Max-Margin Early Event Detectors

Minh Hoai **Fernando De la Torre**
Robotics Institute, Carnegie Mellon University

Abstract

The need for early detection of temporal events from sequential data arises in a wide spectrum of applications ranging from human-robot interaction to video security. While temporal event detection has been extensively studied, early detection is a relatively unexplored problem. This paper proposes a maximum-margin framework for training temporal event detectors to recognize partial events, enabling early detection. Our method is based on Structured Output SVM, but extends it to accommodate sequential data. Experiments on datasets of varying complexity, for detecting facial expressions, hand gestures, and human activities, demonstrate the benefits of our approach. To the best of our knowledge, this is the first paper in the literature of computer vision that proposes a learning formulation for early event detection.

1. Introduction

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 healthcare (e.g., risk-of-falling detection) and robotics (e.g., affective computing). 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*, as illustrated in Fig. 1. 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 the human emotional states from facial expression 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.

Despite the importance of early detection, few machine learning formulations have been explicitly developed for early detection. Most existing methods (e.g., [5, 13, 16, 10, 14, 9]) for event detection are designed for offline processing. They have a limitation for processing sequential data as they are only trained to detect complete events. But for early detection, it is necessary to recognize partial events, which are ignored in the training process of existing event detectors.

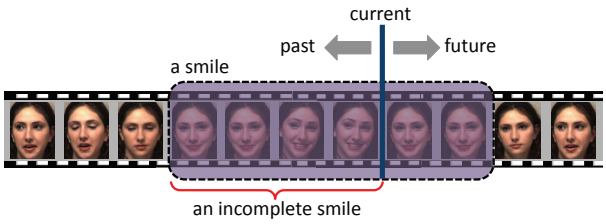


Figure 1. How many frames do we need to detect a smile reliably? Can we even detect a smile before it finishes? Existing event detectors are trained to recognize complete events only; they require seeing the entire event for a reliable decision, preventing early detection. We propose a learning formulation to recognize partial events, enabling early detection.

This paper 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 Structured Output SVM (SOSVM) [17], 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. 2 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 augmenting the set of training examples; it trains a detector to localize the temporal extent of a target event, even when the target event has 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.

The learning formulation of MMED is a constrained quadratic optimization problem. This formulation is the-

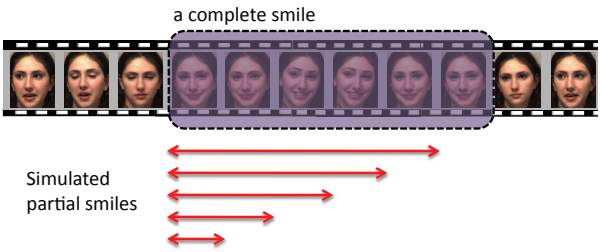


Figure 2. Given a training time series that contains a complete event, we simulate the sequential arrival of training data and use partial events as positive training examples. The red segments indicate the temporal extents of the partial events. We train a *single* event detector to recognize *all* partial events, but our method does more than augmenting the set of training examples.

oretically justified. In Sec. 3.2, we discuss two ways for quantifying the loss for continuous detection on sequential data. We prove that, in both cases, the objective of the learning formulation is to minimize an upper bound of the true loss on the training data.

MMED has numerous benefits. First, MMED inherits the advantages of SOSVM, including its convex learning formulation and its ability for accurate localization of event boundaries. Second, MMED, specifically designed for early detection, is superior to SOSVM and other competing methods regarding the timeliness of the detection. Experiments on datasets of varying complexity, ranging from sign language to facial expression and human actions, showed that our method often made faster detections while maintaining comparable or even better accuracy.

2. Previous work

This section discusses previous work on early detection and event detection.

2.1. Early detection

While event detection has been studied extensively in the literature of computer vision, little attention has been paid to early detection. Davis and Tyagi [2] addressed rapid recognition of human actions using the probability ratio test. This is a passive method for early detection; it assumes that a generative HMM for an event class, trained in a standard way, can also generate partial events. Similarly, Ryoo [15] took a passive approach for early recognition of human activities; he developed two variants of the bag-of-words representation to mainly address the computational issues, not timeliness or accuracy, of the detection process.

Previous work on early detection exists in other fields, but its applicability in computer vision is unclear. Neill *et al.* [11] studied disease outbreak detection. Their approach, like online change-point detection [3], is based on detecting the locations where abrupt statistical changes occur. This technique, however, cannot be applied to detect temporal events such as smiling and frowning, which must and can be detected and recognized independently of the background. Brown *et al.* [1] used the n-gram model for predictive typing, i.e., predicting the next word from previous words. However, it is hard to apply their method to computer vision, which does not have a well-defined language model yet. Early detection has also been studied in the context of spam filtering, where immediate and irreversible decisions must be made whenever an email arrives. Assuming spam messages were similar to one another, Haider *et al.* [6] developed a method for detecting batches of spam messages based on clustering. But visual events such as smiling or frowning cannot be detected and recognized just by observing the similarity between constituent frames, because this characteristic is neither requisite nor exclusive to these events.

It is important to distinguish between forecasting and detection. Forecasting predicts the future while detection interprets the present. For example, financial forecasting (e.g., [8]) predicts the next day's stock index based on the current and past observations. This technique cannot be directly used for early event detection because it predicts the raw value of the next observation instead of recognizing the event class of the current and past observations. Perhaps, forecasting the future is a good first step for recognizing the present, but this two-stage approach has a disadvantage because the former may be harder than the latter. For example, it is probably easier to recognize a partial smile than to predict when it will end or how it will progress.

2.2. Event detection

This section reviews SVM, HMM, and SOSVM, which are among the most popular algorithms for training event detectors. None of them are specifically designed for early detection.

Let $(\mathbf{X}^1, \mathbf{y}^1), \cdots, (\mathbf{X}^n, \mathbf{y}^n)$ be the set of training time series and their associated ground truth annotations for the events of interest. Here 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 $\mathbf{y}^i = [s^i, e^i]$ consists of two numbers indicating the start and the end of the event in time series $\mathbf{X}^i$. Suppose the length of an event is bounded by l_{min} and l_{max} and we denote $\mathcal{Y}(t)$ be the set of length-bounded time intervals from the 1^{st} to the t^{th} frame:

$$\mathcal{Y}(t) = \{\mathbf{y} \in \mathbb{N}^2 | \mathbf{y} \subset [1, t], l_{min} \leq |\mathbf{y}| \leq l_{max}\} \cup \{\emptyset\}.$$

Here $|\cdot|$ is the length function. For a time series $\mathbf{X}$ of length l, $\mathcal{Y}(l)$ is the set of all possible locations of an event; the empty segment, $\mathbf{y} = \emptyset$, indicates no event occurrence. For an interval $\mathbf{y} = [s, e] \in \mathcal{Y}(l)$, let $\mathbf{X}_\mathbf{y}$ denote the subsegment of $\mathbf{X}$ from frame s to e inclusive. Let $g(\mathbf{X})$ denote the output of the detector, which is the segment that maximizes

the detection score:

$$g(\mathbf{X}) = \underset{\mathbf{y} \in \mathcal{Y}(l)}{\operatorname{argmax}} f(\mathbf{X}_{\mathbf{y}}; \theta). \quad (1)$$

The output of the detector may be the empty segment, and if it is, we report no detection. $f(\mathbf{X}_{\mathbf{y}}; \theta)$ is the detection score of segment $\mathbf{X}_{\mathbf{y}}$, and θ is the parameter of the score function. Note that the detector searches over temporal scales from l_{min} to l_{max}. In testing, this process can be repeated to detect multiple target events, if more than one event occur.

How is θ learned? Binary SVM methods learn θ by requiring the score of positive training examples to be greater than or equal to 1, i.e., $f(\mathbf{X}^i_{\mathbf{y}^i}; \theta) \geq 1$, while constraining the score of negative training examples to be smaller than or equal to -1. Negative examples can be selected in many ways; a simple approach is to choose random segments of training time series that do not overlap with positive examples. HMM methods define $f(\cdot, \theta)$ as the log-likelihood and learn θ that maximizes the total log-likelihood of positive training examples, i.e., maximizing $\sum_i f(\mathbf{X}^i_{\mathbf{y}^i}; \theta)$. HMM methods ignore negative training examples. SOSVM methods learn θ by requiring the score of a positive training example $\mathbf{X}^i_{\mathbf{y}^i}$ to be greater than the score of any other segment from the same time series, i.e., $f(\mathbf{X}^i_{\mathbf{y}^i}; \theta) > f(\mathbf{X}^i_{\mathbf{y}}; \theta)\ \forall \mathbf{y} \neq \mathbf{y}^i$. SOSVM further requires this constraint to be well satisfied by a margin: $f(\mathbf{X}^i_{\mathbf{y}^i}; \theta) \geq f(\mathbf{X}^i_{\mathbf{y}}; \theta) + \Delta(\mathbf{y}^i, \mathbf{y})\ \forall \mathbf{y} \neq \mathbf{y}^i$, where $\Delta(\mathbf{y}^i, \mathbf{y})$ is the loss of the detector for outputting $\mathbf{y}$ when the desired output is $\mathbf{y}^i$ [12]. Though optimizing different learning objectives and constraints, all of these aforementioned methods use the same set of positive examples. They are trained to recognize *complete* events only, inadequately prepared for the task of early detection.

3. Max-Margin Early Event Detectors

As explained above, existing methods do not train detectors to recognize partial events. Consequently, using these methods for online prediction would lead to unreliable decisions as we will illustrate in the experimental section. This section derives a learning formulation to address this problem. We use the same notations as described in Sec. 2.2.

3.1. Learning with simulated sequential data

Let $\varphi(\mathbf{X}_{\mathbf{y}})$ be the feature vector for segment $\mathbf{X}_{\mathbf{y}}$. We consider a linear detection score function:

$$f(\mathbf{X}_{\mathbf{y}}; \theta) = \begin{cases} \mathbf{w}^T \varphi(\mathbf{X}_{\mathbf{y}}) + b & \text{if } \mathbf{y} \neq \emptyset, \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

Here $\theta = (\mathbf{w}, b)$, $\mathbf{w}$ is the weight vector and b is the bias term. From now on, for brevity, we use $f(\mathbf{X}_{\mathbf{y}})$ instead of $f(\mathbf{X}_{\mathbf{y}}; \theta)$ to denote the score of segment $\mathbf{X}_{\mathbf{y}}$.

To support early detection of events in time series data, we propose to use partial events as positive training examples (Fig. 2). 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 $\mathbf{y}^i_t$ be the part of event $\mathbf{y}^i$ that has already happened, i.e., $\mathbf{y}^i_t = \mathbf{y}^i \cap [1, t]$, which is possibly empty. Ideally, we want the output of the detector on time series $\mathbf{X}^i$ at time t to be the partial event, i.e.,

$$g(\mathbf{X}^i_{[1,t]}) = \mathbf{y}^i_t. \quad (3)$$

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.,

$$g(\mathbf{X}^i_{[1,t]}) = \underset{\mathbf{y} \in \mathcal{Y}(t)}{\operatorname{argmax}} f(\mathbf{X}^i_{\mathbf{y}}). \quad (4)$$

From (3)-(4), the desired property of the score function is:

$$f(\mathbf{X}^i_{\mathbf{y}^i_t}) \geq f(\mathbf{X}^i_{\mathbf{y}})\ \forall \mathbf{y} \in \mathcal{Y}(t). \quad (5)$$

This constraint requires the score of the partial event $\mathbf{y}^i_t$ to be higher than the score of any other time series segment $\mathbf{y}$ which has been seen in the past, $\mathbf{y} \subset [1, t]$. This is illustrated in Fig. 3. 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 $\Delta(\mathbf{y}^i_t, \mathbf{y})$, the loss of the detector for outputting $\mathbf{y}$ when the desired output is $\mathbf{y}^i_t$ (in our case $\Delta(\mathbf{y}^i_t, \mathbf{y}) = 1 - \frac{2|\mathbf{y}^i_t \cap \mathbf{y}|}{|\mathbf{y}^i_t| + |\mathbf{y}|}$). The desired constraint is:

$$f(\mathbf{X}^i_{\mathbf{y}^i_t}) \geq f(\mathbf{X}^i_{\mathbf{y}}) + \Delta(\mathbf{y}^i_t, \mathbf{y})\ \forall \mathbf{y} \in \mathcal{Y}(t). \quad (6)$$

This constraint should be enforced for all $t = 1, \cdots, l^i$. As in the formulations of SVM and SOSVM, constraints are allowed to be violated by introducing slack variables, and we obtain the following learning formulation:

$$\underset{\mathbf{w}, b, \xi^i \geq 0}{\text{minimize}}\ \frac{1}{2}||\mathbf{w}||^2 + \frac{C}{n}\sum_{i=1}^{n} \xi^i, \quad (7)$$

$$\text{s.t. } f(\mathbf{X}^i_{\mathbf{y}^i_t}) \geq f(\mathbf{X}^i_{\mathbf{y}}) + \Delta(\mathbf{y}^i_t, \mathbf{y}) - \frac{\xi^i}{\mu\left(\frac{|\mathbf{y}^i_t|}{|\mathbf{y}^i|}\right)}$$

$$\forall i, \forall t = 1, \cdots, l^i, \forall \mathbf{y} \in \mathcal{Y}(t). \quad (8)$$

Here $|\cdot|$ denotes the length function, and $\mu\left(\frac{|\mathbf{y}^i_t|}{|\mathbf{y}^i|}\right)$ is a function of the proportion of the event that has occurred at time t. $\mu\left(\frac{|\mathbf{y}^i_t|}{|\mathbf{y}^i|}\right)$ is a slack variable rescaling factor and should correlate with the importance of correctly detecting at time t whether the event $\mathbf{y}^i$ has happened. $\mu(\cdot)$ can be any

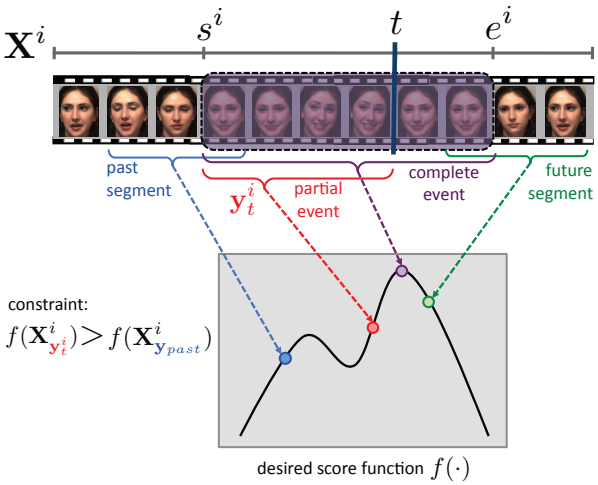


Figure 3. The desired score function for early event detection: the complete event must have the highest detection score, and the detection score of a partial event must be higher than that of any segment that ends before the partial event. To learn this function, we explicitly consider partial events during training. At time t, the score of the truncated event (red segment) is required to be higher than the score of any segment in the past (e.g., blue segment); however, it is not required to be higher than the score of any future segment (e.g., green segment). This figure is best seen in color.

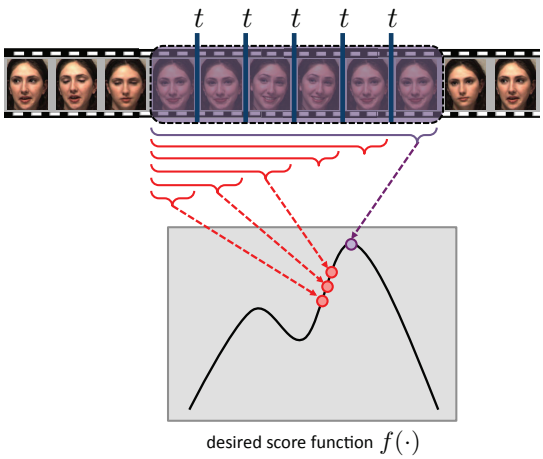


Figure 4. Monotonicity requirement – the detection score of a partial event cannot exceed the score of an encompassing partial event. MMED provides a principled mechanism to achieve this monotonicity, which cannot be assured by a naive solution that simply augments the set of training examples.

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: $\mu(x) = 0$ for $0 < x \leq \alpha$; $\mu(x) = (x - \alpha)/(\beta - \alpha)$ for $\alpha < x \leq \beta$; and $\mu(x) = 1$ for $\beta < x \leq 1$ or $x = 0$. 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 (8). Notably, our method does more than augmenting the set of training examples; it enforces the monotonicity of the detector function, as shown in Fig. 4.

For a better understanding of Constraint (8), 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) $t \geq s^i$, $\mathbf{y} = \emptyset$ (event has started; compare the partial event against the detection threshold); iii) $t \geq s^i$, $\mathbf{y} \neq \emptyset$ (event has started; compare the partial event against any non-empty segment). Recall $f(\mathbf{X}_\emptyset) = 0$ and $\mathbf{y}_t^i = \emptyset$ for $t < s^i$, cases (i), (ii), (iii) lead to Constraints (9), (10), (11), respectively:

$$f(\mathbf{X}^i_{\mathbf{y}}) \leq -1 \; \forall \mathbf{y} \in \mathcal{Y}(s^i - 1) \setminus \{\emptyset\}, \tag{9}$$

$$f(\mathbf{X}^i_{\mathbf{y}_t^i}) \geq 1 \; \forall t \geq s^i, \tag{10}$$

$$f(\mathbf{X}^i_{\mathbf{y}_t^i}) \geq f(\mathbf{X}^i_{\mathbf{y}}) + \Delta(\mathbf{y}_t^i, \mathbf{y}) \; \forall t \geq s^i, \mathbf{y} \in \mathcal{Y}(t) \setminus \{\emptyset\}. \tag{11}$$

Constraint (9) prevents false detection when the event has not started. Constraint (10) requires successful recognition of partial events. Constraint (11) trains the detector to accurately localize the temporal extent of the partial events.

The proposed learning formulation Eq. (7) is convex, but it contains a large number of constraints. Following [17], we propose to use constraint generation in optimization, i.e., we maintain a smaller subset of constraints and iteratively update it by adding the most violated ones. Constraint generation is guaranteed to converge to the global minimum. In our experiments described in Sec. 4, this usually converges within 20 iterations. Each iteration requires minimizing a convex quadratic objective. This objective is optimized using Cplex[1] in our implementation.

3.2. Loss function and empirical risk minimization

In Sec. 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 describe two general approaches for quantifying the loss of a detector on sequential data. In both cases, what Eq. (7) minimizes is an upper bound on the loss.

As previously explained, $\Delta(\mathbf{y}, \hat{\mathbf{y}})$ is the function that quantifies the loss associated with a prediction $\hat{\mathbf{y}}$, if the true output value is $\mathbf{y}$. Thus, in the setting of offline detection, the loss of a detector $g(\cdot)$ on a sequence-event pair $(\mathbf{X}, \mathbf{y})$ is quantified as $\Delta(\mathbf{y}, g(\mathbf{X}))$. Suppose the sequence-event pairs $(\mathbf{X}, \mathbf{y})$ are generated according to some distribution $P(\mathbf{X}, \mathbf{y})$, the loss of the detector g is $\mathcal{R}^{\Delta}_{true}(g) = \int_{\mathcal{X}\times\mathcal{Y}} \Delta(\mathbf{y}, g(\mathbf{X}))dP(\mathbf{X}, \mathbf{y})$. However, P is unknown so the performance of $g(.)$ is described by the empirical risk

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

on the training data $\{(\mathbf{X}^i, \mathbf{y}^i)\}$, assuming they are generated i.i.d according to P. The empirical risk is $\mathcal{R}^{\Delta}_{emp}(g) = \frac{1}{n}\sum_{i=1}^{n}\Delta(\mathbf{y}^i, g(\mathbf{X}^i))$. It has been shown that SOSVM minimizes an upper bound on the empirical risk $\mathcal{R}^{\Delta}_{emp}$ [17].

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 $\mathbf{y} = g(\mathbf{X}^i_{[1,t]})$ for time series $\mathbf{X}^i$ at time t as $\Delta(\mathbf{y}^i_t, \mathbf{y})\mu\left(\frac{|\mathbf{y}^i_t|}{|\mathbf{y}^i|}\right)$. Here $\Delta(\mathbf{y}^i_t, \mathbf{y})$ accounts for the difference between the output $\mathbf{y}$ and true truncated event $\mathbf{y}^i_t$. $\mu\left(\frac{|\mathbf{y}^i_t|}{|\mathbf{y}^i|}\right)$ is the scaling factor; it depends on how much the temporal event $\mathbf{y}^i$ 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:

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

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

In the following, we state and prove a proposition that establishes that the learning formulation given in Eq. 7 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. (7) 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 (8) with $\mathbf{y} = g(\mathbf{X}^i_{[1,t]})$ and together with the fact that $f(\mathbf{X}^i_{g(\mathbf{X}^i_{[1,t]})}) \geq f(\mathbf{X}^i_{\mathbf{y}^i_t})$, we have $\xi^{i*} \geq \Delta(\mathbf{y}^i_t, g(\mathbf{X}^i_{[1,t]}))\mu\left(\frac{|\mathbf{y}^i_t|}{|\mathbf{y}^i|}\right)\ \forall t$. Thus $\xi^{i*} \geq \max_t\{\Delta(\mathbf{y}^i_t, g(\mathbf{X}^i_{[1,t]}))\mu\left(\frac{|\mathbf{y}^i_t|}{|\mathbf{y}^i|}\right)\}$. 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.

4. Experiments

This section describes our experiments on several publicly available datasets of varying complexity.

4.1. Evaluation criteria

This section describes several criteria for evaluating the accuracy and timeliness of detectors. We used the area under the ROC curve for accuracy comparison, Normalized Time to Detection (NTtoD) for benchmarking the timeliness of detection, and $F1$-score for evaluating localization quality.

Area under the ROC curve: 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 the ROC curve which is the function of TPR against FPR. We use the area under the ROC for evaluating the detector accuracy.

AMOC curve: To evaluate the timeliness of detection we used Normalized Time to Detection (NTtoD) which is defined as follows. Given a testing time series with 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 lower NTtoD at the cost of higher FPR and vice versa. For a complete characteristic picture, we varied the detection threshold and plotted the curve of NToD versus FPR. This is referred as the Activity Monitoring Operating Curve (AMOC) [4].

F1-score curve: The ROC and AMOC curves, however, do not provide a measure for how well the detector can localize the event of interest. For this purpose, we propose to use the frame-based $F1$-scores. Consider running a detector on a times series. At time t the detector output the segment $\mathbf{y}$ while the ground truth (possibly) truncated event is $\mathbf{y}^*$. The $F1$-score is defined as the harmonic mean of precision and recall values: $F1 := 2\frac{Precision*Recall}{Precision+Recall}$, with $Precision := \frac{|\mathbf{y}\cap\mathbf{y}^*|}{|\mathbf{y}|}$ and $Recall := \frac{|\mathbf{y}\cap\mathbf{y}^*|}{|\mathbf{y}^*|}$. For a new test time series, we can simulate the sequential arrival of data and record the $F1$-scores as the event of interest unroll from 0% to 100%. We refer to this as the *F1-score curve*.

4.2. Synthetic data

We first validated the performance of MMED on a synthetically generated dataset of 200 time series. Each time series contained one instance of the event of interest, signal 5(a).i, and several instances of other events, signals 5(a).ii–iv. Some examples of these time series are shown in Fig. 5(b). We randomly split the data into training and testing subsets of equal sizes. During testing we simulated the sequential arrival of data and recorded the moment that MMED started to detect the start of the event of interest. With 100% precision, MMED detected the event when it had completed 27.5% of the event. For comparison, SOSVM required observing 77.5% of the event for a positive detection. Examples of testing time series and results are depicted in Fig. 5(b). The events of interest are drawn in

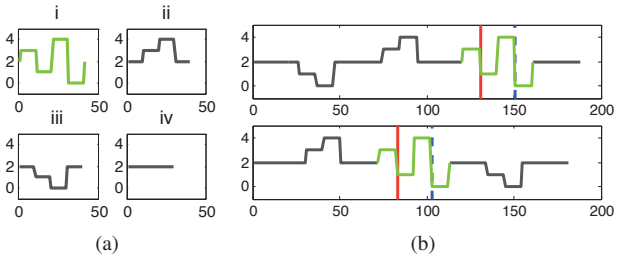


Figure 5. Synthetic data experiment. (a): time series were created by concatenating the event of interest (i) and several instances of other events (ii)–(iv). (b): examples of testing time series; the solid vertical red lines mark the moments that our method starts to detect the event of interest while the dash blue lines are the results of SOSVM.

green and the solid vertical red lines mark the moments that our method started to detect these events. The dash vertical blue lines are the results of SOSVM. Notably, this result reveals an interesting capability of MMED. For the time series in this experiment, the change in signal values from 3 to 1 is exclusive to the target events. MMED was trained to recognize partial events, it implicitly discovered this unique behavior, and it detected the target events as soon as this behavior occurred. In this experiment, we represented each time series segment by the L_2-normalized histogram of signal values in the segment (normalized to have unit norm). We used linear SVM with $C = 1000, \alpha = 0, \beta = 1$.

4.3. Auslan dataset – Australian sign language

This section describes our experiments on a publicly available dataset [7] 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 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 [7] reported high recognition performance on this dataset using HMMs. 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 the 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 often referred to as a 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 calculated, and the concatenation of these mean vectors was used as the feature representation of the window.

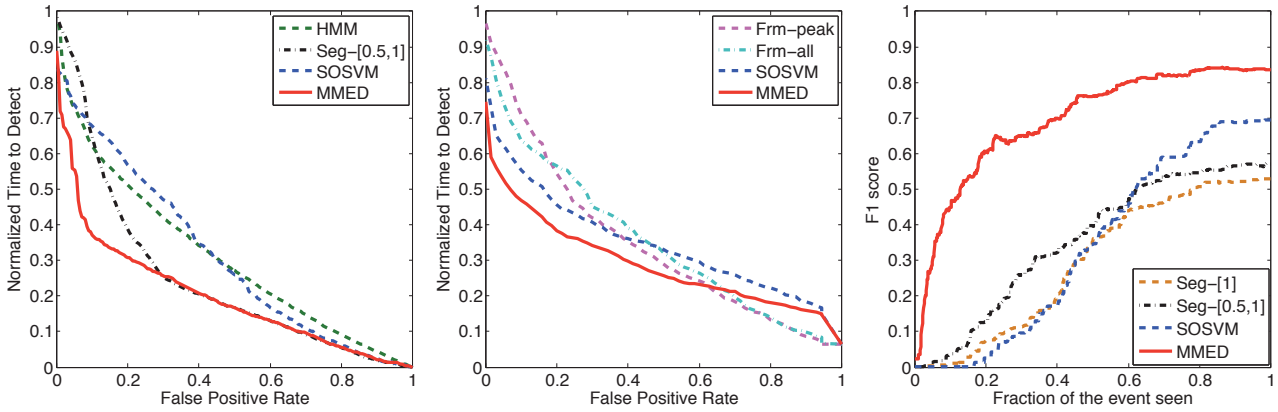
A naive strategy for early detection is to use truncated events as positive examples. For comparison, we implemented *Seg-[0.5,1]*, a binary SVM that used the first halves of the I-love-you sentences in addition to the full sentences as positive training examples. Negative training examples were random 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 in Fig. 6(a). In this experiment, we used linear SVM with $C = 1, \alpha = 0.25, \beta = 1$.

4.4. Extended Cohn-Kanade dataset – expression

The Extended Cohn-Kanade dataset (CK+) [10] 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 same representation as [10], where each frame is represented by the canonical normalized appearance feature, referred as CAPP in [10]. 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



(a) Auslan, AMOC (b) CK+, AMOC (c) Weizmann, F1 curve

Figure 6. Performance curves. (a, b): AMOC curves on Auslan and CK+ datasets; at the same false positive rate, MMED detects the event of interest sooner than the others. (c): F1-score curves on Weizmann dataset; MMED provides better localization for the event of interest, especially when the fraction of the event observed is small. This figure is best seen in color.

contrast, SOSVM and MMED are segment-based. Since a facial expression is a deviation of the neutral expression, 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 necessary a neutral face, this representation led to good recognition results.

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. 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. 6(b) plots the AMOC curves, and Fig. 7 displays some qualitative results. In this experiment, we used a linear SVM with $C = 1000, \alpha = 0, \beta = 0.5$.

4.5. Weizmann dataset – human action

The Weizmann dataset contains 90 video sequences of 9 people, each performing 10 actions. Each video sequence in this dataset only consists of a single action. To measure the accuracy and timeliness of detection, we performed experiments on longer video sequences which were created by concatenating existing single-action sequences. Following [5], we extracted binary masks and computed Euclidean distance transform for frame-level features. Frame-level feature vectors were clustered using k-means to create a codebook of 100 temporal words. Subsequently, each frame

Figure 7. Disgust (a) and fear (b) detection on CK+ dataset. From left to right: 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.

was represented by the ID of the corresponding codebook entry and each segment of a time series was represented by the histogram of temporal words associated with frames inside the segment.

We trained a detector for each action class, but considered them one by one. We created 9 long video sequences, each composed of 10 videos of the same person and had the event of interest at the end of the sequence. We performed leave-one-out cross validation; each cross validation fold trained the event detector on 8 sequences and tested it on the leave-out sequence. For the testing sequence, we computed the normalized time to detection at 0% false positive rate. This false positive rate was achieved by raising the threshold for detection so that the detector would not fire before the event started. We calculated the median normalized time to detection across 9 cross validation folds and averaged these median values across 10 action classes; the resulting values for Seg-[1], Seg-[0.5,1], SOSVM, MMED are 0.16, 0.23, 0.16, and 0.10 respectively. Here Seg-[1] was

a segment-based SVM, trained to classify the segments corresponding to the complete action of interest. Seg-[0.5,1] was similar to Seg-[1], but used the first halves of the action of interest as additional positive examples. For each testing sequence, we also generated a F1-score curve as described in Sec. 4.1. Fig. 6(c) displays the $F1$-score curves of all methods, averaged across different actions and different cross-validation folds. MMED significantly outperformed the other methods. The superiority of MMED over SOSVM was especially large when the fraction of the event observed was small. This was because MMED was trained to detect truncated events while SOSVM was not. Though also trained with truncated events, Seg-[0.5,1] performed relatively poor because it was not optimized to produce correct temporal extent of the event. In this experiment, we used the linear SVM with $C = 1000, \alpha = 0, \beta = 1$.

5. Conclusions

This paper addressed the problem of early event detection. We proposed MMED, a temporal classifier specialized in detecting events as soon as possible. Moreover, MMED provides localization for the temporal extent of the event. MMED is based on SOSVM, but extends it to anticipate sequential data. During training, we simulate the sequential arrival of data and train a detector to recognize incomplete events. It is important to emphasize that we train a *single* event detector to recognize *all* partial events and that our method does more than augmenting the set of training examples. Our method is particularly suitable for events which cannot be reliably detected by classifying individual frames; detecting this type of events requires pooling information from a supporting window. Experiments on datasets of varying complexity, from synthetic data and sign language to facial expression and human actions, showed that our method often made faster detections while maintaining comparable or even better accuracy. Furthermore, our method provided better localization for the target event, especially when the fraction of the seen event was small. In this paper, we illustrated the benefits of our approach in the context of human activity analysis, but our work can be applied to many other domains. The active training approach to detect partial temporal events can be generalized to detect truncated spatial objects [18].

Acknowledgments: 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 Y. Shi for the useful discussion on early detection, L. Torresani for the suggestion of F1 curves, M. Makatchev for the discussion about AMOC, T. Simon for AU data, and P. Lucey for providing CAPP features for the CK+ dataset.

References

[1] P. F. Brown, P. V. deSouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4), 1992.

[2] J. Davis and A. Tyagi. Minimal-latency human action recognition using reliable-inference. *Image and Vision Computing*, 24(5):455–472, 2006.

[3] F. Desobry, M. Davy, and C. Doncarli. An online kernel change detection algorithm. *IEEE Transactions on Signal Processing*, 53(8):2961–2974, 2005.

[4] 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.

[5] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2247–2253, 2007.

[6] P. Haider, U. Brefeld, and T. Scheffer. Supervised clustering of streaming data for email batch detection. In *International Conference on Machine Learning*, 2007.

[7] M. Kadous. *Temporal classification: Extending the classification paradigm to multivariate time series*. PhD thesis, 2002.

[8] K.-J. Kim. Financial time series forecasting using support vector machines. *Neurocomputing*, 55(1-2):307–319, 2003.

[9] T. Lan, Y. Wang, and G. Mori. Discriminative figure-centric models for joint action localization and recognition. In *International Conference on Computer Vision*, 2011.

[10] 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.

[11] D. Neill, A. Moore, and G. Cooper. A Bayesian spatial scan statistic. In *Neural Information Processing Systems*. 2006.

[12] M. H. Nguyen, T. Simon, F. De la Torre, and J. Cohn. Action unit detection with segment-based SVMs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[13] S. M. Oh, J. M. Rehg, T. Balch, and F. Dellaert. Learning and inferring motion patterns using parametric segmental switching linear dynamic systems. *International Journal of Computer Vision*, 77(1–3):103–124, 2008.

[14] A. Patron-Perez, M. Marszalek, A. Zisserman, and I. Reid. High Five: Recognising human interactions in TV shows. In *Proceedings of British Machine Vision Conference*, 2010.

[15] M. Ryoo. Human activity prediction: Early recognition of ongoing activities from streaming videos. In *Proceedings of International Conference on Computer Vision*, 2011.

[16] S. Satkin and M. Hebert. Modeling the temporal extent of actions. In *European Conference on Computer Vision*, 2010.

[17] 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.

[18] A. Vedaldi and A. Zisserman. Structured output regression for detection with partial truncation. In *Proceedings of Neural Information Processing Systems*, 2009.