A Time Series Clustering based Framework for Multimedia Mining and Summarization

Radhakrishnan, R.; Divakaran, A.; Xiong, Z.

TR2004-046 May 13, 2004

Abstract

Past work on multimedia analysis has shown the utility of detecting specific temporal patterns for different content genres. In this paper, we propose a unified, content-adaptive, unsupervised mining framework to bring out such temporal patterns from different multimedia genres. We formulate the problem of pattern discovery from video as a time series-clustering problem. We treat the sequence of low/mid level audio-visual features extracted from the video as a time series and perform a temporal segmentation based on eigenvector analysis of the affinity matrix constructed from statistical models estimated from the time series. Our temporal segmentation detects transition points and outliers from a sequence of observations from a stationary background process. We define a confidence measure on each of the detected outliers as the probability that it is an outlier. Then, we establish a relationship between the mining parameters and the confidence measure using bootstrapping and kernel density estimation thereby enabling a systematic method to choose the mining parameters for any application. Furthermore, the confidence measure can be used to rank the detected transitions in terms of their departures from the background process. Our experimental results with sequences of low and mid level audio features extracted from sports video show that highlight events can be extracted effectively as outliers from a background process using the proposed framework. We proceed to show the effectiveness of the proposed framework in bringing out patterns from surveillance videos without any a priori knowledge. Finally, we show that such temporal segmentation into background and outliers, along with the ranking based on the departure from the background, can be used to generate content summaries of any desired length.

ACM SIGMM International Workshop on Multimedia Information Retrieval (MIR)

Publication History:

1. First printing, TR-2004-046, May 2004

A Time Series Clustering based Framework for Multimedia Mining and Summarization

Regunathan Radhakrishnan, Ajay Divakaran and Ziyou Xiong
Mitsubishi Electric Research Laboratory,
Cambridge, MA 02139
E-mail: {regu, ajayd, zxiong}@merl.com

Abstract

Past work on multimedia analysis has shown the utility of detecting specific temporal patterns for different content genres. In this paper, we propose a unified, content-adaptive, unsupervised mining framework to bring out such temporal patterns from different multimedia genres. We formulate the problem of pattern discovery from video as a time series clustering problem. We treat the sequence of low/mid level audio-visual features extracted from the video as a time series and perform a temporal segmentation based on eigenvector analysis of the affinity matrix constructed from statistical models estimated from the time series. Our temporal segmentation detects transition points and outliers from a sequence of observations from a stationary background process. We define a confidence measure on each of the detected outliers as the probability that it is an outlier. Then, we establish a relationship between the mining parameters and the confidence measure using bootstrapping and kernel density estimation thereby enabling a systematic method to choose the mining parameters for any application. Furthermore, the confidence measure can be used to rank the detected transitions in terms of their departures from the background process. Our experimental results with sequences of low and mid level audio features extracted from sports video show that "highlight" events can be extracted effectively as outliers from a background process using the proposed framework. We proceed to show the effectiveness of the proposed framework in bringing out patterns from surveillance videos without any a priori knowledge. Finally, we show that such temporal segmentation into background and outliers, along with the ranking based on the departure from the background, can be used to generate content summaries of any desired length.

1 Introduction

Past work on multimedia content summarization has mostly focussed on detecting known patterns to provide an abstract representation of the content. As a result, today we know what kinds of patterns are useful for summarizing a particular genre of multimedia and how to extract them using supervised statistical learning tools. For news video, detection of story boundaries either by closed caption and/or speech transcript analysis or by using speaker segmentation and face information have been shown to be useful [10][13]. For situation comedies, detection of physical setting using mosaic representation of a scene and detection of major cast using audio-visual cues have been shown to be useful [1][14]. For sports video summarization, one approach has been to detect domain-specific events & objects that are correlated with highlights using audio visual cues [3][9]. The other approach has been to extract play-break segments in an unsupervised manner [7]. For movie content, detection of syntactic structures like two-speaker dialogues and also detection of specific events like explosions have been shown to useful[4][14]. For surveillance content, detection of "unusual" events using object segmentation and tracking from video has been shown to be effective[2].

Even though we know what kinds of patterns are to be detected for particular genre, the detection task itself can be challenging due to the intra-genre variations as a result of differing multimedia production styles between content providers and other such factors. In surveillance applications, since we don't even know what kind of patterns exist, we cannot build supervised models for event detection. Clearly, in such scenarios there is again a need for a mining framework that can deal with intra-genre variations (in scenarios where we already know what kind of patterns we are looking for) and can act as a pre-processing stage to help build supervised models for some consistent patterns that are brought out by the mining framework.

We layout some of the requirements of a multimedia mining framework for summarization below:

- It should be content-adaptive and unsupervised.
- It should have a common feature extraction and statistical analysis framework to bring out patterns. A supervised post processing stage can act upon discovered patterns to pick only the "interesting" ones as the meaning of "interesting" changes depending on the application and genre.
- It is also desirable to have a ranking scheme for the summary candidates so as to enable summary length modulation.

In this paper, we propose such a framework keeping in mind the aforementioned requirements. It is based on a time-series analysis of low-level audio-visual features followed by segmentation using eigenvector analysis. Our approach treats feature extraction and mining as genre independent pre-processing steps. A genre dependent post-processing is performed on the discovered patterns to present an "interesting" summary to the end user. We apply this framework to two different genres namely sports and surveillance.

The rest of the paper is organized as follows. In the next section, we first formulate the multimedia mining problem and then propose our framework. We evaluate its performance on a synthetic time series data and thus derive the relationship between the tuning parameters and mining performance. In section 3, we present mining results on different genres. In section 4, we discuss our results and present our conclusions.

2 Multimedia Mining Framework

Our proposed framework is motivated by the observation that "interesting" events in multimedia happen sparsely in a background of usual or "uninteresting" events. Some examples of such events are:

- **sports**: A burst of overwhelming audience reaction following a highlight event in a background of commentator's speech.
- **situation comedy**: A burst of laughter following a comical event in a background of dialogues.
- **surveillance**: A burst of motion and screaming noise following a suspicious event in a silent or static background.

This motivates us to formulate the problem of mining for "interesting" events in multimedia as that of detecting outliers or "unusual" events by statistical modelling of a stationary background process in terms of low/mid-level audio-visual features. Note that the background process may be stationary only for small period of time and can change over time. This implies that background modelling has to be performed adaptively throughout the content. It also implies that it may be sufficient to deal with one background process at a time and detect outliers. In the following subsection, we elaborate on this more formally.

2.1 Problem formulation

Let C_1 represent a realization of the dominant or "usual" class and can be thought of as the background process. Let C_2 represent a realization of "unusual" class and can be thought of as the foreground process. Given any time sequence of observations or low-level audio-visual features from the two the classes of events (C_1 and C_2), such as

$$...C_1C_1C_1C_1C_1C_2C_2C_1C_1C_1...$$

then the problem of mining for unusual events is that of finding C_2 and the corresponding times of occurrences of its realizations.

To begin with, the statistics of the class C_1 are assumed to be stationary. However, there is no assumption about the class C_2. The class C_2 can even be a collection of a diverse set of random processes. The only requirement is that the number of occurrences of C_2 is relatively rare compared to the number of occurrences of the dominant class. Note that this formulation is a special case of a more general problem namely clustering of a time series in which a single highly dominant process does not necessarily exist. We treat the sequence of low/mid level audio-visual features extracted from the video as a time series and perform a temporal segmentation to detect transition points and outliers from a sequence of observations.

Before we present our mining framework, we need to review the related theoretical background on the graph theoretical approach to clustering, as well as on kernel density estimation.

2.2 Segmentation using eigenvector analysis of affinity matrix

Segmentation using eigenvector analysis has been proposed in [5] for images. This approach to segmentation is related to graph theoretic formulation of grouping. The set of points in an arbitrary feature space are represented as a weighted undirected graph where the nodes of the graph are points in the feature space and an edge is formed between every pair of nodes. The weight on each edge is the similarity between nodes. Let us denote the similarity between nodes i and j, as $w(i, j)$.

In order to understand the partitioning criterion for the graph, let us consider partitioning it into two groups **A** and **B** and $A \bigcup B = V$.

$$N_{cut}(A, B) = \frac{\sum_{i\epsilon A, j\epsilon B} w(i, j)}{\sum_{i\epsilon A, j\epsilon V} w(i, j)} + \frac{\sum_{i\epsilon A, j\epsilon B} w(i, j)}{\sum_{i\epsilon B, j\epsilon V} w(i, j)} \quad (1)$$

It has been shown in [5] that minimizing N_{cut}, minimizes similarity between groups while maximizing association within individual groups. Shi and Malik show that

$$min_x N_{cut}(x) = min_y \frac{y^T(D - W)y}{y^T Dy} \quad (2)$$

with the condition $y_i \epsilon \{-1, b\}$. Here W is a symmetric affinity matrix of size $N \times N$ (consisting of the similarity between nodes i and j, $w(i, j)$ as entries) and D is a diagonal matrix with $d(i, i) = \sum_j w(i, j)$. x & y are cluster indicator vectors i.e. if **y(i)** equals -1, then feature point **'i'** belongs to cluster A else cluster B. It has also been shown that the solution to the above equation is same as the solution to the following generalized eigenvalue system if y is relaxed to take on real values.

$$(D - W)y = \lambda Dy \quad (3)$$

This generalized eigenvalue system is solved by first transforming it into the standard eigenvalue system by substituting $z = D^{\frac{1}{2}}y$ to get

$$D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}}z = \lambda z \quad (4)$$

It can verified that $z_0 = D^{\frac{1}{2}}\overrightarrow{1}$ is a trivial solution with eigenvalue equal to 0. The second generalized eigenvector (the smallest non-trivial solution) of this eigenvalue system provides the segmentation that optimizes N_{cut} for two clusters.

2.3 Kernel density estimation

Given a random sample $x_1, x_2, ...x_n$ of n observations of d-dimensional vectors from some unknown density (f) and a kernel (K), an estimate for the true density can be obtained as:

$$\widehat{f}(x) = \frac{1}{nh^d}\sum_{i=1}^{n} K(\frac{x - x_i}{h}) \quad (5)$$

where h is the bandwidth parameter. If we use the mean squared error (MSE) as a measure of efficiency of the density estimate, the tradeoff between bias and variance of the estimate can be seen as shown below:

$$MSE = E[\widehat{f}(x) - f(x)]^2 = Var(\widehat{f}(x)) + Bias(\widehat{f}(x))]^2 \quad (6)$$

It has been shown in [8] that the bias is proportional to h^2 and the variance is proportional to $n^{-1}h^{-d}$. Thus, for a fixed bandwidth estimator one needs to choose a value of h that achieves the optimal tradeoff. We use a data driven bandwidth selection algorithm proposed in [12] for the estimation.

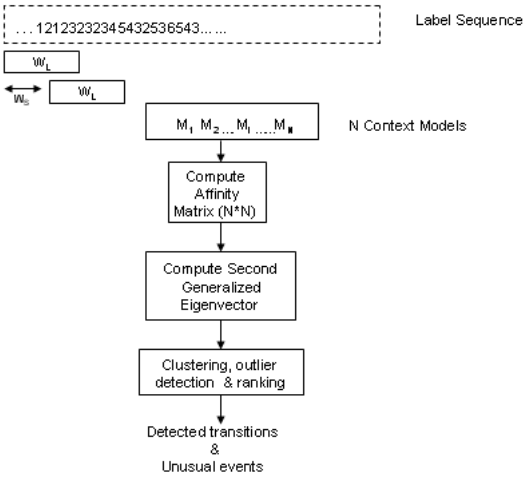

. . . 1212323234 5432536543... ...
Label Sequence
W_L
W_S
W_L
M_1 M_2 ... M_i .....M_N
N Context Models
Compute Affinity Matrix (N*N)
Compute Second Generalized Eigenvector
Clustering, outlier detection & ranking
Detected transitions & Unusual events

Figure 1: Mining framework

2.4 Proposed pattern discovery framework

Given the problem of detecting times of occurrences of C_1 & C_2 from a time series of observations from C_1 and C_2, we propose the following time series clustering framework:

- Sample the input time series on a uniform grid. Let each time series sample (consisting of a sequence of observations) be referred to as context.
- Statistically model the time series observations within each context.
- Compute the affinity matrix for the whole time series using the context models and a commutative distance metric defined between two context models.
- Use the second generalized eigenvector of the computed affinity matrix to identify distinct clusters & outlier context models.

Figure 1 illustrates the above mining framework. In this framework, there are three key issues namely the statistical model for the context and the choice of the two parameters, W_L and W_S. The choice of the statistical model for the time series sample in a context would depend on the underlying background process. A simple unconditional PDF estimate would suffice in case of a memoryless background process. However, if the process has some memory, the chosen model would have to account for it. For instance, a Hidden Markov Model would provide a first order approximation.

The choice of the two parameters (W_L and W_S) would be determined by the confidence with which one wants to say that something is an "unusual" event.The size of the window W_L determines the reliability of the statistical model of a context.The size of the sliding factor, W_S, determines the resolution at which unusual event is detected. In the following subsection, we analyze how W_L determines the confidence on the detected unusual event.

2.5 Confidence measure on detected unusual events

In the proposed time series clustering framework, we are first attempting to estimate the parameters of the background process from the observations within W_L . Then, we measure how different it is from other context models. The difference is caused, either by the observations from C_2 within W_L or by the variance of the estimate of the background model. If the observed difference between two context models is "significantly higher than allowed" by the variance of the estimate itself, then we are "somewhat confident" that it was due to the corruption of one of the contexts with observations from C_2.

In the following, we quantify what is "significantly higher than allowed" and what is "somewhat confident" in terms W_L for two types of background models that we will be dealing with.

2.5.1 Confidence measure for Binomial & Multinomial PDF Model for the background process

For the background process to be modelled by binomial or multinomial PDF, the observations have to be discrete labels. Let us represent the set of discrete labels (the alphabet of C_1 and C_2) by $S = \{A, B, C, D, E\}$. Given a context consisting of W_L observations from S, we can estimate the probability of each of the symbols in S using the relative frequency definition of probability.

Let us represent the unbiased estimator for probability of the symbol A as $\widehat{p}_A$. $\widehat{p}_A$ is a binomial random variable but can be approximated by a Gaussian random variable with mean as p_A and variance as $\sqrt{\frac{p_A(1-p_A)}{W_L}}$ when $W_L \geq 30$.

As mentioned earlier, in the mining framework we are interested in knowing the confidence interval of the random variable ,d, which measures the difference between two estimates of context models. For mathematical tractability, let us consider the Euclidean distance metric between two PDF's, even though it is only a monotonic approximation to a rigorous measure such as the Kullback-Leibler distance.

$$d = \sum_{i \epsilon S} (\widehat{p_{i,1}} - \widehat{p_{i,2}})^2 \tag{7}$$

Here $\widehat{p}_{i,1}$ and $\widehat{p}_{i,2}$ represent the estimates for the probability of i^{th} symbol from two different contexts of size W_L. Since $\widehat{p}_{i,1}$ and $\widehat{p}_{i,1}$ are both Gaussian random variables, d is a χ^2 random variable with n degrees of the freedom where n is the cardinality of the set S.

Now, we can assert with certain probability,

$$N\% = \int_L^U f_{\chi_n^2}(x)dx \tag{8}$$

that any estimate of d $(\widehat{d})$ lies in the interval [L,U]. In other words, we can be N% confident that the difference between two context model estimates outside this interval was caused by the occurrence of C_2 in one of the contexts. Also,we can rank all the outliers using the probability density function of d.

To verify the above analysis, the following simulation was performed. We generated two contexts of size W_L from a known binomial or multinomial PDF (assumed to be the background process). Let us represent the models estimated from these two contexts by M_1 and M_2 respectively. Then, we use Bootstrapping and kernel density estimation to verify the analysis on PDF of d as shown below:

1. Generate W_L symbols from M_1 and M_2.
2. Re-estimate the model parameters ($\widehat{p}_{i,1}$ and $\widehat{p}_{i,2}$) based on the generated data and compute the chosen distance metric (d) for comparing two context models.
3. Repeat steps 1 and 2, N times.
4. Use kernel density estimation to get the PDF of d, $\widehat{p}_{i,1}$ & $\widehat{p}_{i,2}$.

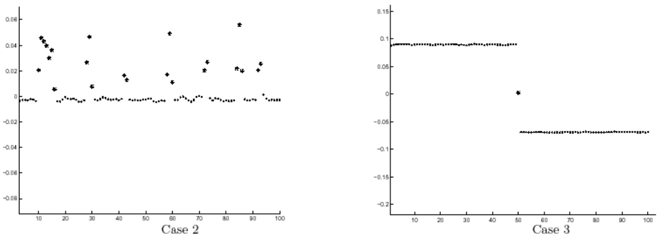


Figure 2: Experiments with synthetic time series data for case 2 and case 3

Figure 3 shows the estimated PDFs for binomial model parameters for two different context sizes and also the PDF of the defined distance metric between the context models. $\widehat{p}_{i,1}$ and $\widehat{p}_{i,2}$ are Gaussian random variables in accordance with Demoivre-Laplace theorem and hence the PDF of distance metric is χ^2 with two degrees of freedom. Hence, any difference caused by the occurrence of symbols from another process (C_2) can be quantified using the PDF of the distance metric. Also, note that the variance of the distance metric decreases as the number of observations within the context increases. Figure 3 shows the PDF estimates for the case of multinomial PDF model for the background. Note that the PDF estimate for the distance metric is χ^2 with 4 degrees freedom which is consistent with the number of symbols in the used multinomial PDF model.

2.5.2 Confidence measure for GMM & HMM models for the background process

When the observations of the memoryless background process are not discrete labels, one would model its PDF using a Gaussian Mixture Model(GMM). If the process has first order memory, one would model its first-order PDF using a Hidden Markov Model (HMM). Let $\lambda = (A, B, \pi)$ represent the model parameters for both the HMM and GMM where A is the state transition matrix, B is the observation symbol probability distribution and π is the initial state distribution. For a GMM A and π are simply equal to 1 and B represents the mixture model for the distribution. For a HMM with continuous observations, B is a mixture model in each of the states. For a HMM with discrete labels as observations, B is a multinomial PDF in each of the states. Two models (HMMs/GMMs) that have different parameters can be statistically equivalent [6] and hence the following distance measure is used to compare two context models (λ_1 and λ_2 with observation sequences O_1 and O_2 respectively.

$$D(\lambda_1, \lambda_2) = \frac{1}{W_L}(\log P(O_1|\lambda_1) + \log P(O_2|\lambda_2) \\ - \log P(O_1|\lambda_2) - \log P(O_2|\lambda_1)) \tag{9}$$

The first two terms in the distance metric measure the likelihood of training data given the estimated models. The last two cross terms measure the likelihood of observing O_2 under λ_1 and vice versa. If the two models are different, one would expect the cross terms to be much smaller than the first two terms. The defined distance metric doesn't lend itself to a similar analysis as in the case of binomial and multinomial models that can help us find its pdf. Hence, we apply bootstrapping to get several observations of the distance metric and use kernel density estimation to get the PDF of the defined distance metric.

Figure 3 shows the PDF of the log likelihood differences for GMMs for different sizes of context. Note that the support of the PDF decreases as W_L increases from 100 to 600. The reliability of the two context models for the same background process increases as the amount of training data increases and hence the variance of normalized log likelihood difference decreases. Therefore, it is possible to quantify any log likelihood difference value caused by corruption of observations from another process (C_2). Similar analysis shows the same observations hold for HMMs as context models as well. Figure 3 shows the PDF of the log likelihood differences for HMMs for different sizes of the context.

2.5.3 Using confidence measures to rank outliers

In the previous two sections, we looked at the estimation of the PDF of a specific distance metric for context models (memoryless models and HMMs) used in the proposed mining framework. Then, for a given time series of observations from C_1 and C_2, we compute the affinity matrix for a chosen size of W_L for the context model. We use the second generalized eigenvector to detect inliers and outliers. Then, the confidence metric for an outlier context, M_j is computed as:

$$p(M_j \epsilon O) = \frac{1}{\#I}(\sum_{i \epsilon I} P_{d,i}(d \leq d(M_i, M_j))) \tag{10}$$

where $P_{d,i}$ is the density estimate for the distance metric using the observations in the inlier context i. O and I represent the set of outliers and inliers respectively. If the density estimate obtained (either through bootstrapping and kernel density estimation or through a similar analysis as for binomial and multinomial cases) has a finite support, some of the outliers that are very distinct from the inliers cannot be ranked as $P_{d,i}(d \leq d(M_i, M_j))$ will be equal to 1 for all of them. In such cases, the distance itself can be used to rank the outliers. The order of ranking will not be affected by the use of $d(M_i, M_j)$ instead of $P_{d,i}(d \leq d(M_i, M_j))$ as the cumulative distribution function (CDF) is a monotonically increasing function. However, the use of $d(M_i, M_j)$ will make it difficult to merge ranked lists as the meaning of $d(M_i, M_j)$ is dependent on the background.

2.6 Results with synthetic time series data

In order to illustrate the effectiveness of the proposed mining framework, we first work with synthetic time series data. The time series generation framework is the same as in [11].

In this framework, we have a generative model for both C_1 and C_2 and the dominance of one over the other can also be governed by a probability parameter. It is also possible to control the percentage of observations from C_2 in a given context.

There are three possible scenarios one can consider with two processes C_1 and C_2 generating label sequences:

- case 1: Sequence completely generated from C_1. This case is trivial and less interesting.
- case 2: Sequence dominated by observations from C_1 i.e. $P(C_1) >> P(C_2)$. The eigenvector analysis of the affinity matrix for this case is shown in Figure 2. There are outliers at times of occurrence of C_2 .
- case 3: Sequence with observations from C_1 and C_2 with no single dominant class i.e $P(C_1) \approx P(C_2)$. The eigenvector analysis for this case (shown in Figure 2) suggests the existence of two clusters and also shows the transition point.

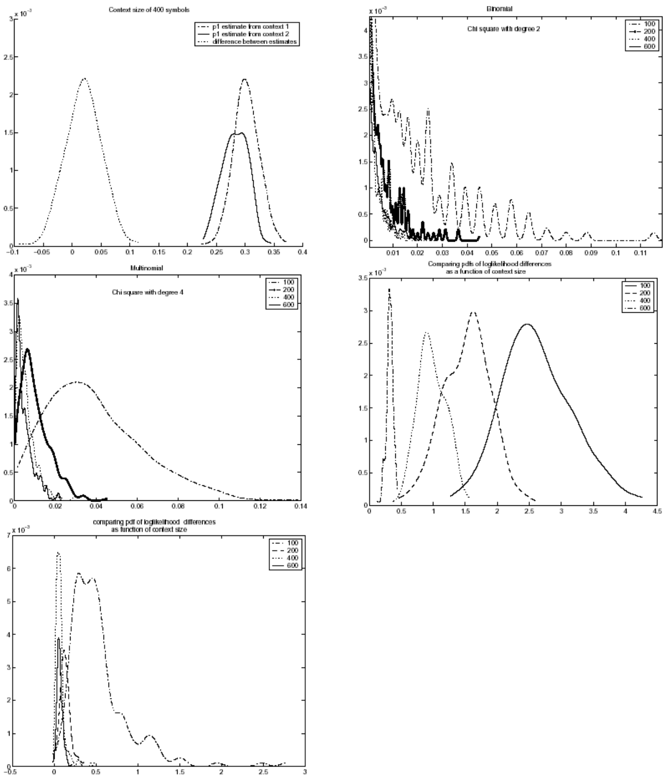


Figure 3: PDFs of distance metrics for different background models

3 Experimental Results

In this section, we present mining results with two different content genres mainly using low-level audio features and semantic audio classification labels at the frame-level and second-level. Since the proposed framework is a formalization and a superset of a well tested heuristic for sports highlights extraction proposed in [15], we chose a data set consisting of just one 3 hr long golf game and a 2 hr long soccer game for the experiments in sports. For surveillance, we chose 30 min of elevator surveillance data and 30 min of traffic intersection video.

3.1 Feature extraction & Classification Framework

We collected training data from several hours of broadcast video data for the following sound classes namely Applause, Cheering, Female speech, Laughter, Male speech & Music. Then, we extracted 12 Mel Frequency Cepstral Coefficients (MFCC) for every 8ms frame and logarithm of energy, from all the clips in the training data. We trained Gaussian Mixture Models (GMMs) to model the distribution of features for each of the sound classes. The number of mixture components were found using the minimum description length principle. Then, given a test clip, we extract the features for every frame and assign a class label corresponding to the sound class model for which the likelihood of the observed features is maximum. For all the experiments to be described in the following sections, we use one of the following time series to mine for "interesting" events at different scales:

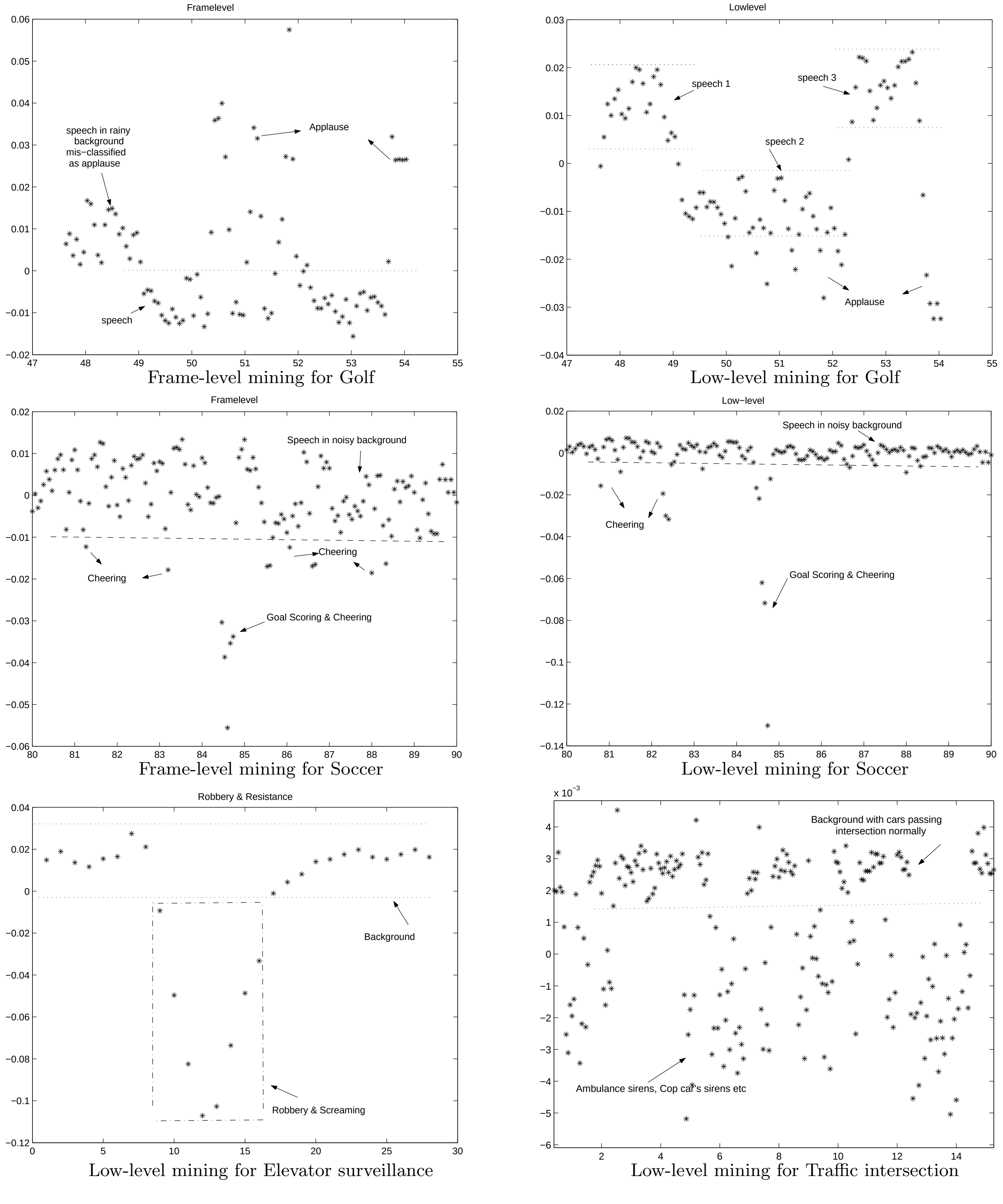
- Time series of 12 MFCC features and logarithm of energy extracted for every frame of 8ms.
- Time series of classification labels for every frame.
- Time series of classification labels for every second of audio. The most frequent frame label in one second is assigned as the label for that second.

3.2 Application to sports video

As mentioned earlier, "interesting" events in sports video happen in a background of the usual process. In a golf game, the usual process is the commentator's speech itself. In a soccer game, the usual process is the commentator's speech in a relatively noisy background. But, in order to extract the program segments from the whole video, we use the same mining framework at a coarser scale as described in the next section. It is based on the observation that commercials are "unusual" in the background of the whole program.

3.2.1 Mining using second-level labels to extract program segments

Since the proposed mining framework assumes that the background process is stationary for the whole time series, our first step is to cluster the time series from the whole sports video to identify the contiguous sections of the time series that have the same background. Figure 5 shows the affinity matrix for a 3 hour long golf game. We used 2-state HMMs to model each time series of 120 (W_L) classification labels with a step size of 10 (W_S). The affinity matrix was constructed using the computed pairwise likelihood distance metric defined earlier. Note that the affinity matrix shows dark regions against a single background. The dark regions



Frame-level mining for Golf

Low-level mining for Golf

Frame-level mining for Soccer

Low-level mining for Soccer

Low-level mining for Elevator surveillance

Low-level mining for Traffic intersection

Figure 4: Comparison of mining with low-level audio features and frame-level classification labels for sport & surveillance

(outliers) were verified to be times of occurrences of commercial sections. Since we use the time series of the labels at one second resolution, the detected outliers give a coarse segmentation of the whole video into two clusters: the segments that represent the program and the segments that represent the commercials. Also, such a coarse segmentation is possible only because we used a time series of classification labels instead of low-level features. The use of low-level audio features at this stage may bring out some fine scale changes that are not relevant for distinguishing program segments from non-program segments. For instance, low-level features may distinguish two different speakers in the content while a more general speech label would group them as one.

3.2.2 Mining for "highlights" from the extracted program segments

Highlight events together with audience reaction in sports video last for only few seconds. This implies that we cannot mine the second-level classification labels to extract highlight events. If we use second-level classification labels, the size of W_L has to be small enough to detect events at that resolution. However, our analysis on the confidence measures earlier, indicates that a small value of W_L, would lead to a less reliable context model thereby producing a lot of false alarms. Therefore, we are left with atleast the following two options:

- To mine the time series of frame-level classification labels instead of second-level labels.
- To mine the time series of low-level MFCC features.

Clearly, mining the frame-level classification labels is computationally more efficient. Also, as pointed out earlier, working with labels can suppress irrelevant changes in the background process like speaker changes. Figure 4 shows the second generalized eigenvector of the affinity matrix for a section of golf program segment. The size of W_L used was equal to 8s of frame level classification labels with a step size of 4s. The context model used for classification labels was a 2-state HMM. In case of mining with low-level features, the size of W_L was equal to 8s of low-level features with a step size of 4s. The context model was a 2-Component GMM. Note that there are outliers at times of occurrences of applause segments in both cases. In the case of low-level feature mining, there were atleast two clusters of speech as indicated by the plot of eigenvector and affinity matrix. Speech 3 (marked in the figure) is an interview section where a particular player is being interviewed. Speech 1 is the commentator's speech itself during the game. Since we used low-level features, these time segments appear as different clusters. However, the eigenvector from frame-level mining affinity matrix shows a single speech background from the 49^{th} min to the 54^{th} min. However, the outliers from the 47^{th} min to the 49^{th} min in the framelevel mining results were caused by mis-classification of speech in "windy" background as applause. Note that the low-level feature mining doesn't have this false alarm. In summary, low-level feature mining is good only when there is a stationary background process in terms of low-level features. In this example, stationarity is lost due to speaker changes. Frame-level mining is susceptible to noisy classification and can bring out false outliers.

Figure 4 shows the mining results of frame level mining and low-level mining, for 10 min of a soccer game with the same set of mining parameters as for the golf game. Note that both of them show the goal scoring moment as an outlier. However, the background model of low-level feature mining has smaller variance than the background model of frame-level label mining. This is mainly due to the classification errors at the frame levels for soccer audio.

	[1]	[2]	[3]	[4]
[1]	1.00	0.00	0.00	0.00
[2]	0.00	0.93	0.00	0.07
[3]	0.00	0.00	0.97	0.03
[4]	0.00	0.00	0.10	0.90

Table 1: Recognition Matrix (Confusion Matrix) on a 70% training/30% testing split of a data set composed of 4 audio classes [1]: Neutral speech; [2]: Foot steps; [3]: Banging; [4]: Non-neutral or Excited speech; Average recognition rate = 95%

3.3 Application to surveillance video

In the case of sports video mining, we used some a priori knowledge about the domain to train sound classes such as applause, cheering etc to extract two more time series apart from the time series of low-level features. In surveillance, often we don't know beforehand what kinds of sounds can characterize the given data and help us detect unusual events. We show that the proposed mining framework provides a systematic methodology to acquire domain knowledge to identify "distinguishable" sound classes. Without any a priori knowledge, we use low-level features in such scenarios to effectively characterize the domain and detect events.

3.3.1 Mining elevator surveillance video

In this section, we apply the mining procedure on a collection of elevator surveillance video data. The data set contains recordings of suspicious activities in elevators as well as some event free clips. A 2 component GMM was used to model the PDF of the low-level audio features in the 8s context. Figure 4 shows the second generalized eigenvector and the affinity matrix for one such clip with a suspicious activity.

In all the clips with suspicious activity, the outliers turned out to be clips of banging sound against elevator walls and excited speech. Since we now know what audio classes are highly correlated to suspicious activity, we trained supervised models (GMMs) for each of the following classes: Normal speech, Foot Steps, Bang, Excited or Non-neutral speech.

Table 1 presents the classification results for these audio classes. The audio classes of neutral speech and foot steps characterize the background process (C_1) whereas short bursts of excited speech and banging sounds correlate with the unusual event in this scenario. After extracting the audio labels, the mining procedure can be repeated with the discrete audio labels as well to detect events.

3.3.2 Mining traffic intersection video

We clipped 30 min from a 2 hr 40 min long traffic intersection video for mining. The video consists of clips where cars cross an intersection without an event. It also has an accident

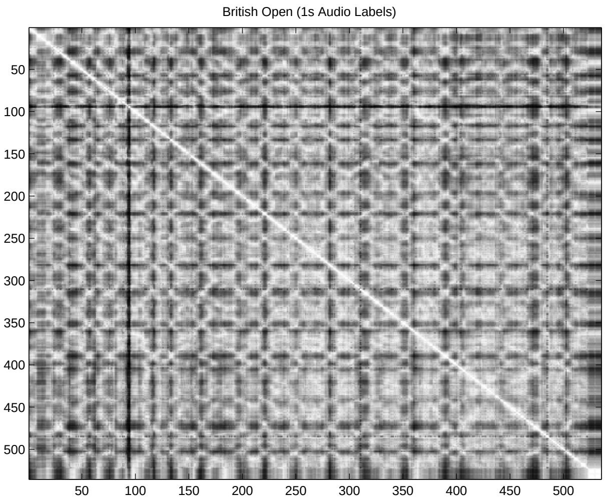


Figure 5: Affinity matrix for a 3 hour long British Open golf game using Audio Classification Labels

event and a number of ambulances and police cars crossing the intersection. The proposed framework was used with the following parameters to mine for outliers: W_L = low-level features for 8s with $W_S = 4$s. The context model used was 2-component GMM. Figure 4 shows the affinity matrix and the second generalized eigenvector for the first 15 min of this content. It was observed that there were outliers whenever an ambulance crossed the intersection. The accident that occurred with a crashing sound at the 24^{th} min was also an outlier.

4 Conclusion and Future Work

We proposed a unified, content-adaptive, unsupervised mining framework that is based on statistical modelling of a time series of low-level features or mid-level semantic labels followed by segmentation using graph theoretic formulation of grouping. The proposed framework also gives a confidence measure on the detected events. The effectiveness of the framework was demonstrated in two different domains: sports video and surveillance video using time series mining of low-level and mid-level audio features. In case of sports, the mining framework was applied at two different resolutions. First, at a coarser resolution for eliminating commercial segments and then at finer resolution for every contiguous program segment to extract highlight moments. The proposed framework was also applied to elevator surveillance video to detect suspicious activity in elevators using low-level audio features. It was applied successfully to mine for events in a traffic intersection video. The effectiveness with surveillance content suggests that the framework can applied to a completely new domain to get some domain knowledge about the kinds of distinguishable patterns that exist for a chosen feature.

In our future work, we would extend this framework to drama and news videos as well. Also, we would incorporate motion and color features from video as well in the mining process.

References

[1] A.Aner and J.R.Kender . Video summaries through mosaic-based shot and scene clustering. *Proc. European Conference on Computer Vision* (2002).

[2] G. Wu, Y. Wu, L. Jiao, Y.-F. Wang, and E. Chang. Multi-camera spatio-temporal fusion and biased sequence-data learning for security surveillance. *ACM Multimedia* (2003).

[3] H.Pan, P.Van Beek and M.I.Sezan. Detection of slow-motion replay segments in sports video for highlights generation. *Proc. IEEE International Conf. on Acoustics, Speech and Signal Processing* (2001).

[4] H.Sundaram and S.F.Chang. Determining computable scenes in films and their structures using audio-visual memory models. *ACM Multimedia* (2000).

[5] J.Shi and J. Malik . Normalized cuts and image segmentation. *Proc. of IEEE Conference on Computer Vision and Pattern Recognition* (1997).

[6] L.Rabiner and B.H.Juang. Fundamentals of speech recognition. *Prentice Hall Signal Processing Series* (1993), 364,365.

[7] L.Xie, S.-F.Chang, A.Divakaran, H.Sun . Unsupervised mining of statistical temporal structures in video. *Video Mining, Azreil Rosenfeld, David Doermann, Daniel Dementhon Eds, Kluwer Academic Publishers* (2003).

[8] M.P.Wand and M.C.Jones. Kernel smoothing. *London:Chapman & Hall* (1995).

[9] M.Xu, N.Maddage, C.Xu, M.Kankanhalli, Q.Tian. Creating audio keywords for event detection in soccer video. *Proc. of ICME* (2003).

[10] Rainer Lienhart . Automatic text recognition for video indexing. *Proc. ACM Multimedia* (1996).

[11] R.Radhakrishnan, Z.Xiong, A.Divakaran and T.Kan. Time series analysis and segmentation using eigenvectors for mining semantic audio label sequences. *ICME* (2004).

[12] S.J.Sheather and M.C.Jones. A reliable data-based bandwidth selection method for kernel density estimation. *J.R. Statist. Society* (1991).

[13] Winston Hsu and Shih-Fu Chang. A statistical framework for fusing mid-level perceptual features in news story segmentation. *Proc. of ICME* (2003).

[14] Y.Li and C.C.Kuo . Content-based video analysis,indexing and representation using multimodal information. *Ph.D Thesis, University of Southern California* (2003).

[15] Z. Xiong, R. Radhakrishnan, A. Divakaran and T.S. Huang. Audio-based highlights extraction from baseball, golf and soccer games in a unified framework. *Proc. of ICASSP* (2003).