Hierarchical Temporal Association Mining for Video Event Detection in Video Databases

Min Chen[1], Shu-Ching Chen[1], Mei-Ling Shyu[2]
[1]Distributed Multimedia Information System Laboratory
School of Computing & Information Sciences
Florida International University, Miami, FL, 33199, USA
Tel: (305)-348-3480, Email: {mchen005, chens}@cs.fiu.edu
[2]Department of Electrical and Computer Engineering
University of Miami, Coral Gables, FL, 33124, USA
Tel: (305)-284-5566, Email: shyu@miami.edu

Abstract

With the proliferation of multimedia data and ever-growing requests for multimedia applications, new challenges are emerged for efficient and effective managing and accessing large audio-visual collections. In this paper, we present a novel framework for video event detection, which plays an essential role in high-level video indexing and retrieval. Especially, since temporal information in a video sequence is critical in conveying video content, a hierarchical temporal association mining approach is developed to systematically capture the characteristic temporal patterns with respect to the events of interest. In this process, the unique challenges caused by the loose video structure and skewed data distribution issues are effectively tackled. In addition, an adaptive mechanism is proposed to determine the essential thresholds which are generally defined manually in the traditional association rule mining (ARM) approach. This framework thus largely relaxes the dependence on the domain knowledge and contributes to the ultimate goal of automatic video content analysis.

1. Introduction

The fast proliferation of video data archives has increased the need for automatic video content analysis. Such automatic analysis greatly eases the authoring of content structure and increases data accessibility, which is critical for video database management and multimedia applications including broadcast video, video-on-demand, web search, etc.

As an essential step to facilitate automatic video content analysis, video event detection has attracted great attentions from the research society. In the literature, most of the existing frameworks in video event detection are conducted in a two-step procedure [6]. The first step is video content processing, where the video clip is segmented into certain analysis units and their representative features are extracted. In order to effectively characterize the video documents, there are quite a number of research efforts devoted in this step to extract feature descriptors at low-level, mid-level and object-level [4], where a shot is generally adopted as the analysis unit. The second step is called the decision-making process that extracts the semantic index from the feature descriptors. Decision-making can be roughly grouped into knowledge-based approaches and statistical approaches [7]. Knowledge-based approaches typically combine the output of different media descriptors into rule-based classifiers, whereas the statistical approaches include the use of C4.5 decision trees [3], support vector machines [1], dynamic Bayesian Network [10], etc. to improve the framework robustness. In this paper, we will focus our research efforts into two critical issues in video event detection which have yet not been well studied.

- First, normally a single analysis unit (e.g., a shot) which is separated from its context has less capability of conveying semantics [11]. Temporal information in a video sequence plays an important role in conveying video content. Consequently, an issue arises as how to properly localize and model context which contains essential clues for identifying events. One of the major challenges is that for videos, especially those with loose content structure (e.g., sports videos), such characteristic context might occur at uneven inter-arrival times and display at different sequential orders. Some studies tried to adopt temporal evolution of certain feature descriptors for event detection. For instance, temporal evolutions of so-called visual descriptors such as

"Lack of motion," "Fast pan," and "Fast zoom" were employed for soccer goal detection in [6], with the assumption that any interesting event affects two consecutive shots. In [5], the temporal relationships of the sub-events were studied to build event detection grammar. However, such setups are largely based on domain knowledge or human observations, which highly hinder the generalization and extensibility of the framework.

- Second, the events of interest are often highly infrequent. Therefore, the classification techniques must deal with the class-imbalance (or called skewed data distribution) problem. The difficulties in learning to recognize rare events include: few examples to support the target class, the majority (i.e., nonevent) class dominating the learning process, etc.

To address these issues, we propose a hierarchical temporal association mining approach which integrates association rule mining (ARM) and sequential pattern discovery to systematically determine the temporal patterns for target events in this paper. In addition, an adaptive mechanism is adopted to update the minimum *support* and *confidence* threshold values by exploring the characteristics of the data patterns. Such an approach largely relaxes the dependence on domain knowledge or human efforts. Furthermore, the challenges posed by skewed data distribution are effectively tackled by exploring frequent patterns in the target class first and then validating them over the entire database. The mined temporal pattern is thereafter applied to further alleviate the class imbalance issue. We use soccer videos as our test bed due to its popularity and loose structure.

The remainder of the paper is organized as follows. Section 2 introduces the background of this study as well as the related work. We detail the framework in Section 3. Section 4 shows the experimental results, and Section 5 concludes this paper.

2. Background and Related Work

Association rules are an important type of knowledge representation revealing implicit relationships among the items present in a large number of transactions. Given $I = \{i_1, i_2, \ldots, i_n\}$ as the item space, a transaction is a set of items which is a subset of I. In the original market basket scenario, the items of a transaction represent items that were purchased concurrently by a user. An association rule is an implication of the form [$X \rightarrow Y$, *support*, *confidence*], where X and Y are sets of items (or itemsets) called antecedent and consequence of the rule with $X \subset I$, $Y \subset I$, and $X \cap Y = \emptyset$. The *support* of the rule is defined as the percentage of transactions that contain both X and Y among all transactions in the input data set; whereas the *confidence* shows the percentage of transactions that contain Y among transactions that contain X. The intended meaning of this rule is that the presence of X in a transaction implies the presence of Y in the same transaction with a certain probability. Therefore, traditional ARM aims to find frequent and strong association rules whose support and confidence values exceed the user-specified minimum *support* and minimum *confidence* thresholds.

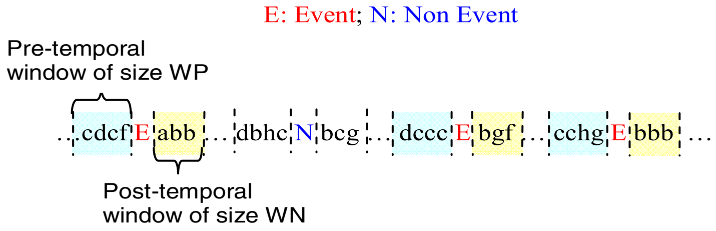
E: Event; N: Non Event
Pre-temporal window of size WP
…cdcf E abb … dbhc N bcg … dccc E bgf … cchg E bbb …
Post-temporal window of size WN

Figure 1. An example video sequence

Intuitively, the problem of finding temporal patterns can be converted as to find adjacent attributes (i.e., X) which have strong associations with (and thus characterize) the target event (i.e., Y), and thus ARM provides a possible solution. Here, assume the analysis is conducted at the shot-level, the adjacent shots are deemed as the transaction and the attributes (items) can be the feature descriptors (low-, mid- or object-level extracted from different channels) or event types in the transaction. However, as discussed below, the problem of temporal pattern discovery for video event detection has its own unique characteristics, which differs greatly from the traditional ARM.

Without loss of generalization, an event E is normally the result of previous actions (called pre-actions or AP) and might result in some effects (post-actions or AN). Given an example video sequence illustrated in Fig. 1, we define pre-transactions TP (such as $\{c, d, c, f\}$ and $\{d, c, c, c\}$) and post-transactions TN (such as $\{a, b, b\}$ and $\{b, b, b\}$) as covered by the pre-temporal windows and post-temporal windows, respectively. The characters 'a', 'b', etc. denote the attributes of the adjacent shots. Note that if the feature descriptors are used as the attributes, certain discretization process should be conducted to create a set of discrete values to be used by ARM. A temporal context for target event E is thus composed of its corresponding pre-transaction and post-transaction, such as $<\{c, d, c, f\}\{a, b, b\}>$ and $<\{c, c, h, g\}\{b, b, b\}>$. The purpose of temporal association mining is thus to derive rules $<AP, AN>\rightarrow E$ that are frequent and strong, where $AP \subset TP$ and $AN \subset TN$. Mainly, temporal pattern mining differs from the traditional ARM in two aspects.

- First, an itemset in traditional ARM contains only distinct items without considering the quantity of each item in the itemset. However, in event detection, it is indispensable that an event is characterized by not only the attribute type but also its occurrence frequency. For instance, in surveillance video, a car passes by a bank once is considered normal, whereas special attention might be required if the same car appears frequently within a temporal window around the building. In soccer video, several close views appear in a temporal window might signal an interesting event, whereas one single close view is generally not a clear indicator. Therefore, a multiset concept is adopted, which as defined in mathematics, is a variation of a set that can contain the same item more than once. To our best knowledge, such an issue has not been addressed in the existing video event detection approaches. A slightly similar work was presented in [11], where ARM is applied to the temporal domain to facilitate event detection. However, it uses the traditional itemset concept. In addition, it searches the whole video to identify the frequent itemsets. Under the situation of rare event detection where the event class is largely under-represented, useful patterns is most likely overshadowed by the irrelevant itemsets.
- Second, in traditional ARM, the order of the items appeared in a transaction is considered as irrelevant. Therefore, transaction $\{a, b\}$ is treated the same as $\{b, a\}$. In fact, this is an essential feature we adopted to address the issue of loose video structure. Specifically, the characteristic context information can occur at uneven inter-arrival times and display at different sequential orders as mentioned earlier. Therefore, given a reasonably small temporal window, it is preferable to ignore the appearance order of the attributes inside a pre-transaction or post-transaction. However, considering the rule $<AP, AN>\rightarrow E$, AP always occurs ahead of its corresponding AN, and the order between them is important in characterizing a target event. Therefore, in this stage, we will adopt the idea of sequential pattern discovery [8], where a sequence is defined as an ordered list of elements. In our case, each element is a multiset. In other words, the sequence $<\{a, b\}\{c\}>$ is considered to be different from $<\{c\}\{a, b\}>$. Note that in this paper, we use braces for multisets and angle brackets for sequences.

After conveying the general concept via the simple example, we will first give an overview of the proposed framework. The technical details of the proposed *hierarchical temporal association mining* component will be presented in the next section.

The proposed framework is divided into three major components based on their functionalities, namely *visual and audio feature extraction*, *hierarchical temporal association mining*, and *multimodal data mining*, as illustrated in Fig. 2. In the *visual and audio feature extraction* component, an unsupervised video shot boundary detection subcomponent [2] is used to temporally segment the raw soccer video sequences into a set of consecutive video shots. The detected shot boundaries are thus passed to the feature extraction subcomponent, where the shot-level multimodal features (visual and audio features) are extracted. Here, visual features are captured with the assistance of color analysis and object segmentation techniques, whereas audio features are exploited in both time-domain and frequency-domain. A complete list of multimodal features and their detailed feature descriptions can be found in [3]. The *hierarchical temporal association mining* component is then performed to explore the temporal patterns significant for characterizing the events and results in a set of temporal rules, which are effectively employed to capture the candidate video events and to alleviate the class-imbalance issue. Note that instead of predefining the temporal patterns subjectively as in [3], this proposed approach searches for optimal temporal patterns automatically and robustly. Finally, the events of interest are detected automatically in the multimodal data mining component.

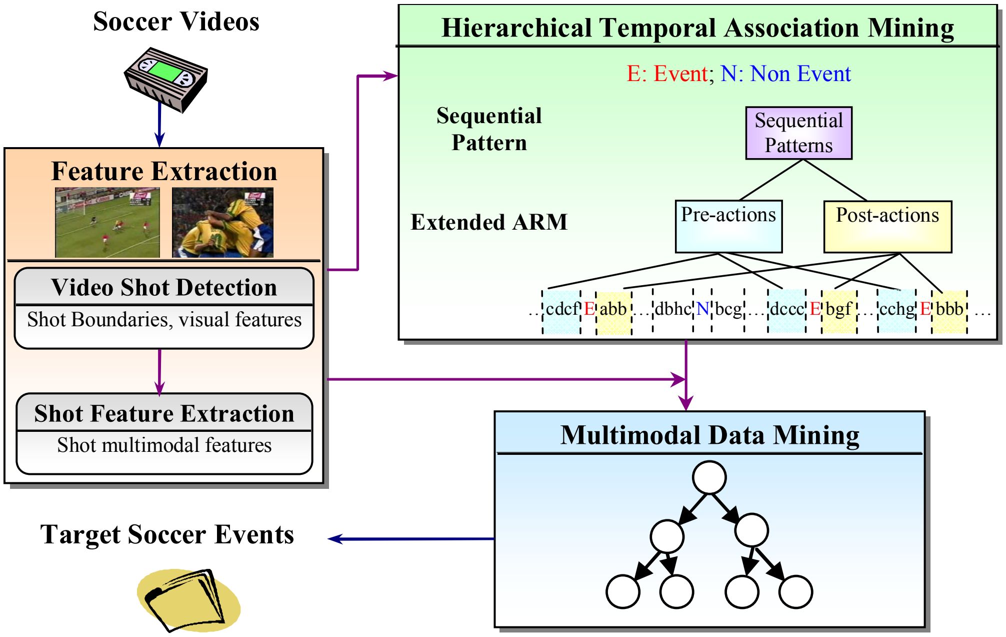

Figure 2. Framework architecture

3. Hierarchical Temporal Association Mining

Since we target to capture temporal patterns characterizing the contextual conditions around each target event, a hierarchical temporal association mining mechanism is proposed. As discussed earlier, due to the loose structure of videos, the attributes within the temporal windows (pre-temporal or post-temporal) have no orders. Meanwhile, the appearance frequency of the attributes is important in indicating the events. Hence, the proposed extended ARM algorithm is applied to find pre-actions *AP* and post-actions *AN* (called "Extended ARM" in Fig. 2), and then sequential pattern discovery is utilized where *AP* and *AN* are considered as the elements in a sequence (called "Sequential Patterns" in Fig. 2). Thereafter, the temporal rules are derived from the frequent and strong patterns. The approach is first presented with the predefined minimum *support* and *confidence* thresholds, and an adaptive updating mechanism is introduced to define them automatically.

Let $D_v = \{V_i\}$ be the training video database and *NF* be the number of attributes in the database, where V_i ($i = 1, \ldots, N_v$) is a video clip and N_v is the cardinality of D_v, we have the following definitions.

Definition 1. A video sequence V_i is an ordered collection of units $V_i = <V_{i1}, V_{i2}, \ldots, V_{in_i}>$, where each unit V_{ij} ($j = 1, \ldots, n_i$) is a 3-tuple $V_{ij} = (F_{ij}, s_{ij}, C_{ij})$. Here, n_i is the number of units in V_i, $F_{ij} = \{F_{ijk}\}$ indicates the set of unit attributes ($k = 1, \ldots, NF$), s_{ij} denotes its associated unit number, and $C_{ij} = \{yes, no\}$ is the class label showing the eventness of the unit.

In our study, the unit is defined at the shot level and the unit attribute, as mentioned earlier, can be the feature descriptors or event types of the shot. As usual, the task is to find all frequent and strong patterns from the transactions given the target event *E*. Therefore, the pre-transactions (*TP*) and post-transactions (*TN*) need to be constructed.

Definition 2. Given a unit V_{ij} ($j = WP+1, \ldots, n_i-WN$), the pre-temporal window size *WP* and post-temporal window size *WN*, its associated TP_{ij} and TN_{ij} are defined as $TP_{ij} = \{F_{ip}\}$ ($p = j-WP, \ldots, j-1$) and $TN_{ij} = \{F_{iq}\}$ ($q = j+1, \ldots, j+WN$).

3.1. Frequent patterns

We proceed by first finding all frequent patterns. Different from traditional ARM, to alleviate the problem of class imbalance problem, the frequent patterns are searched for the minority class only. In other words, in counting the frequent patterns and calculating the support values, only those $TP_E = \{TP_{ij}\}$ and $TN_E = \{TN_{ij}\}$ will be checked where C_{ij} = '*yes*'. As shown in Fig. 1, the multisets $\{d, b, h, c\}$ and $\{b, c, g\}$ around the nonevent N will not be checked in this step. Then the discrimination power of the patterns is validated against the nonevent class (in Section 3.2).

In order to mine the frequent pre-actions and post-actions, the *itemMultiset* (the counterpart of *itemset* in traditional ARM) is defined.

Definition 3. An *itemMultiset* T is a combination of unit attributes. T matches the characterization of an event in window WP or WN if T is the subset of TP_{ij} or TN_{ij} where C_{ij} = '*yes*'.

For example, if a post-temporal window with size WN for an event E (see Fig. 1) contains unit attributes $\{a, b, b\}$, then we call $T = \{b, b\}$ a match of the characterization of event E, whereas $T = \{a, a\}$ is not. Consequently, we revise the traditional *support* and *confidence* thresholds as follows.

Definition 4. An *itemMultiset* T has *support* s in D_v if s% of all $TP_E = \{TP_{ij}\}$ (or $TN_E = \{TN_{ij}\}$) for target event E are matched by T. T is frequent if s exceeds the predefined *min_sup*.

Mathematically, *support* is defined as

$$Support = Count(T, TP_E)/|TP_E| \quad (1)$$

$$\text{or } Support = Count(T, TN_E)/|TN_E| \quad (2)$$

From the equations, we can see that our definition of support is not simply an extension of the one used in traditional ARM. It is restricted to $TP_E = \{TP_{ij}\}$ or $TN_E = \{TN_{ij}\}$ which are associated with the target events (i.e., C_{ij} = '*yes*'). An *itemMultiset* which appears in D_v periodically might not be considered as frequent if it fails to be covered by these TP_E or TN_E. The pseudo code for finding frequent *itemMultisets* is listed in Table 1.

The general idea is to maintain in memory, for each target event, all the units within its associated TP_{ij} and TN_{ij}, which are then stored in B_p and B_n (steps 1 to 19), and Extended *Apriori* algorithm (extended *Apriori* like algorithm) is applied to find the frequent pre-actions and post-actions from B_p and B_n (steps 20 to 21).

Table 1. Logic to find all frequent actions

```
Input: video database $D_v$, pre-temporal window size $WP$, post-temporal window size $WN$, minimum support $min\_sup$, target-event type $E$
Output: frequent actions $AP$, $AN$
FrequentActions($D_v$, $WP$, $WN$, $min\_sup$, $E$)
1)  $B_p = \emptyset$; $T = \emptyset$; $B_n = \emptyset$
2)  for each video sequence $V_i \in D_v$
3)    for each unit $V_{ij} = (F_{ij}, s_{ij}, C_{ij}) \in V_i$
4)      for each unit $V_{ik} = (F_{ik}, s_{ik}, C_{ik}) \in T$
5)        if $(s_{ij} - s_{ik}) > WP$
6)          Remove $V_{ik}$ from $T$
7)        endif
8)      endfor
9)      if $V_{ij}$ is a target event  // i.e., $C_{ij}$ = 'yes'
10)       $B_p = B_p \cup \{F_{ik} \mid (F_{ik}, \cdot) \in T\}$
11)       $PS = s_{ij+1}$
12)       while $(PS - s_{ij}) < WN$
13)         $B_n = B_n \cup \{F_{ik} \mid s_{ik} = PS\}$
14)         $PS$ is set to its next shot until it is the end of $V_i$
15)       Endwhile
16)     endif
17)     $T = T \cup V_{ij}$
18)   Endfor
19) endfor
20) Use extended Apriori over $B_p$ to find $AP$ with $min\_sup$
21) Use extended Apriori over $B_n$ to find $AN$ with $min\_sup$
```

The procedure of the extended *Apriori* algorithm is shown in Table 2, which will be explained by an example. Since in the transactions (*TP* or *TN*) and *itemMultisets* we allow the existence of duplicated elements, we need to consider each unit attribute as a distinct element even though some attributes might have the same values, except for the construction of 1-*itemMultisets*. The frequent pre-patterns and post-patterns, obtained using the proposed extended *Apriori* algorithm upon the example video sequence shown in Fig. 1, are listed in Tables 3 and 4, respectively.

Here, we assume the minimum *support* count is set to 2 and the frequent actions are highlighted in yellow. Since we consider the ordering of the units and the inter-arrival times between the units and target events within each time window to be irrelevant in finding the frequent pre- and post-patterns, for the sake of simplicity, all the units inside the transactions and *itemMultisets* are sorted in the algorithm. Note that the computational cost for such procedures is minimal because the transactions are constructed only for minority class and the number of elements in such transactions is small. Without loss of generality, the window size is reasonably small since only the temporally adjacent shots have strong association with the target events.

Table 2. The procedure of extended A-priori algorithm

1) Construct 1-*itemMultisets*. Count their *supports* and obtain the set of all frequent 1-*itemMultisets* as in traditional *Apriori* algorithm

2) A pair of frequent *k*-*itemMultisets* are merged to produce a candidate (*k*+1)-*itemMultisets*. The merges are conducted in two steps:

2.1) A pair of frequent *k*-*itemMultisets* are merged if their first (*k*-1) items are identical, and

2.2) A frequent *k*-*itemMultiset* can be merged with itself only if all the elements in the *multiset* are with the same value.

3) The *supports* are counted and the frequent *itemMultisets* are obtained as the traditional *Apriori* algorithm. Go to step 2.

4) The algorithm terminates when no further merge can be conducted.

Table 3. Frequent pre-actions

1	#	frequent	2	#	frequent	3	#	frequent
{*c*}	3	Yes	{*c,c*}	3	Yes	{*c,c,c*}	1	No
{*d*}	2	Yes	{*c,d*}	2	Yes	{*c,c,d*}	2	Yes
{*f*}	1	No	{*d,d*}	0	No			
{*g*}	1	No						
{*h*}	1	No						

Table 4. Frequent post-actions

1	#	frequent	2	#	frequent	3	#	frequent
{*a*}	1	No	{*b,b*}	2	Yes	{*b,b,b*}	1	No
{*b*}	3	Yes						
{*g*}	1	No						
{*f*}	1	No						

As mentioned earlier, the ordering between pre-actions *AP* and post-actions *AN* needs to be observed, and so the idea of sequential pattern discovery is adopted (omitting the detailed algorithm here). However, it is worth noting that instead of scanning *TP* and *TN* to explore the frequent sequential patterns, the *Apriori* like principle can be applied to simplify the process, which states that for a particular sequence to be frequent, its element(s) must be frequent as well. For instance, given the examples shown in Fig. 1 and frequent pre- and post-actions listed above, respectively, we know sequence $<\{a\}\{d\}>$ is not frequent since its pre-action element $<\{a\}>$ is not frequent. Therefore, the frequent sequential patterns can be constructed upon the frequent *AP* and *AN*. Note that it is legal to have null pre-action or post-action in a sequential pattern (e.g., $<\{\}\{b, b\}>$ or $<\{c, c, d\}\{\}>$).

After we create the 1-*itemMutlisets*, we can extract the corresponding sequential patterns. Then when we make another pass over the transactions to find frequent 2-*itemMultisets*, the support of the constructed sequential pattern can be counted as well. The procedure terminates until no more frequent *(k+1)*-*itemMultisets* can be identified.

3.2. Strong patterns

To validate that these patterns effectively characterize the event of interest, a restrict solution is to adopt the traditional association measure called *confidence*, where a similar idea presented in [9] can be adopted. The general idea is to count the number of times each of the patterns occurs outside the windows of the target events.

Definition 5. A sequential pattern P has *confident c* in D_v if c% of all transactions matched by T are associated with the target event. P is strong if c exceeds *min_conf*.

Intuitively, we take inputs of a set of transactions, which correspond to all $TP_N = \{TP_{ij}\}$ and $TN_N = \{TN_{ij}\}$ with C_{ij} = '*no*'. In fact, such lists can be obtained in algorithm 1 when we scan through the unit sequence and store them in $B_p^{'}$ and $B_n^{'}$, respectively. Let x_1 and x_2 be the counts when the pattern T is matched in B and $B^{'}$. Here $B=\{b_1, b_2, \ldots, b_n\}$ is constructed by linking $B_p=\{b_{p1}, b_{p2}, \ldots, b_{pn}\}$ and $B_n=\{b_{n1}, b_{n2}, \ldots, b_{nn}\}$, where

b_i=<b_{pi}, b_{ni}>. Similarly, $B^{'}$ can be constructed by $B^{'}_p$ and $B^{'}_n$. The *confidence* of P is defined as follows.

$$confidence(P, B, B^{'}) = x_1/(x_1+x_2).$$

This metric is thus applied to compare with *min_conf* and to validate whether the temporal patterns are strong.

3.3. Temporal rules

Once we obtain the frequent and strong temporal patterns, we will build temporal rules to facilitate the event detection. The principle is defined as follows.

Definition 6. Given two patterns, P_i and P_j, $P_i \succ P_j$ (also called P_i has a higher rank than P_j) if

1. The *confidence* of P_i is greater than that of P_j, or
2. Their *confidences* are the same, but the *support* of P_i is greater than that of P_j, or
3. Both the *confidences* and *supports* of P_i and P_j are the same, but P_i is more specific than P_j (i.e., P_j is a subsequence of P_j).

The rules are in the form of P_i→E (targeted event). Let R be the set of generated rules and D_v be the training database. The basic idea is to choose a set of high ranked rules in R to cover all the target events in D_v. Such temporal rules are applied in data pruning process to generate a candidate event set and to alleviate class imbalance problem in the data classification stage.

3.4. Adaptive metrics updating mechanism

The performance of the proposed approach is partially related to four parameters, namely *WP*, *WN*, *min_sup* and *min_conf*. Among them, *WP* and *WN* can be determined relatively straightforward as generally only the temporally adjacent shots have strong association with the target events. Therefore, they can be set to any reasonably small values such as 3 or 4. In addition, in our earlier work [1], an advanced approach was proposed to identify the significant temporal window with regard to the target event, which can be incorporated into this framework to define the window size. Therefore, in this section, an adaptive metrics updating mechanism is proposed to define *min_sup* and *min_conf* in an iterative manner.

The richness of the generated patterns is partially dependent on *min_sup*, which in most existing works is defined manually based on domain knowledge. However, given a training database, it is infeasible to expect the users to possess the knowledge of the complete characteristics of the training set. Therefore, the proposed approach addresses this issue by refining the *support* threshold $SupTH_{k+1}$ iteratively based on the statistical analysis of the frequent patterns obtained using threshold $SupTH_k$.

Given k^{th} threshold $SupTH_k$, let R_k be the number of attributes in the largest frequent *itemMultisets*, we have Sup_{kr}={*supports* of all *r-itemMultisets}*, where r=1, …, R_k. Equations (3) to (5) define *min_sup*.

$$diff(r) = \text{mean}(Sup_{kr})\text{-mean}(Sup_{kr+1}),\ r=1,\ldots,R_k\text{-}1 \quad (3)$$

$$r_k = \underset{r}{\arg\max}(diff(r)) \quad (4)$$

$$\text{if } diff(r_k) > \frac{1}{2}R_k,\ SupTH_{k+1} = \frac{mean(Sup_{kr_k}) + mean(Sup_{kr_{k+1}})}{2}$$
$$\text{else } min\_sup = SupTH_k \quad (5)$$

The idea is that the learned frequent patterns in the previous round can help reveal certain knowledge regarding the training data set and thus help refine the *support* threshold intelligently. Specifically, we study the biggest fluctuation between the *supports* of two adjacent *itemMultisets*. Since (r+1)-*itemMultisets* are rooted from *r-itemMultisets*, if the difference is greater than $\frac{1}{2}R_k$, the support threshold is adjusted to avoid the possible over-fitting issue and improve framework efficiency. Note that the initial support threshold $SupTH_0$ can be set to a reasonably small value.

For the confidence threshold, a similar criterion is adopted to examine the biggest difference between two adjacent sequential patterns with the condition that the generated rules in R should be able to cover all target events in D_v. In other words, if the newly defined confidence threshold $ConTH_{k+1}$ causes the missing of target events in D_v, $ConTH_k$ is chosen as *min_conf*.

4. Experimental results

To assess the performance of our algorithm, we use soccer videos as our test bed and goal shots as the target event. Totally 27 soccer videos with total duration of more than 9 hours were collected from a variety of sources with various production styles. In the data set, the number of goal shots and nongoal shots is 39 and 4,624, respectively, which shows a largely skewed data distribution.

In the experiment, the training data set (2/3rds of the data) was used to train the model which was then tested by the remaining 1/3rd data (called testing data set). A so-called 5-fold cross-validation scheme was adopted, where the whole data set was divided randomly five times to obtain five different groups of training and testing data sets. Therefore, five models were constructed and each was tested by its corresponding testing data.

In feature extraction process, 15 low-level features including 5 visual features (pixel-change, histo-change, grass_ratio, background_mean, background_var) and 10 audio features (1 volume feature, 5 energy features and 4 spectrum flux features) are extracted at the shot-level. Meanwhile, 2 clip-level volume features are also captured to explore information in a finer granularity. Here, an audio clip is defined as an audio track with the duration of one second. In addition, two middle-level features (camera view and excitement label) are derived from low-level features and used to derive the temporal rules. This data pruning process filters out many inconsistent and irrelevant shots and produces a candidate event set where the goal shots accounted for about 6% of the remaining data set.

The resulting candidate pool was then passed to the decision tree based multimodal data mining process for further classification. We chose C4.5 not only because it is one of the most commonly used algorithms in the machine learning and data mining communities but also because it has become a de facto standard against which every new algorithm is judged.

Table 5. Performance of goal event detection

	# of goal	Iden	Missed	Misiden	Recall (%)	Precision (%)
Test 1	11	10	1	2	90.9	83.3
Test 2	11	11	0	2	100.0	84.6
Test 3	10	10	0	2	100.0	83.3
Test 4	12	11	1	2	91.7	84.6
Test 5	11	11	0	2	100.0	84.6
				Average	**96.5**	**84.1**

The precision and recall values were computed for all the testing data sets in these five groups (denoted as Test 1, Test 2, etc.) to evaluate the performance of our proposed framework. As shown in Table 5, the “Missed’ column indicates a false negative, which means that the goal events are misclassified as nongoal events; whereas the ‘Misiden’ column represents a false positive, i.e., the nongoal events are identified as goal events.

Consequently, precision and recall are defined as follows:

$$\text{Recall} = \frac{\text{Iden}}{\text{Iden} + \text{Missed}}, \text{ Precision} = \frac{\text{Iden}}{\text{Iden} + \text{Misiden}}.$$

From the above results, we can clearly see that the performance is quite promising in the sense that the average recall and precision values reach 96.5% and 84.1%, respectively. In addition, the performance across different testing data sets is greatly consistent. Furthermore, the dependency on predefined domain knowledge is largely relaxed since an automatic temporal association mining process is adopted in our framework to discover, represent, and apply the characteristic event temporal patterns.

5. Conclusions

As one of the main aspects for video database management, video content analysis attracts great interests from both the academia and industry. In this paper, we propose a novel framework for video event detection, which integrates the strength of feature extraction, temporal analysis, and multimodal data mining. Especially, the proposed hierarchical temporal association mining mechanism offers a robust solution to explore and employ the characteristic temporal patterns with respect to the events of interest. This approach effectively addresses the issues of loose video structure and skewed data distribution. It also largely relaxes the dependency on domain knowledge and contributes to the ultimate goal of automatic content analysis. Using soccer video as the test bed, the experimental results demonstrate the effectiveness and robustness of the proposed framework.

6. Acknowledgement

For Min Chen, it was partially supported by Florida International University Dissertation Year Fellowship. For Shu-Ching Chen, this work was supported in part by NSF EIA-0220562, NSF HRD-0317692, and

Florida Hurricane Alliance Research Program sponsored by the National Oceanic and Atmospheric Administration. For Mei-Ling Shyu, this research was supported in part by NSF ITR (Medium) IIS-0325260.

7. References

[1] M. Chen, S.-C. Chen, M.-L. Shyu, and K. Wickramaratna, "Semantic Event Detection via Temporal Analysis and Multimodal Data Mining," *IEEE Signal Processing Magazine*, Special Issue on Semantic Retrieval of Multimedia, vol. 23, no.2, 2006, pp. 38-46.

[2] S.-C. Chen, M.-L. Shyu, and C. Zhang, "Innovative Shot Boundary Detection for Video Indexing," Edited by Sagarmay Deb, *Video Data Management and Information Retrieval*, Idea Group Publishing, 2005, pp. 217-236.

[3] S.-C. Chen, M.-L. Shyu, C. Zhang, and M. Chen, "A Multimodal Data Mining Framework for Soccer Goal Detection Based on Decision Tree Logic," *International Journal of Computer Applications in Technology*, vol. 27, no. 4, 2006, pp. 312-323.

[4] A. Ekin, A. M. Tekalp, and R. Mehrotra, "Automatic Soccer Video Analysis and Summarization," *IEEE Transactions on Image Processing*, vol. 12, no. 7, 2003, pp. 796-807.

[5] Y.-L. Kang, J.-H. Lim, Q. Tian, and M. S. Kankanhalli, "Soccer Video Event Detection with Visual Keywords," in *Proceedings of IEEE Pacific-Rim Conference on Multimedia*, vol. 3, 2003, pp. 1796-1800.

[6] R. Leonardi, P. Migliorati, and M. Prandini, **"**Semantic Indexing of Soccer Audio-visual Sequences: A Multimodal Approach based on Controlled Markov Chains," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 5, 2004, pp. 634-643.

[7] C. G. M. Snoek and M. Worring, "Multimedia Event-Based Video Indexing Using Time Intervals," *IEEE Transactions on Multimedia*, vol. 7, no. 4, 2005, pp. 638-647.

[8] P.-N. Tan, M. Steinbach and V. Kumar, Introduction to Data Mining, Addison Wesley, ISBN: 0-321-32136-7.

[9] R. Vilalta and S. Ma, "Predicting Rare Events in Temporal Domains," in *Proceedings of IEEE International Conference on Data Mining*, 2002, pp. 474-481.

[10] F. Wang, Y.-F. Ma, H-J. Zhang, and J.-T. Li, "Dynamic Bayesian network based event detection for soccer highlight extraction," in *Proceedings of International Conference on Image Processing*, vol. 1, 2004, pp. 633-636.

[11] X. Zhu, X. Wu, A. K. Elmagarmid, Z. Feng, and L. Wu, "Video Data Mining: Semantic Indexing and Event Detection from the Association Perspective," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 5, 2005, pp. 665-677.