Elsevier Editorial System(tm) for Pervasive and Mobile Computing
Manuscript Draft

Manuscript Number: PMC-D-08-00114R2

Title: Wearable-Sensor Activity Analysis Using Semi-Markov Models with a Grammar.

Article Type: Research Paper

Corresponding Author: Mr Owen Thomas,

Corresponding Author's Institution:

First Author: Owen Thomas

Order of Authors: Owen Thomas; Owen Thomas; Peter Sunehag; Gideon Dror; Sungrack Yun; Sungwoong Kim; Matthew Robards; Alex Smola; Daniel Green; Philo Saunders

Wearable-Sensor Activity Analysis Using Semi-Markov Models with a Grammar ☆

O. Thomas*,[a], P. Sunehag[a], G. Dror[b], S. Yun[c], S. Kim[c], M. Robards[a], A. Smola[d], D. Green[e], P. Saunders[e]

[a] *Locked Bag 8001, NICTA, Canberra, 2601, ACT, Australia*
[b] *School of Computer Science, The Academic College of Tel-Aviv-Yaffo, Tel Aviv 61083, Israel*
[c] *LG Semicon Hall 2106, KAIST, 373-1, Guseong-dong, Yuseong-gu, Daejeon, 305-701, Republic of Korea*
[d] *Yahoo! Research, Santa Clara, 95050 CA, USA*
[e] *Department of Physiology, Australian Institute of Sport, Belconnen, 2616, ACT, Australia*

Abstract

Detailed monitoring of training sessions of elite athletes is an important component of their training. In this paper we describe an application that performs a precise segmentation and labeling of swimming sessions. This allows a comprehensive break-down of the training session, including lap times, detailed statistics of strokes, and turns. To this end we use semi-Markov models (SMM), a formalism for labeling and segmenting sequential data, trained in a max-margin setting. To reduce the computational complexity of the task and at the same time enforce sensible output, we introduce a grammar into the SMM framework. Using the trained model on test swimming sessions of different swimmers provides highly accurate segmentation as well as perfect labeling of individual segments. The results are significantly better than those achieved by discriminative hidden Markov models.

Key words: activity recognition, semi Markov models, machine learning, human performance, accelerometer

1. Introduction

Recently, motivated by ever increasing levels of interaction between humans and computing devices, context aware computing has received extensive re-

☆The authors acknowledge the staff and athletes at the AIS who contributed to this research.

*Corresponding author
Email addresses: owen.thomas@nicta.com.au (O. Thomas), peter.sunehag@nicta.com.au (P. Sunehag), gideon@mta.ac.il (G. Dror), yunsungrack@kaist.ac.kr (S. Yun), leehwiso@kaist.ac.kr (S. Kim), matthew.robards@nicta.com.au (M. Robards), alex@smola.org (A. Smola), daniel.green@ausport.gov.au (D. Green), philo.saunders@ausport.gov.au (P. Saunders)

search interest. In this broad field, researchers have been developing models and systems for predicting a system's location and status using context aware computing methods [25]. One important facet of this work, and the focus of the research presented in this paper, is that of human activity recognition. The classification of a person's activity, which can be performed using, for example, computer vision [15, 8], has significant potential in diverse application domains such as patient care, chronic disease management and promotion of lifelong health and well-being for the aging population [6]. In more recent times, motivated by developments in the underlying technology, the use of *wearable sensors* for activity recognition has received considerable interest. For example, their use has been investigated in the monitoring of rehabilitation of patients in post-ambulatory conditions [2] and in monitoring Parkinson's disease patients [17].

Wearable sensors have also been investigated extensively for the purpose of gait event detection [30, 26, 27, 9, 13, 10]. Gait event detection involves detection - often in real time - of the phases of gait during walking, which has been of considerable interest in improving quality of life in children with cerebral palsy [27, 13], and in functional electrical stimulation (FES) walking [30, 26]. Of particular interest in this work is the use of accelerometers for the purpose of human activity recognition [30], and the use of machine learning techniques such as the support vector machine - used in [10] for separation of gait phases.

Aside from these applications in health care, sensors have been used in sports and performance arts. This work discusses using wearable sensors technology for the monitoring and analysis of training sessions of professional athletes [11].

The majority of activity recognition research has focused on the task of *recognition*. The sequential data used was either manually segmented or generated in a manner where *segmentation* is not required. For instance, each video sequence represented a single type of activity. In real world scenarios, however, segmentation and recognition are intertwined and therefore they need to be addressed jointly.

In this paper we consider the problem of activity recognition via a combined segmentation and labeling of sports accelerometer data. Given 3-dimensional accelerometer data from a sports training session we describe a method that simultaneously segments the data into atomic actions and labels each action with high accuracy. We further show that the method is robust across multiple athletes; a model trained particular to one athlete performs accurately when applied to another. To perform the labeled segmentation we use a variant of the recent and popular semi-Markov Conditional Random Fields [22] framework. In comparing our results to those achieved by a HMM we demonstrate the superiority of semi-Markov models (SMM) for solving problems of segmentation and by extension, activity recognition.

The paper is structured as follows: In the subsequent section we discuss the data available and segmentation problem in more detail and introduce the HMM and SMM segmentation systems. We then describe the experimental procedure and present the results from experiments comparing the predictive capacity of the HMM and SMM systems. Finally, we finish the paper with a discussion of the two approaches and introduce future work in this area.

2. Method

In this section we will formulate our task, describe our data, introduce the concept of a grammar in this context, provide definitions of two machine learning methods together with the procedures for how to perform training and testing with them. This will provide the components for our general approach to creating an activity recognition system. The approach can be described as follows:

1. Collect video of, and body-worn sensor data from, a person performing the activities of interest.
2. Decide what the exact states of interest are.
3. Formulate a grammar by deciding what state can reasonably follow which other state.
4. Manually create a labeled segmentation of the training data using the video
5. Choose a loss function $\Delta(y, \hat{y})$ which says how much worse a given labeled segmentation y is compared to the truth $\hat{y}$.
6. Train a semi-Markov model using the annotated training data, the grammar and the loss function Δ
7. Now the generated parameters can be used for prediction on new data

2.1. Data

Swimming sensor data was collected from three, eight-lap training sessions from three elite female swimmers at the Australian Institute of Sport, in Canberra, Australia. The three training sessions were representative of their training regime, each consisted of eight laps with strokes consistent across a lap, but potentially alternating between laps (Table 1). Each lap consisted of one of four stroke types: Butterfly, backstroke, breaststroke and freestyle. The data was not evenly distributed (Table 1), for example each swimmer performed 10 laps of freestyle to two laps of butterfly. In each session, 3-dimensional acceleration data was sampled at 100Hz via a Catapult Innovations minimaxX accelerometer attached to the swimmers back. Each session of eight laps comprises approximately $33000 - 34000$ samples, which results in approximately 100000 samples per swimmer. The device is placed in a special pocket on the swim suit that has been designed for the device. This makes the device well attached and, therefore, decreases the noise. To train and evaluate the performance of our method, the data of each was labeled by comparing the sensor read outs to a video of the swimmer in that session. In Figure 2 we show example accelerometer data used in training.

In order to develop the HMM and SMM formalisms in the following sections we now describe the data in more precise terms. We assume we are given a sequence of observations x of length N, such that each observation $x(t)$ specifies a three-tuple $x(t)[0..2]$ containing the 3-dimenstional acceleration at time t. Associated with x we assume a labeled segmentation y, which, given x, is the

Session Name	Butterfly	Backstroke	Breaststroke	Freestyle
Medley	2	2	2	2
Freestyle	0	0	0	8
Freestyle / Backstroke	0	4	0	4

Table 1: Training session descriptions for each of the tree swimmers. Each swimmer performed three training sessions each consisting of eight laps, with each lap containing only one stroke type. Here we show the break down of each session by the number of laps of each stroke type performed in that session.

Figure 1: Markov property between neighboring segments.

quantity we seek to predict. We assume both are generated from a joint distribution $\Pr(x, y)$. To keep notation simple we only present a single sequence x and labeled segmentation y pair, extensions to multiple sequence / segmentation pairs are trivial. We model a labeled segmentation y, for a given observation sequence x, as a sequence of segments, where each segment is represented as a 2-tuple specifying the label of the segment and the coordinate in x at which it begins. That is, for the m'th segment, $y(m) = (l_m, n_m)$, l_m denotes the label for the segment and n_m the coordinate in x it begins at. We enforce that for neighboring segments m and $m + 1$ that $n_{m+1} > n_m$, also that $n_0 = 0$ and that for all m, $n_m < N$. Given neighboring segments m and $m + 1$ we can find the length of segment m via $n_{m+1} - n_m$. We assume that labels l_m are drawn from a set of labels $\mathfrak{L} = 1, \ldots, L$. Figure 1 presents this graphically, as well as the dependency we assume between neighboring segments. We assume that the labels and segment boundaries may depend on their immediate neighbors, as indicated by the red arcs, or possibly longer ranging interactions between them, as indicated by the green arcs which display second order dependencies. It is our goal to find an estimator which is able to estimate an annotation y' given a sequence x' based on the training data (x, y).

2.2. Grammar

In both the HMM and SMM methods described later, we make use of a grammar on label transitions to both improve computational performance (the pertinent algorithms show quadratic complexity in the number of permissible label transitions) and prediction accuracy, via incorporation of prior knowledge. The grammar restricts the labels of segment $m + 1$ conditional on the label of

Figure 2: Two samples of the segmented raw acceleration signal. The left trace depicts butterfly strokes followed by a turn and six strokes of backstroke. The right trace is a close up view of the same turn shown in the left image. The blue, green and red traces correspond to forward, lateral and vertical accelerations, given in units of g, the freefall acceleration. Time is given in units of 10 milliseconds, corresponding to the rate of sampling.

segment m. Such a grammar is clearly evident in the data at hand, the training regime dictates that once the swimmers are swimming a lap they may not switch stroke type until the lap is finished. Also, once a swimmer performs a turn at the end of the lap, they may not immediately turn again. Formally we denote a grammar as a function, G, such that $G(l_m)$ returns the set of permissible labels for segment $m+1$ given the preceding segment label l_m.

2.3. Hidden Markov Models

A Hidden Markov Model (HMM) is a probabilistic graphical model that can be used to define a conditional distribution over a labeling, y, given a sequence of vectors $\mathbf{x}$, formed from a transformation of the input sequence x. We assume the model is parameterized by a weight vector w, the transformation of x into a feature vector representation $\mathbf{x}$ is discussed in Section 2.3.1. Following an approach similar to that used in many, existing sequence analysis problems, we represent the swimming activity as a two level HMM. At the top level the swimmer transitions between different strokes and each stroke type (a label) is decomposed into a series of M individual, label-specific, states, or sub-labels, that must be transitioned through in a specific order, potentially with self transitions (Figure 3). Such a two level system permits application of the Grammar, by restricting certain high level transitions and it permits modeling of the temporal structure of strokes, by different sub-labels within each stroke. We explicitly define the discriminate function, $F(\mathbf{x}, y)$, as the log of the conditional probability of y given $\mathbf{x}$ and w. Formally,

$$\begin{aligned}\hat{y}(\mathbf{x}) &= \underset{y \in \mathcal{Y}(\mathbf{x})}{\operatorname{argmax}} F(\mathbf{x}, y) \\ &= \underset{y \in \mathcal{Y}(\mathbf{x})}{\operatorname{argmax}} \log p(y|\mathbf{x}, w) \\ &= \underset{y \in \mathcal{Y}(\mathbf{x})}{\operatorname{argmax}} \log p(\mathbf{x}|y, w)p(y) \end{aligned} \tag{1}$$

where $p(y)$ is the probability of a label sequence. We estimate one HMM for each label and each HMM consists of several states representing different statistical characteristics as illustrated in Figure 3. A left-to-right HMM is chosen to represent forward progressing states [18]. This means that each label corresponds to a sequence of states that has to be passed through before a higher level transition to a new label can occur. The states represent different phases of the label's signal and the HMM can stay in one phase, i.e. keep transitioning to itself, for a variable length of time.

We assume the feature vector at each position is drawn from a mixture of K Gaussian distributions. The HMM for label l has the parameter set $\{a^l_{jj+1}, c^l_{jk}, \boldsymbol{\mu}^l_{jk}, \boldsymbol{\Lambda}^l_{jk}\}$ where a^l_{jj+1} is the transition probability from state j to state $j+1$, c^l_{jk} is the weight of the kth Gaussian in state j, $\boldsymbol{\mu}^l_{jk}$ is the mean vector of the kth Gaussian in state j, $\boldsymbol{\Lambda}^l_{jk}$ is the covariance matrix of the kth

Figure 3: Left-to-right HMM with both null and emitting states. Each segment $[x_{n_{i-1}+1}, ..., x_{n_i}]$ starts from the initial null state, stays emitting states and ends at the final null state. An observation belongs to an emitting state and feature vectors of each state are described with Gaussian mixture model.

Gaussian in state j. The following constraints hold, $\forall l$, $\forall j$,

$$a^l_{jj+1} + a^l_{jj} = 1, \ \sum_{k=1}^{K} c^l_{jk} = 1 \tag{2}$$

.

Each HMM has a given number of states which is determined by the statistical variation in observations. We assume that the number of Gaussian mixtures is the same for all HMMs.

We can represent $\log p(\mathbf{x}|y, w)$ as

$$\log p(\mathbf{x}|y, w) = \sum_{i=1}^{m} \log p([\mathbf{x}_t]^{t=n_m}_{t=n_{m-1}+1}|l_m, w). \tag{3}$$

The state output probability of $\mathbf{x}_t$ at state s is expressed as

$$p(\mathbf{x}_t|s, l_m, w) = \sum_{k=1}^{K} c_{sk} \mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}^{l_m}_{sk}, \boldsymbol{\Lambda}^{l_m}_{sk}) \tag{4}$$

where $\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Lambda})$ is the Gaussian distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Lambda}$. Since the state information is not given, we compute $\log p([\mathbf{x}_t]^{t=n_m}_{t=n_{m-1}+1}|l_m, w)$ using equation (4) for all possible state sequences as

$$\begin{aligned}
\log p([\mathbf{x}_t]^{t=n_m}_{t=n_{m-1}+1}|l_m, w) &= \log \sum_{\mathbf{s}} p([\mathbf{x}_t]^{t=n_m}_{t=n_{m-1}+1}, \mathbf{s}|l_m, w) \\
&= \log \sum_{\mathbf{s}} p([\mathbf{x}_t]^{t=n_m}_{t=n_{m-1}+1}|\mathbf{s}, l_m, w) p(\mathbf{s}|l_m, w) \\
&= \log \sum_{\mathbf{s}} \left[\prod_{t=n_{m-1}+1}^{t=n_m} p(\mathbf{x}_t|s_t, l_m, w) \right] \left[\prod_{t=n_{m-1}+1}^{t=n_m} a^{l_m}_{s_t s_{t+1}} \right]
\end{aligned} \tag{5}$$

where $\mathbf{s} = s_{n_{m-1}+1}, ..., s_{n_m}$ is the state sequence and the summation is over all possible state sequences. The number of possible state sequences can be extremely large, thus dynamic programming is used to compute the relevant quantities [18]

2.3.1. Features

We transformed the three dimensional accelerometer data into a 12 dimensional feature vector at each position. Let $\tilde{x}_t = [x_t; \log \|x_t\|^2] \in \mathbb{R}^4$ be the

augmented acceleration data at time t by adding the log of the signal energy. It is concatenated by the first and second order regression coefficients (referred to as delta and acceleration coefficients respectively) which are defined as

$$\dot{x}_t = \frac{(\tilde{x}_{t+1} - \tilde{x}_{t-1}) + 2(\tilde{x}_{t+2} - \tilde{x}_{t-2})}{2(1^2 + 2^2)} \tag{6}$$

$$\ddot{x}_t = \frac{(\dot{x}_{t+1} - \dot{x}_{t-1}) + 2(\dot{x}_{t+2} - \dot{x}_{t-2})}{2(1^2 + 2^2)}. \tag{7}$$

The 12-dimensional feature vector x_t is obtained as

$$\mathbf{x}_t = [\tilde{x}_t, \dot{x}_t, \ddot{x}_t]. \tag{8}$$

These features are a common way of introducing temporal information into speech or speaker recognition systems [7, 14] where they are known to enhance accuracy. We have found that they are also useful in our activity recognition task.

2.3.2. Training

The most widely used criterion for estimating the HMM parameter set w is the maximum likelihood (ML) which finds the optimal $\hat{w}$ such that

$$\hat{w} = \underset{w}{\operatorname{argmax}} \log p(\mathbf{x}|y, w). \tag{9}$$

In this case, the quantity to be maximized is a function of incomplete data since there is a missing variable which indicates the sequence of states and mixtures that correspond to $\mathbf{x}$. Thus, we maximize $\log p(\mathbf{x}|y, w)$ by iterating the following expectation and maximization (EM) algorithm: [5]

$$\begin{aligned} \text{E step}: &\quad Q(w|w_{old}) = E[\log p(\mathbf{x}, \mathbf{z}|y, w)|\mathbf{x}, w_{old}] \\ \text{M step}: &\quad w_{new} = \underset{w}{\operatorname{argmax}} Q(w|w_{old}) \end{aligned} \tag{10}$$

where $\mathbf{z}$ is the missing variable. When applied to HMMs, the EM algorithm is referred to as the Baum-Welch algorithm and is discussed in detail in [3].

	Bf	Ba	Br	Fs	Turn	Pre and post Ts.
# of states	10	11	10	11	7	6

Table 2: The number of states for the HMMs. Fewer states are allocated to the HMMs of turns compared with those of strokes due to short durations and rapid variations of signals from turns.

2.4. Semi Markov Models

Given a sequence as described in Section 2.1 we make a crucial assumption on the segmentation and annotation: given the annotation and boundaries of the current activity, the annotations of the earlier and those of the subsequent

activities are independent of each other. That is, for a first order dependency we have

$$p(y|x,(n_i,l_i)) = p(y_{\text{left}}|x,(n_i,l_i))p(y_{\text{right}}|x,(n_i,l_i)) \tag{11}$$

where 'left' and 'right' are defined in respect to the segment with boundary n_i and label l_i. This assumption is sensible since it is unlikely that the segmentation of activities that occurred prior to the current activity should have any bearing on future segments. It has the attractive property that finding such a segmentation is simplified since the dependency graph decomposes into a chain.

Modelling the setting by means of a conditional exponential model, that is by explicitly modelling $p(y|x)$, has the advantage of statistical consistency, however, it has the significant drawback of increased computational requirements, in particular when performing stochastic subgradient descent on long sequences. Moreover, it is difficult to take loss functions explicitly into account. Consequently we choose a maximum margin setting.

To accurately reflect the problem being modeled and to ensure tractable estimation we limit the length of each segment to lie in a finite range of values, dependent upon the segment label. For instance, it is reasonable to assume that a breast stroke will not take more than a few seconds but that it will last for at least one second. This means that labels are associated both with a set of admissible target labels that they may transition to and a range of admissible durations. Formally we state this as

$$(n_{i+1},l_{i+1}) \in S(n_i,l_i) := \{(n,l)|l \in G(l_i) \text{ and } n-n_i \in T(l)\}. \tag{12}$$

Here $G(l)$ is the set of all labels which may succeed a label l and $T(l)$ is the range of admissible durations of l. Finally, we refer to $\mathcal{Y}(x)$ as the set of all labeled segmentations over x which are consistent with the constraint imposed by (12).

We aim to find a discriminant function $F(x,y)$ such that the maximizer of F subject to the constraints of $\mathcal{Y}(x)$ provides a good annotation of the data. That is, we want to find F such that the prediction $\hat{y}(x)$ satisfying

$$\hat{y}(x) := \operatorname*{argmax}_{y \in \mathcal{Y}(x)} F(x,y) \tag{13}$$

is somehow suitable. For computational convenience we restrict ourselves to linear discriminant functions of the form

$$F(x,y) = \langle \Phi(x,y), w \rangle . \tag{14}$$

The problem of finding F becomes one of finding a suitable vector w. Here $\Phi(x,y)$ is a joint feature map which decomposes into a sum over segment specific features, i.e. over features which only depend on each of the segments (n_{i-1},n_i,l_i) via

$$\Phi(x,y) = \sum_{n_i \in y} \phi(x,n_{i-1},n_i,l_i). \tag{15}$$

Recall that l_i indicates the label at segment i. This decomposition is appropriate if we assume that given a segment boundary and a previous segment label, segment interaction is Markovian.

2.4.1. Features

Semi-Markov models permit rich feature definitions for segments. From the preceding section we see that an individual segment feature function, ϕ, is defined over a segment n_i, n_{i+1}. Consequently, features may be defined relative to either coordinate and relative to the entire length of the segment. More concretely, we make use of simple duration invariant features for each label. For each label type we define an identical, 30-dimensional feature vector. Given a segment starting at index p and length l, we take ten means for each channel starting at indices $p, p + \frac{l}{10}, \ldots, p + l - \frac{l}{10}$, over a length of $\frac{l}{10}$. That is, we divide the segment into ten equally spaced bins and we take the mean of each channel in each bin, producing a 30-dimensional feature vector. Linear features defined over a window permit significant reuse of computation during the viterbi procecdue and so increase computational efficiency of training and prediction.

2.4.2. Training

We now introduce and motivate a definition of *suitability* for the parameter vector w in the discriminate function $F(x, y) = \langle \Phi(x, y), w \rangle$ and discuss a method for finding such a w. First through, we begin by introducing a *loss* function, $\Delta(y, \hat{y})$, between a predicted segment, $\hat{y}$ over a sequence x and the corresponding true segmentation for x, y. A loss function is a method for encoding how incorrect $\hat{y}$ is given y and is central to finding a suitable w. In activity segmentation we measure incorrectness by quantifying both label error and segmentation error. That is, we penalize both placing incorrect labels at positions, for instance predicting an incorrect stroke type, and also incorrect segmentation boundaries, for example missing a boundary between strokes or defining a segment to be too long or too short, formally we state

$$\Delta(y, \hat{y}) = \Delta_{\text{label}}(y, \hat{y}) + \Delta_{\text{segment}}(y, \hat{y}). \tag{16}$$

where

$$\Delta_{\text{label}}(y, \hat{y}) = \sum_{i=1}^{N} \delta_{y_i, \hat{y}_i}, \ \Delta_{\text{segment}}(y, \hat{y}) = \sum_{i=1}^{\hat{m}} \min_{1 \le j \le m} |n_j - \hat{n}_i|. \tag{17}$$

Here y_i is the label assigned to position i in the sequence x, and n_i is the boundary where the label switches from l_i to l_{i+1}, as described in Section 2.1. Moreover, $\hat{y}_i, \hat{l}_i, \hat{n}_i, \hat{m}$ all refer to $\hat{y}$. Note that $\Delta(y, \hat{y})$ may be written as a sum over functions $\delta((\hat{n}_{i-1}, \hat{n}_i, \hat{l}_i), y)$, which depend only on the boundaries and labels of a segment, which we shall show later is required for tractable estimation of w.

Given n pairs of sequences and annotations (x^i, y^i) drawn independently and identically from some distribution $\Pr(x, y)$, ideally [29] we seek a w such that

that the expected loss $\mathbf{E}_{x,y}[\Delta(y, F(x,y))]$ is minimized. Since Pr is not available, we settle for a regularized version of the empirical loss instead, choosing $\frac{1}{2}\|w\|^2$ as a regularizer. Moreover, we use the max-margin setting of [28] to provide a convex upper bound on the loss $\Delta(y, \operatorname{argmax}_{y'} F(x,y'))$ in order to render the optimization problem tractable. That is, we aim to minimize:

$$\frac{\lambda}{2}\|w\|^2 + \frac{1}{n}\sum_{i=1}^{n} \max_{y' \in \mathcal{Y}(x^i)} \left\langle \Phi(x^i, y') - \Phi(x^i, y^i), w \right\rangle + \Delta(y^i, y') \tag{18}$$

Here $\lambda > 0$ is a regularization constant to trade off model complexity and empirical risk. It is well known [29] that the argument in the sum majorizes the loss associated with w and that (18) is convex.

We could, in principle, compute the Wolfe dual of the constrained optimization problem associated with (18), however, it is computationally more attractive to solve the problem in primal space. This is particularly true when the computation of subgradients is costly. More to the point, we perform stochastic gradient descent (SGD) on the objective function [16]. Denote by $y^*(x,y)$ a solution of the problem

$$y^*(x, y, w) := \operatorname*{argmax}_{y' \in \mathcal{Y}(x)} \langle \Phi(x, y'), w \rangle + \Delta(y, y'). \tag{19}$$

This is the worst margin violator of the constraints in (18). It follows that the subgradient of any summand of the objective function contains $g_i(w) := \lambda w + \Phi(x^i, y^*(x^i, y^i)) - \Phi(x^i, y^i)$. Given predetermined step sizes $\eta_t > 0$ the SGD update equation is $w_{t+1} = w_t - \eta_t g_t$. We cycle through random permutations of the observations to achieve convergence. Since we use a schedule of the form $\eta_t = \frac{\tau}{t+\tau}$ where $\tau \geq 0$, the step sizes satisfy the Robbins-Monro conditions [21], $\sum \eta_t = \infty$ and $\sum \eta_t^2 < \infty$, which are required for convergence.

2.5. Dynamic Programming

Solving the optimization problem (19) or the problem of maximizing either the SMM or HMM discriminant function with respect to $y \in \mathcal{Y}(x)$ requires dynamic programming. All those problems can be written as

$$\operatorname*{argmax}_{y' \in \mathcal{Y}(x)} \sum_{i=1}^{m} f(n'_{i-1}, n'_i, l'_i) \tag{20}$$

where for the maximization of the SMM discriminant function $F(x,y)$ we have $f(n'_{i-1}, n'_i, l'_i) = \left\langle \phi(n'_{i-1}, n'_i, l'_i, x), w \right\rangle$ and for finding the margin violator we have $f(n'_{i-1}, n'_i, l'_i) = \left\langle \phi(n'_{i-1}, n'_i, l'_i, x), w \right\rangle + \delta((n'_{i-1}, n'_i, l'_i), y)$.

Denote by $V(n,l)$ a score function and let $U(n,l)$ be a tuple of a position and label. In this case we may carry out dynamic programming via

$$U(n,l) := \operatorname*{argmax}_{(n',l') \text{ with } (n,l) \in S(n',l')} V(n',l') + f(n',n,l) \tag{21}$$

$$V(n,l) := \max_{(n',l')} V(n',l') + f(n',n,l). \tag{22}$$

Algorithm 1 SMM Training

INPUTS:

- training data pairs $\{x_i, y_i)\}_i$ of real time series x_i and labelled segmentations y_i
- Initialization parameters w_0
- regularization constant $\lambda > 0$
- Step sizes $\eta_t > 0$
- Features map Φ
- Stopping criteria. E.g. convergence threshold or number of iterations

OUTPUTS:

- Parameters w

1. Let $t = 0$
2. DO:
   (a) Pick a data pair by randomizing i
   (b) Calculate $y^*(x_i, y_i, w_t)$ defined by 19
   (c) Calculate gradient $g_t(w_t) := \lambda w_t + \Phi(x^i, y^*) - \Phi(x^i, y^i)$
   (d) Calculate new parameters through $w_{t+1} = w_t - \eta_t g_t$.
3. WHILE (stopping criteria not met)

Once the recursion reaches $(N, -)$ (here '-' denotes the final null label) we may traverse $U(n, l)$ backwards to obtain a full segmentation of the sequence. An implementation of the recursion (21) requires $O(N \cdot \sum_{l=1}^{L} \sum_{l' \in G(l)} |T(l')|)$ time and $O(N \cdot L)$ space. Here $T(l')$ is the range of the duration of label l'. This shows that the grammar is critical in making an implementation feasible.

For the HMM case $f(n'_{i-1}, n'_i, l'_i) = \log p([\mathbf{x}_{it}]_{t=1}^{t=T_i} | l_i, w) p(y)$ The duration for each turn and stroke is restricted by insertion penalty [24] instead of $S(n, l)$ in this case. The insertion penalty is a fixed value added to each label when it transits from the end of one label to the start of the next.

3. Experiments and results

We performed experiments that assessed the ability of the SMM and HMM to accurately predict swimming segmentations and to generalize across swimmers, the results of which we report in Tables 4 & 5. For each of the three swimmers, we trained one SMM and one HMM upon both the medley and the freestyle / backstroke training sessions of that swimmer. That means that we are using approximately 67000 samples for training each time. Table 3 summarize how many laps of different strokes we have in each training set and how many turns there are. There are approximately 20 strokes (stroke cycles in the case of freestyle and backstroke) in a lap.

	Bf	Ba	Br	Fs	Turn .
#	2	6	2	6	14

Table 3: Number of laps of the different strokes and number of turns in a training set

The performance of each model was subsequently evaluated by predicting segmentations across all sessions for the other two, i.e. non-training, swimmers and comparing these predictions with the true segmentations for that session. In the following section we discuss the experimental configuration of both the SMM and HMM systems introduced earlier, following this we describe the evaluation criteria used to compare the performance of the HMM and SMM systems. We are using the same grammar for both systems. This grammar is simply based on the observation that one of the four strokes can transition either to another stroke of the same type or to a turn. A turn cannot transition to another turn. These assumptions that are implemented through the grammar are encoding our understanding that the data consists of laps of swimming where every lap consists of just one type of swimming and that consecutive laps have a turn between them.

3.1. System configuration

For the SMM system we chose $\lambda = 0.1$ as a regularization parameter and the SGD optimization was run for 300 iterations for each SMM using the rate $\eta_t = \frac{1}{1+t}$, requiring approximately 2 hours for each model when run on a 3Ghz Intel Core2 Duo machine (a MacBook). The training data for each swimmer is approximately 67000 samples long.

The training time is not an important issue since we only train a system once and then we can use it hundreds or thousands of times. Since we show that the system works across swimmers we do not need to redo the procedure for every new swimmer. If a swimmer arrives whose strokes differ substantially from the previous athletes we would be able to retrain using the old parameters as initialization. Performing a prediction, or in other words to generate a labeled segmentation, for a session of eight laps takes approximately 30 seconds on the computer described above.

The device, a minimaxX unit from Catapult Innovations [1], stores sensor data including 3D acceleration sampled at a 100 Hz, in a 1GB internal flash memory. After the swimming session, the data is uploaded through a USB 2.0 interface. Our trained system can then be run on the uploaded data and generate a detailed labeled segmentation from which lap times, lap types and stroke rates can be calculated and displayed in a User Interface.

3.2. Results

Following model training we assessed the generalization ability of both the SMM and HMM approaches. For each of the three trained SMM and HMM mod-

[1] http://www.catapultinnovations.com/minimaxx.html

els we generated six predicted segmentations, one for each of the other two, i.e. non-training, swimmers' freestyle, freestyle / backstroke and medley sessions. In Tables 4 & 5 we report the average segmentation and label error for the HMM and SMM systems. For the labeling error we report the average false positive and false negative rates, which corresponds closely to the SMM loss presented in Section 2.4. Similarly, given a predicted segmentation $\hat{y}$ and true segmentation y we define the segmentation error to be the symmetric difference between the two segmentations, that is, $\frac{1}{|y|}(\sum_{i=1}^{|y|} \min_{j=1}^{|\hat{y}|} |n_i - n_j| + \sum_{i=1}^{|\hat{y}|} \min_{j=1}^{|y|} |n_i - n_j|)$. Note that the segmentation error is slightly different from the loss used in SMM training (Section 2.4). Ideally, in SMM training, we would like to use the symmetric segmentation error as our segmentation loss. This error, however, cannot be decomposed as a sum over individual, predicted segments and so incorporating it would prohibit tractable calculation of equation 19.

	Butterfly	Backstroke	Breaststroke	Freestyle	T.urn
FP (%)	0.07	0.11	0.14	0.06	0.88
FN (%)	0.027	1.1	0.12	0.06	1.2
Seg (ms)	53.5 (± 55.2)	99.0 (± 133)	109 (± 143)	87.1 (± 136)	452 (± 330)

Table 4: Average SMM prediction errors. A summary of the labeling and the segmentation error, detailed for each label. Average labeling errors are quoted as false positive and false negative percentage rates. The standard deviation, taken across entire predicted segmentations, is included in brackets. Segmentation errors are quoted in milliseconds, together with their standard deviations, taken across individual segments.

	Butterfly	Backstroke	Breaststroke	Freestyle	Turn
FP (%)	0.06	0.1	0.2	0.09	0.9
FN (%)	0.09	0.09	0.2	0.09	1.0
Seg (ms)	443 (± 228)	572 (± 368)	916 (± 543)	528 (± 342)	464 (± 301)

Table 5: Average HMM prediction errors, reporting the same statistics as in preceding table.

Both the HMM and the SMM label the underlying accelerometer data with high accuracy, reporting average errors of less than 1 %. The SMM, however, places considerably higher accuracy segment boundaries than the HMM. Both systems perform poorly in estimating turn state boundaries. We speculate this is a consequence of limited training examples for turns and an observed high variability in turn styles. We expect that with more training data, these errors would decrease.

4. Discussion

In this paper we introduced a method for sports activity recognition that generalizes well and produces high accuracy predictions with exposure to little training data. The problem we studied was one of jointly segmenting and

labeling sensor data into activity atoms, a broad problem that has application certainly beyond swimming training and also beyond sports monitoring in general. The key to our success is that we treat the problem in a joint fashion and that we use a discriminative training algorithm. The HMM system attempts to learn a model for a probability distribution of accelerometer data given an underlying segmentation, that is $\Pr(x|y, w)$. We refer to this as a generative model as we can sample or *generate* accelerometer sequences from the estimated distribution. In contrast, the discriminative SMM system merely learns rules for producing a segmentation y, given x, arguably a simpler task. Recently [31], research has shown the superiority of discriminative systems over generative models when the model describing the underlying distribution is unknown, as is the case in our problem. Further, SMMs have the capacity to use richer segment features, which HMMs do not and so are naturally more suited to problems of labeled segmentation, such as activity recognition.

Our approach, a semi-Markov model equipped with a grammar, derives its power from the capacity to consider varying length segments. This increased learning power comes at a potentially severe computational price and it is only made feasible through using a restrictive grammar. In activity recognition, a person can often be thought of as being in a certain context or environment where a certain set of actions are possible. A person can do different things if he is in a car rather than running. Furthermore, there are only some particular ways of transitioning between running and driving a car. This kind of complex grammar, which our daily lives adhere to, can be effectively represented using an automaton. The task of making mobile devices aware of the context of the user is a major current technological challenge that we believe that the framework presented in this article is perfectly suited to. The grammar, which is inherent in our swimming segmentation problem, also exists in many other kinds of sequential data analysis problems, such as activity analysis both in real life and on the internet, in security applications, for example, surveillance, in the analysis of network traces in a server center, or even in financial data analysis.

References

[1] R. Aylward, S. D. Lovell and J. A. Paradiso. Compact, Wireless, Wearable Sensor Network for Interactive Dance Ensembles. In *2006 International Workshop on Wearable and Implantable Body Sensor Networks*, 65–70, 2006.

[2] O. Aziz, B. P. L. Lo, G. Z. Yang, R. King, and A. Darzi. Pervasive body sensor network: An approach to monitoring the post-operative surgical patient. In *2006 International Workshop on Wearable and Implantable Body Sensor Networks*, pages 13–18, 2006.

[3] L. E. Baum and T. Peterie and G. Souled and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171, 1970.

[4] L. Bao and S. Intille. Activity recognition from user-annotated acceleration data. In A. Ferscha and F. Mattern, editors, *Pervasive Computing*, volume LNCS 3001, pages 1–7, 2004.

[5] A. P. Dempster and N. M. Laird and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Statistical Society Series B*, 39:1–38, 1977.

[6] M. Ermes, J. Parkka, J. Mantyjarvi, and I. Korhonen. Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions. *IEEE Transactions on Information Technology in Biomedicine*, 12(1):20–26, 2008.

[7] S. Furui, Speaker-independent isolated word recognition using dynamic features of speech spectrum *IEEE Transactions on Acoustics, Speech and Signal Processing*, 34(1): 52–59, 1986.

[8] 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.

[9] M. Hansen, M. K. Haugland, and T. Sinkjaer Evaluating Robustness of Gait Event Detection based on Machine learning and Natural Sensors In *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, volume 12(1), pages 81–88, 2004.

[10] B. Huang, M. Chen, X. Shi, and Y. Xu Gait EventDetection with Intelligent Shoes In *Proceedings of the 2007 International Conference on Information Acquisition*, 2007.

[11] D. James, N. Davey, and T. Rice. An accelerometer based sensor platform for insitu elite athlete performance analysis. In *Sensors, Proc. of IEEE*, volume 3, pages 1373–1376, 2005.

[12] J. Janssen and N. Limnios. *Semi-Markov Models and Applications.* Kluwer Academic, 1999.

[13] R. T. Lauer, B. T. Smith, and R. R. Betz Application of a Neuro-Fuzzy Network for Gait Event Detection Using Electromyography in the Child With Cerebral Palsy In *IEEE Transactions on Biomedical Engineering*, volume 52(9), pages 1532–1540, 2005.

[14] F. Lefevre, C. Montacie and M.-J. Caraty, On the Influence of the Delta Coefficients in a HMM-based Speech Recognition System, In *Proc. ICSLP98"* 1998

[15] O. Masoud and N. Papanikolopoulos. Recognizing human activities. In *AVSS '03: Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance*, page 157, 2003.

[16] A. Nedich and D. P. Bertsekas. Convergence rate of incremental subgradient algorithms. In S. Uryasev and P. M. Pardalos, ed., *Stochastic Optimization: Algorithms and Applications*, pages 263–304. Kluwer, 2000.

[17] S. Patel, D. M. Sherrill, R. Hughes, T. Hester, T. Lie-Nemeth, P. Bonato, D. Standaert, and N. Huggins. Analysis of the severity of dyskinesia in patients with parkinson's disease via wearable sensors. In *2006 International Workshop on Wearable and Implantable Body Sensor Networks*, 123–126, 2006.

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

[19] G. Raetsch and S. Sonnenburg. Large scale hidden semi-markov svms. In *NIPS 19*, 2006.

[20] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman. Activity recognition from accelerometer data. *American Association for Artificial Intelligence*, 2005.

[21] H. E. Robbins and S. Monro. A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407, 1951.

[22] S. Sarawagi and W. Cohen. Semi-markov conditional random fields for information extraction. In S. Thrun, L. Saul, and B. Schölkopf, editors, *NIPS 16*, 2004.

[23] Q. Shi, Y. Altun, A. Smola, and S. V. N. Vishwanathan. Semi-markov models for sequence segmentation. In *EMNLP*, 2007.

[24] S. Young and G. Evermann and D. Kershaw and G. Moore and J. Odell and D. Ollason and D. Povey and V. Valtchev and P. Woodland. The HTK Boook (for HTK version 3.2). Univ. Cambridge, Cambridge, U.K., 2002.

[25] B. N. Schilit, N. Adams, and R. Want Context-Aware Computing Applications. In *IEEE Workshop on Mobile Computing Systems and Applications*, 1994.

[26] M. M. Skelly and H. H. Chizeck Real-Time Gait Event Detection for Paraplegic FES Walking In *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, volume 9(1), pages 59–68, 2001.

[27] B. T. Smith, D. J. Coiro, R. Finson, R. R. Betz, and J. McCarthy Evaluation of Force-Sensing Resistors for Gait Event Detection Ro Trigger Electrical Stimulation to Improve Walking in the Child With Cerebral Palsy. In *IEEE Transactions on Neuural Systems and Rehabilitation Engineering*, volume 10(1), pages 22–29 2002.

[28] B. Taskar, C. Guestrin, and D. Koller. Max-margin Markov networks. In S. Thrun, L. Saul, and B. Schölkopf, editors, *NIPS 16*, pages 25–32, Cambridge, MA, 2004. MIT Press.

[29] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *J. Mach. Learn. Res.*, 6:1453–1484, 2005.

[30] R. Williamson and B. J. Andrews Gait Event Detection for FES Using Accelerometers and Supervised Machine Learning In *IEEE Transactions on Rehabilitation Engineering*, volume 8(3), pages 312–319, 2000.

[31] P. Liang and M. Jordan. An Asymptotic Analyssi of Generative, Discriminative and Psuedolikelihood Estimators. In *ICML '08: The 25th International Conference on Machine Learning*, Helsinki, Finland, 2008

Figure 1

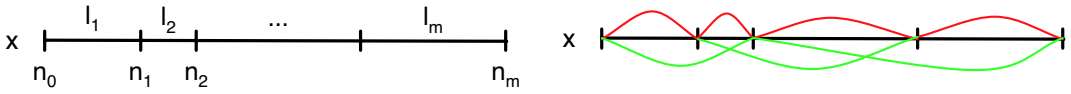


Figure 2

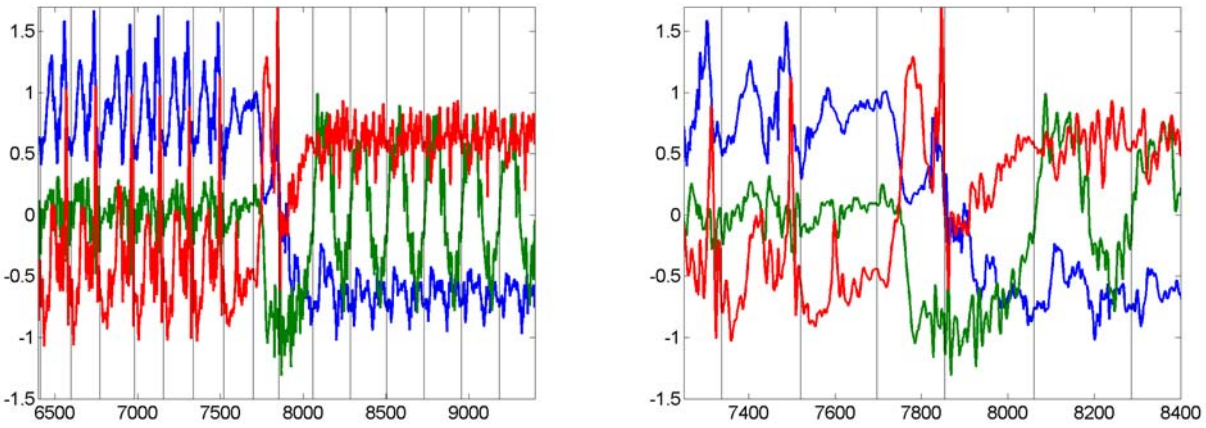
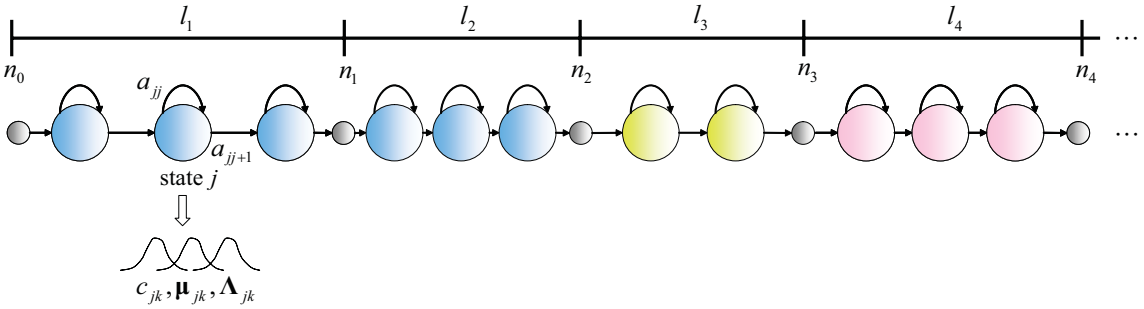

1.5
1
0.5
0
-0.5
-1
-1.5
6500
7000
7500
8000
8500
9000
1.5
1
0.5
0
-0.5
-1
-1.5
7400
7600
7800
8000
8200
8400

Figure 3


l_1
l_2
l_3
l_4
n_0
n_1
n_2
n_3
n_4
a_{jj}
a_{jj+1}
state j
$c_{jk}, \boldsymbol{\mu}_{jk}, \boldsymbol{\Lambda}_{jk}$

***Response to Reviewers**

We have added a new paragraph in the beginning of section 2 before 2.1 that describes our general approach to activity recognition.

1. You are right that we have accidentally neglected to say what grammar we actually use for the swimming system. We have now included a description in the beginning of chapter 3 just before 3.1 starts. Its a very simple, yet effective, grammar that says that a stroke can only transition to the same stroke or a turn while a turn cannot transition to a turn. It enforces output that consists of laps of the same kind of swimming and reduces the possible labeled segmentations significantly.
When it comes to labels vs states we intended to only use the term label when we talk about SMM while for the dual layer HMM approach the states are a sort of sublabel as explained in 2.3. We have found that the word state was accidentally used at some places for the SMM as well as a synonym for label. We have now changed those words to "label". This happened in two places right above equation 10, one time just below eq 10 and one time at the end of 2.5, six lines from the end.

2. This question is from the first round of revision when the extra log was removed.

3. We have now included pseudo-code at the end of 2.4.

4. We have included comments including two references at the end of 2.3.1. The features are known to useful for speech and speaker recognition and they turned out to be suitable for activity recognition as well. These "delta features" introduce temporal information into the features.

5. This question was addressed the first time. In particular the name of the sensor device and its manufacturer was included. We have now added that the computer was a MacBook.

6. This question was also addressed the first time. The amount of training data was specified and we clarified that the computational efficiency was much better when a trained system is being used and that this makes it practical. The training time is not important since one train once and then repeatedly use the trained system.

We have now added furher information about the training data in section 3.1.

7. One of the figures illustrate how the data is seen from a semi-Markov perspective. This picture demonstrate the basic assumption that underlies the SMM approach. That kind of pictures has been used before to illustrate the semi-Markov assumption (e.g. in Shi, Alton, Smola. Vishwanathan [23])
Another shows an example of data and of a labeled segmentation to illustrate what the task is and we have one picture that illustrate the HMM approach with its labels and states.