Shallow Parsing of a Tennis Game from Audio Events

Qiang Huang
School of Computing Sciences
University of East Anglia
Norwich, UK
h.qiang@uea.ac.uk

Stephen Cox
School of Computing Sciences
University of East Anglia
Norwich, UK
s.j.cox@uea.ac.uk

Abstract—This paper proposes a method to infer the syntactical units of a sports game (tennis) from a stream of game events. We assume that we are given a sequence of events within the game (examples of events are “serve”, “rally”, “score announcement” etc.), with their durations, and our goal is to segment them into “units” that are meaningful for the game, such as a “point”. Such a segmentation is essential for understanding the way that the events relate to each other, and hence for inferring automatically the structure of the game. We use a multi-gram based technique to segment the event stream into variable-length sequences by estimating the optimal (maximum-likelihood) segmentation using the Viterbi algorithm. We then make use of some extra contextual information, namely the time gap between two adjacent match events, which is in itself a reasonable indicator of segmentation. By integrating this feature into the multi-gram segmentation, we considerably enhance segmentation performance. The results show that our approach is an effective way to parse a tennis game from a stream of events with minimal human intervention.

Keywords—Shallow parsing; variable-length unit; segmentation; game learning;

I. INTRODUCTION

There have recently been several studies on analysis of sports video which have concentrated on the classification of low-level events, using either purely visual information [1], purely audio information [2] or a combination of the two [3]. These works focused on signal-processing and pattern classification techniques that enable a particular object, activity, scene or event within a game to be classified. However, in order to understand a game, we must move beyond low-level recognition of events and activities themselves to understanding how the events relate to each other and how they are combined from a continuous event stream.

In this paper, we assume that we are presented with a characterization of a tennis game as a sequence of events, each event being one of a small set of recognised events (see Table 1 for a description of these). In this work, we have used only the audio track of a DVD of the game to obtain the events: we are currently integrating the knowledge obtained from an analysis of the audio track with knowledge obtained from the video signal. Previous work by us [2] and others [3] has shown that by using appropriate pattern-classification techniques, we can uncover such a sequence from audio information with quite good accuracy. The challenge is then to parse this event symbol stream and reasonably segment it into game units.

The game units we use for tennis are points, which consist of a different number of match events and form a natural unit for segmenting the event stream. A point is completed after one or more serves (a serve is the initial delivery of the ball across the net from one player) are followed by zero or more ball hits by both players (a succession of ball hits is termed a rally), until a stopping criterion is met (e.g. the ball is stopped by the net or lands beyond the extent of the court). A crucial point is that points have different numbers of match events in them. The shortest possible sequence of events in a point is “Serve-Applause-Score” (corresponding to an “ace” serve), whereas a longer point could be e.g. “Serve-Let-Serve-Rally-Applause-Score”, where the first serve was faulty and a rally was played. A rally is detected as a sequence of ball-hit events. After a point is completed, play stops and the score is announced. Points form the basis of a larger scoring unit, namely games, which form the basis for a larger-still unit, namely sets, but for present purposes, we confine our attention to points.

Previous work used data mining techniques for sequence extraction[4] and structure inference[5], [6]. [4] made use of the Apriori algorithm and the max-subpattern tree to mine partial periodic patterns in time series. [5], [6] describe a “storyline” model dependent on AND-OR graphs, which also builds a tree-like hierarchical structure. However, these methods depend heavily on the fixed structure of graphical models, most of which are developed in supervised way. In addition, the fixed structure of such models severely limits their application to a task in a different domain. In this paper, we use the multi-gram technique, which has been shown to be very effective in modelling both variable length lexical units in language [9] and variable length acoustic units in speech [7].

We extend the technique by adding some contextual information that is easily obtained from our data. The inter-event time is the period between two successive events, which should be useful to effectively parse the event stream. We test this hypothesis in the paper by combining the multi-gram technique with the inter-event timing in a probabilistic
framework.

II. THEORETICAL FRAMEWORK

In their 1995 paper [9], Deligne and Bimbot addressed the following problem: given a sequence of symbols, each symbol being drawn from a finite alphabet, find the best (according to some criterion) segmentation of the sequence into commonly-occurring sub-sequences of symbols, hence defining a (small) set of units. In this paper, the symbols were letters and the units were words: the words are not known in advance but are discovered as a consequence of the segmentation. The segmentation criterion used was maximum likelihood, so the segmentation produced is the most likely one in terms of the probability of occurrence of the words. In our work, we regard each event as a single “letter”, and hence by segmenting the symbol stream, we discover the underlying “words” (game units).

In this section, we firstly describe the principle of the multi-gran technique and how we applied it to our data, and then introduce the integration of time-gap information.

A. Acquisition of variable-length game units

Let \( E = e(1) \cdots e(t) \cdots e(T) \) denote a stream of \( T \) events, and \( D \) denote a possible segmentation of \( E \) into \( q \) sequences of events: \( s(1) \cdots s(q) \). The likelihood of the stream of events \( E \) associated to segmentation \( D \) is computed as:

\[
L(E, D) = \prod_{t=1}^{t=q} Pr(s(t)) \tag{1}
\]

Here, our aim is to find the most likely segmentation of \( E \)

\[
L^* (E) = \max_{D \in \{D\}} L(E, D) \tag{2}
\]

where \( \{D\} \) is the set of all possible segmentations of \( E \) into sequence of events. The multi-gran model is hence fully defined by a set of parameters \( \Theta \) consisting of the probability of each event sequence \( s_i \in V \), where \( V = \{ s_1, \cdots, s_m \} \), a dictionary containing all the sequences of events.

To compute the set of parameters \( \Theta \) from a training corpus \( E \), an iterative Maximum Likelihood (ML) will be used through an Expectation Maximization (EM) algorithm:

\[
L(D|E, \Theta^{(k)}) = \frac{\sum_{D \in \{D\}} c(s_i, D) \times L(D|E, \Theta^{(k)})}{\sum_{D \in \{D\}} c(D) \times L(D|E, \Theta^{(k)})} \tag{3}
\]

where \( c(s_i, D) \) is the number of occurrences of sequence \( s_i \) in segmentation \( D \); \( c(D) \) is the total number of sequences in \( D \); and \( L(D|W, \Theta^{(k)}) \) is the conditional likelihood of the segmentation \( D \) given \( E \) at iteration \( k \) by segmentation \( k \rightarrow L(D|E, \Theta^{(k)}) = L(E, D, \Theta^{(k)}) \).

The estimation of the model parameters can be computed through an iterative forward-backward procedure [7]. It relies on the estimation of a forward variable \( \alpha \) and a backward variable \( \beta \), which are defined as the likelihood of the partial observed stream of events \( E_{(1)}^{(t)} \) and \( E_{(t+1)}^{(T)} \), respectively.

1) Recursion formula for the variable \( \alpha \) for \( 1 \leq t \leq T \):

\[
\alpha(t) = \sum_{l=1}^{n} \alpha(t - l) p([e(t - l + 1) \cdots e(t)])
\]

where \( \alpha(0) = 1 \) and \( \alpha(t) = 0 \) for \( t < 0 \). (\( n \) is the maximal length of a segment.)

2) Recursion formula for the variable \( \beta \) for \( 1 \leq t \leq T \):

\[
\beta(t) = \sum_{l=1}^{n} p([e(t + l) \cdots e(t + l)]) \beta(t + l),
\]

with \( \beta(T) = 1 \) and \( \beta(t) = 0 \) for \( t > T \).

3) Parameter re-estimation for a sequence \( s_i \) of \( l \) events,

\[
\theta_i^{(k+1)} = \frac{\sum_{t=1}^{T} \sum_{l=1}^{n} \delta(t, l,i) \alpha(t) \beta(t)}{\sum_{t=1}^{T} \alpha(t) \beta(t)}
\]

where

\[
\delta(t, l, i) = \begin{cases} 1 & \text{if } [e(t-l+1) \cdots e(t)] = s_i \\ 0 & \text{otherwise} \end{cases}
\]

Go back to Step 1 for \( N \) iterations.

Decoding

1. Initialization

\[
\delta_i(t) = p([e(1) \cdots e(1 + i - 1)])
\]

\[
\psi_1(i) = \begin{cases} 1 & \text{if } l(t, i) = 0 \\ 0 & \text{otherwise} \end{cases}
\]

2. Recursive

\[
\delta_i(t) = \max_{l \leq t \leq T} \{ \delta_i(t - l) \beta(t) \}
\]

\[
\psi(t) = \arg \max_{l \leq T, 1 \leq j \leq n} \{ \delta_i(t - l) \}
\]

3. Traceback (refer to [10])

Figure 1. Estimation of parameters with forward-backward algorithm

B. Incorporating the time gap between two events

As already mentioned in Section 1, it seems likely that the inter-event timing would be useful for parsing of match event stream. Figure 2 shows the distribution of inter-event timing that occur within a game-unit (left curve), and for the situation where the first event is the final event of a game unit and the second event is the first event in the next game unit (right curve). Note that the \( x \)-axis scale of \( \log(t+1) \) has been chosen for the convenience of plotting small values of the time-gap, \( t \).

The figure indicates that although there is some overlap of the tails of “within” and “across” units in the centre of the figure, the two distributions are quite well-separated and hence offer good potential for discrimination.
C. Combination of multigram model and time gap

To take advantage of the context information, we combine it with the multi-gram model into the Viterbi procedure by changing Step 4.2 in Table 1 to

\[
\delta_i(j) = \max_{1 \leq i < n} [\delta_{i-1}(i)]p(\epsilon(1) \cdot \cdot \cdot \epsilon(1 + j - 1))F_i(t')^\lambda \quad (6)
\]

where \(F_i(t')\) is the cumulative distribution of the time gap \(t'\) between two adjacent match events at time \(t_i\) and \(t_{i+1}\), and \(\lambda\) is weighting coefficient. \(\lambda\) needs to be determined empirically, and is set to 15 in this paper. In practice, performance varies little with \(\lambda\).

III. DATA

Our data is taken from the soundtrack of DVDs of the Wimbledon 2008 tournament. These events are interspersed with commentary on the match from a team of commentators, which we ignore in this work. We manually labelled about 4.5 hours of audio data with five distinct audio classes corresponding to eight types of match events (the event “commentary” was also labelled, but is not used here).

<table>
<thead>
<tr>
<th>Audio Class</th>
<th>Event</th>
<th>Description</th>
<th>Match Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>UMP</td>
<td></td>
<td>speech from chair umpire</td>
<td>challenge</td>
</tr>
<tr>
<td></td>
<td></td>
<td>speech from chair umpire</td>
<td></td>
</tr>
<tr>
<td>LJ</td>
<td></td>
<td>cry from line judges</td>
<td>out</td>
</tr>
<tr>
<td>BS</td>
<td></td>
<td>sound of racquet hitting ball</td>
<td>serve</td>
</tr>
<tr>
<td></td>
<td></td>
<td>sound of racquet hitting ball</td>
<td>rally</td>
</tr>
<tr>
<td>CN</td>
<td></td>
<td>crowd noise</td>
<td>applause</td>
</tr>
<tr>
<td></td>
<td></td>
<td>crowd noise</td>
<td>roar</td>
</tr>
<tr>
<td>BP</td>
<td></td>
<td>beep</td>
<td>net</td>
</tr>
</tbody>
</table>

The corpus was divided into two parts: 3.5 hours of data was used for training and one hour for test. Table II shows the number of match events and the number of game-units in the training- and test-sets. When processing the match state sequence, we use a single event symbol to represent a sequence of events that have the same labelling; for instance, several sequential “ball-hit” events would be merged as a single “rally” event.

To test the effectiveness of our work, we measure the accuracy of the segmentation with the estimated F-score:

\[
F\text{-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}
\]

where \(\text{Recall} = (# \text{ correctly identified sequences}) / (# \text{ 'ground-truth' sequences})\) and \(\text{Precision} = (# \text{ correctly identified sequences}) / (# \text{ sequences produced by the segmentation algorithm})\).

IV. EXPERIMENTAL RESULTS AND ANALYSIS

In the experiments, we test:

1) using only the multi-gram approach;

2) combining the time-gap with the multi-gram approach.

When implementing the multi-gram model, we test the segmentation performance using different maximum lengths of a game-unit, from four to seven events (unlike the work described in [1], where a fixed length is used). This is important, because we do no know a priori how many events constitute a game-unit. We also observe how the performance changes as the algorithm is iterated.

Figure 3 and 4 show the results using multi-gram model on the training and test set, respectively. These figures show the value of iterating the multi-gram algorithm. Best performance for the training-set (62.17%) is obtained using a maximum game-unit length of five events, whilst the best performance on the test-set (71.79%) is obtained using a maximum length of six, although this is not statistically significantly different from the best performance obtained using five events. Once again, performance on the test-set is better than on the training-set, which in this case, we attribute to the fact that the match on which the training was based was more complex, containing challenges from the players and a higher proportion of “let” serves.

Table II

<table>
<thead>
<tr>
<th></th>
<th># Match Event</th>
<th># Segment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training set</td>
<td>2571</td>
<td>325</td>
</tr>
<tr>
<td>Test set</td>
<td>593</td>
<td>67</td>
</tr>
</tbody>
</table>
no requirement to iterate the algorithm when the time-gap information is added: the improvement is immediate. For the training-set, including the time-gap data actually lowers the performance obtained from using the multi-gram technique, at least in the early iterations. Analysis of the segmentations produced as the algorithm was iterated indicated that when different segmentation hypotheses were close in probability, any “noise” in the time-gap data could actually reinforce an incorrect segmentation. As iteration produced more sharply-peaked correct segmentations, this effect was lessened.

V. DISCUSSION

In this paper, we have presented a technique that automatically infers the essential building-blocks of the syntax of a sports game from the observed continuous stream of events in the game. We have achieved this by posing the problem as an optimal segmentation problem and using a powerful technique (multi-gram segmentation), combined with using evidence about the timing of events. The performance obtained is very encouraging, and gives us a basis for going on to use the “game-units” as a basis for understanding the game in greater depth. Our future work will focus on how robust our techniques are to discovering the structure of other games.

ACKNOWLEDGMENT: This work was supported under a UK Engineering and Physical Sciences Research Council Grant number EP/F069626/1.

REFERENCES


