaction*, 2009.

K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *J. Machine Learning Research*, 2:265–292, 2001.

A. Efros, A. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *Proc. ICCV*, 2003.

T. Fawcett and F. Provost. Activity monitoring: Noticing interesting changes in behavior. In *Proceedings of the SIGKDD Conference on Knowledge Discovery and Data Mining*, 1999.

E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky. Nonparametric Bayesian learning of switching linear dynamical systems. In *NIPS*. 2009.

L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *IEEE PAMI*, 29(12):2247–2253, 2007.

M. Hoai and F. De la Torre. Maximum margin temporal clustering. In *Proceedings of International Conference on Artificial Intelligence and Statistics*, 2012a.

M. Hoai and F. De la Torre. Max-margin early event detectors. In *Proc. CVPR*, 2012b.

M. Hoai, Z.-Z. Lan, and F. De la Torre. Joint segmentation and classification of

human actions in video. In *Proc. CVPR*, 2011.

M. Kadous. *Temporal classification: Extending the classification paradigm to multivariate time series*. PhD thesis, The University of New South Wales, 2002.

Y. Ke, R. Sukthankar, and M. Hebert. Efficient visual event detection using volumetric features. In *Proc. ICCV*, 2005.

S. Koelstra and M. Pantic. Non-rigid registration using free-form deformations for recognition of facial actions and their temporal dynamics. In *International Conference on Automatic Face and Gesture Recognition*, 2008.

J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. ICML*, 2001.

C. H. Lampert, M. B. Blaschko, and T. Hofmann. Beyond sliding windows: object localization by efficient subwindow search. In *Proc. CVPR*, 2008.

I. Laptev and P. Perez. Retrieving actions in movies. In *Proc. ICCV*, 2007.

I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Proc. CVPR*, 2008.

B. Laxton, J. Lim, and D. Kriegman. Leveraging temporal, contextual and ordering constraints for recognizing complex activities in video. In *Proc. CVPR*, 2007.

D. Lewis. Naive (Bayes) at forty: The independence assumption in information retrieval. In *Proc. ECML*, 1998.

D. Lowe. Object recognition from local scale-invariant features. In *Proc. ICCV*, 1999.

P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews. The extended Cohn-Kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression. In *CVPR Workshop on Human Communicative Behavior Analysis*, 2010.

S. Lucey, I. Matthews, C. Hu, Z. Ambadar, F. De la Torre, and J. Cohn. AAM derived face representations for robust facial action recognition. In *International Conference on Automatic Face and Gesture Recognition*, 2006.

M. H. Nguyen, L. Torresani, F. De la Torre, and C. Rother. Weakly supervised discriminative localization and classification: a joint learning process. In *Proc. ICCV*, 2009.

M. H. Nguyen, T. Simon, F. De la Torre, and J. Cohn. Action unit detection with segment-based SVMs. In *Proc. CVPR*, 2010.

S. Nowozin, G. Bakir, and K. Tsuda. Discriminative subsequence mining for action classification. In *Proc. ICCV*, 2007.

S. M. Oh, J. M. Rehg, T. Balch, and F. Dellaert. Learning and inferring motion patterns using parametric segmental switching linear dynamic systems. *IJCV*, 77(1–3):103–124, 2008.

D. Okanohara, Y. Miyao, Y. Tsuruoka, and J. Tsujii. Improving the scalability of semi-Markov conditional random fields for named entity recognition. In *Proceedings of International Conference on Computational Linguistics*, 2006.

V. Pavlovic and J. M. Rehg. Impact of dynamic model learning on classification of human motion. In *Proc. CVPR*, 2000.

V. Pavlovic, J. M. Rehg, and J. MacCormick. Learning switching linear models of human motion. In *NIPS*, 2000.

C. Piciarelli, C. Micheloni, and G. L. Foresti. Trajectory-based anomalous event

detection. *IEEE Transactions on Circuits and System for Video Technology*, 18 (11):1544–1554, 2008.

L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

S. Sarawagi and W. Cohen. Semi-Markov conditional random fields for information extraction. In *NIPS*, 2005.

S. Satkin and M. Hebert. Modeling the temporal extent of actions. In *Proc. ECCV*, 2010.

L. Shang and K. Chan. Nonparametric discriminant HMM and application to facial expression recognition. In *Conference on Computer Vision and Pattern Recognition*, 2009.

E. Shechtman and M. Irani. Space-time behavior based correlation –or– how to tell if two underlying motion fields are similar without computing them? *IEEE PAMI*, 29(11):2045–2056, 2007.

Q. Shi, L. Wang, L. Cheng, and A. Smola. Discriminative human action segmentation and recognition using semi-Markov model. In *Proc. CVPR*, 2008.

J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proc. ICCV*, 2003.

C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Conditional models for contextual human motion recognition. In *Proc. ICCV*, 2005.

B. Taskar, C. Guestrin, and D. Koller. Max-margin Markov networks. In *NIPS*. 2003.

R. Tibshirani. Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society, Series B*, 58(267–288), 1996.

Y. Tong, W. Liao, and Q. Ji. Facial action unit recognition by exploiting their dynamic and semantic relationships. *Transactions on Pattern Analysis and Machine Intelligence*, pages 1683–1699, 2007.

I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:1453–1484, 2005.

M. Valstar and M. Pantic. Combined support vector machines and hidden markov models for modeling facial action temporal dynamics. In *ICCV Workshop on Human Computer Interaction*, 2007.

H. Vassilakis, A. J. Howell, and H. Buxton. Comparison of feedforward (TDRBF) and generative (TDRGBN) network for gesture based control. In *Proceedings of Revised Papers From the International Gesture Workshop on Gesture and Sign Languages in Human-Computer Interaction*, 2002.

T. Wang, J. Li, Q. Diao, W. Hu, Y. Zhang, and C. Dulong. Semantic event detection using conditional random fields. In *CVPR Workshop*, 2006.

G. Xu, Y.-F. Ma, H.-J. Zhang, and S. Yang. A HMM based semantic analysis framework for sports game event detection. *International Conference on Image Processing*, 2003.

J. Yamato, J. Ohya, and K. Ishii. Recognizing human action in time sequential images using hidden Markov model. In *Proc. CVPR*, 1992.

Y. Zhu, F. De la Torre, and J. Cohn. Dynamic cascades with bidirectional bootstrapping for spontaneous facial action unit detection. In *Affective Computing and Intelligent Interaction*, 2009.