Shots Temporal Prediction Rules for High-Dimensional Data of Semantic Video Retrieval

SHAIMAA TORIAH
Benha University
Faculty of Computers and Informatics
Computer Science Department
EGYPT

ATEF GHALWASH
Helwan University
Faculty Of computers and information
Computer Science Department
EGYPT

ALIAA YOUSSEF
Helwan University
Faculty Of Computers and Information
Computer Science Department
EGYPT

Abstract: Some research in Semantic video retrieval is concerned with predicting the temporal existence of certain concepts. Most of the used methods in those studies depend on rules defined by experts and use ground-truth annotation. The Ground-truth annotation is time consuming and labour intensive. Additionally, it involves a limited number of annotated concepts, and a limited number of annotated shots . Video concepts have interrelated relations, so the extracted temporal rules from ground-truth annotation are often inaccurate and incomplete. However concept detections scores are a large high-dimensional continuous valued dataset, and generated automatically. Temporal association rules algorithms are efficient methods in revealing temporal relations, but they have some limitations when applied on high-dimensional and continuous-valued data. These constraints have led to a lack of research used temporal association rules. So, we propose a novel framework to encode the high-dimensional continuous-valued concept detection scores data into a single stream of characters without loss of important information and to predict a temporal shot behavior by generating temporal association rules.

Key–Words: Semantic Video Retrieval, Temporal Association Rules, Principle Component Analysis, Guassian Mixture Model Clustering, Expectation Maximization Algorithm, Sequential Pattern Discovery Algorithm.

1 Introduction

Tremendous growth in digital devices and digital media has led to the capture and storage of a huge amount of digital videos. As a result, there is an urgent need to manage, analyses, automate and retrieve videos efficiently. One of the most important subjects in video retrieval is semantic video retrieval. Semantic video retrieval is the search and retrieval of videos based on their relevance to a users requirements. Semantic video retrieval still represents a big challenge to researchers, as bridging the gap between the users needs and views and the low level features of videos is a complicated problem that requires a tremendous amount of research. This is called the semantic gap; much research has been done on bridging the semantic gap using various methods and techniques, but it is still an open problem.

Semantic video retrieval involves two aspects. One of them is concerned with concept presence detection according to the context concepts. The other aspect is concerned with temporal concept mining, which predicts the temporal presence of certain concepts in the next shots, so it can enhance or refute the presence of these concepts.

Temporal concept mining relies on the consistency of the video. Temporal concept rule mining may involve expert-made rules, be based on statistical dependency tests, or use information extracted from association rules. Temporal association concept rules are extracted from ground-truth annotation. However, ground-truth annotation involves a limited number of annotated concepts, a limited number of annotated videos, many missing values, and binary values.

The main goal of our paper is to model and automate a framework to reduce the volume of video concept detection score data and extract a compact representation of the temporal concept rules. These rules predict the behaviour of the concepts in the next shot based on the current shot behaviour. The results of our method are tested on the cu-vireo concept detection scores. .

The size of the detection score matrix may exceed 150000X300, which is considered large. Applying temporal association rule learning algorithms on such a large matrix involves many difficulties or is, in some cases, impossible. Some of these difficulties include a long processing time, high space requirements, the huge number of resulting association rules, rule redundancy, and the selection of rule pruning criteria. Thus, most of the studies that apply association rule learning algorithms either use only a slice of the concept detection scores with a small number of concepts or use ground-truth annotation. The major issue with using association rule learning algorithms is that

the association rules cannot be applied on continuous values, i.e., the data should be binary. Although much research has been done on methods for discretizing or categorizing data to try to minimize the loss of information when converting data into binary form, such methods also increase the data dimensionality and do not prevent data loss.

To solve these difficulties, we apply principle component analysis (PCA) in our method to compress the concept detection score matrix without loss of data. Principal component analysis is a form of multidimensional scaling. It is a linear transformation of the variables into a lower dimensional space, which retains the maximal amount of information about the variables. It is a common technique for finding patterns in data of high dimension. This is achieved by transforming to a new set of variables, the principal components (PCs), which are uncorrelated and ordered so that the first few retain most of the variation present in all of the original variables [6].

Then, we cluster video shots using the selected uncorrelated principle components, which contain most of the data variation. More than 25 components can be selected. Therefore, there is an urgent need to apply a clustering algorithm that deals efficiently with high-dimensional data. Our selected clustering technique is the Guassian Mixture Model (GMM), and its parameters are estimated using the expectation maximization algorithm (EM). GMM [6] is also useful for modelling the uncorrelated data. GMM is a parametric probability density function that is represented as a weighted sum of Gaussian component densities. GMMs are commonly used as parametric models of the probability distributions of continuous measurements or features. After the clustering phase, we will have a compact stream of cluster numbers or symbols of length N, where N represents the number of shots.

To extract temporal concept rules, we apply the SPADE algorithm [19]. SPADE was developed by Zaki in 2001. SPADE utilizes combinatorial properties to decompose the original problem into smaller sub-problems that can be independently solved in the main memory using efficient lattice search techniques and simple join operations. All sequences are discovered in only three database scans.

This paper is organized as follows. In Section 2, we review approaches for association rules and temporal rules used in semantic video retrieval. In Section 3, we present our proposed method in detail. Experimental results are reported in Section 4. Finally, we conclude in Section 6 and outline some goals for future work.

2 Related Work

It is very time consuming to upload huge amounts of multimedia content, especially videos, onto the web or even just to store them on storage media. Thus, there is an urgent need for a method to automate, organize, manage, and retrieve videos.

Content-based video retrieval methods are specifically designed for extracting the low level features from videos. Some of them are concerned with shot boundary detection , key frame extraction [5], and feature extraction and analysis [1] . However, the extracted low level features do not cover all the user requirements that are represented in the user queries.

Therefore, many semantic-based video retrieval methods have been proposed to bridge the semantic gap. However, this gap stills represents a challenging problem. Semantic video retrieval is concerned with deducing, reinforcing, or refuting the existence of specific concepts using the context information and concept relationships. These concepts are detected using concept detectors. There are an infinite number of high level concepts that are found in user perspectives, and there is no way to construct concept detectors for this huge number of high level concepts, for which constructing a concept detector is an expensive process. Thus, concept detectors are limited to a few selected concepts [9] [8] [17].

Based on [9], a limited number of reliable concept detectors are constructed in [8]. [9] concludes that the video retrieval systems that use a few thousand concept detectors perform well, even though the individual concept detectors have low detection accuracies. [17] explains how to select the set of concepts for which to construct concept detectors.

In [14], a large scale concept ontology for multimedia (LSCOM) is constructed, and this effort is being led by IBM, Carnegie Melon University, and Columbia University with participation from Cyc Corporation. The Disruptive Technology Office sponsored LSCOM, which was a series of workshops that brought together experts from multiple communities to determine multimedia concepts and their taxonomy. The goal of LSCOM was to achieve a set of criteria such as utility, coverage, observability, and feasibility.

There are two main challenges in semantic video retrieval. The first challenge is to detect those concepts that do not have detectors, and the second challenge is to improve the accuracy of concept detection. Researchers in semantic video retrieval have tried to solve these two challenges by modelling and representing the relationships using ontologies[2], expert-made rules[2], association rules[13], graphs[7],[10],etc.

In [17],[2] inter-concept relationships are mod-

elled using ontologies that are based on the principle that concepts do not appear in isolation but are correlated with one another, and the concept detection is improved by utilizing such related concepts. This is called context-based concept fusion (CBCF).

The [10] refines the annotation of semantic concepts using a graph diffusion technique. In [13], the authors try to exploit the inter-concept association relationships based on concept annotation of video shots to discover the hidden association rules between concepts. These association rules are generated using the Apriori algorithm and are used to improve the detection accuracies of concept detectors. Additionally, there are other research works that are concerned with association rules using [18]. However, they depend on the ground-truth data, in which few concepts are annotated, and a limited number of video shots.

Our work is concerned with temporal concept detection. The following are some research works concerning temporal concept detection.

In [12], it is assumed that temporally adjacent video shots usually share similar visual and semantic content. A thorough study of temporal consistency, defined with respect to semantic concepts and query topics using quantitative measures, is presented, and its implications for video analysis and retrieval tasks are discussed. It is a preliminary analysis that focuses on the video temporal consistency issue and thus focuses on the consistency of adjacent shots, rather than shots in the same neighbourhood. Therefore, the limitation of this work is its failure to consider the consistency of video data beyond the adjacent shots. In [7], a CBCF method called the temporal spatial node balance algorithm (TNSB) is presented, which depends on a physical model. This algorithm refines concept detection scores using a concept fusion task, which depends on the spatial and temporal relationships between concepts. [13] tests whether there is temporal dependence among neighbouring shots using statistical measurements.

Extracting temporal association rules from a huge high-dimensional dataset has some drawbacks, such as requiring a large amount of processing time, requiring a large amount of memory space, and necessitating the extraction of a large number of association rules. Thus, most previous research has been concerned with extracting temporal association rules from either the ground-truth annotations or a small set of concept detection scores. However, this leads to inaccurate temporal association rules due to incomplete and inaccurate data. Therefore, our proposed framework extracts the temporal rules from a large number of continuous high-dimensional data values.

3 Proposed Framework

The main goal of our proposed framework method is to:

1. Compress concept detection scores without loss of data, keep the inter-relationships between concepts, and preserve temporal relationships between video shots.

2. Extract temporal rules for predicting the next shot behaviour, by which we mean that we predict the probability of all concepts existence in the shot by detecting the shot's cluster, rather than predicting the existence of a specific concept, as was done in previous research.

Our proposed method consists of the following steps, as showed in figure [1]:

1. Data Preprocessing.

2. Data modeling using principle component analysis to reduce its dimensionality

3. Clustering shots with Gaussian mixture model and EM algorithm for parameter estimation.

4. Temporal rules extraction process using spade algorithm.

We will explain each step in details in the following subsections.

3.1 Data Preprocessing

As shown in Figure [1], the preprocessing steps are as follows. This step includes loading data and sorting rows according to video numbers and shot numbers to assist in temporal rule detection in the future steps. This step includes the following:

1.
   - Load detection scores from the files, where each file represents the concept detection values for unorganized video shots, into an MxN matrix.
   - Append two columns to the matrix S, the entries of which are the name and shot numbers for each video.
   - Sort the matrix S according to the video numbers and shots numbers.

3.2 Data dimensionality reduction using principle component analysis (PCA)

In this stage, we transform and represent our data using principle component analysis. The goal of principle component analysis is to identify and find patterns to reduce the dimensionality of the dataset with

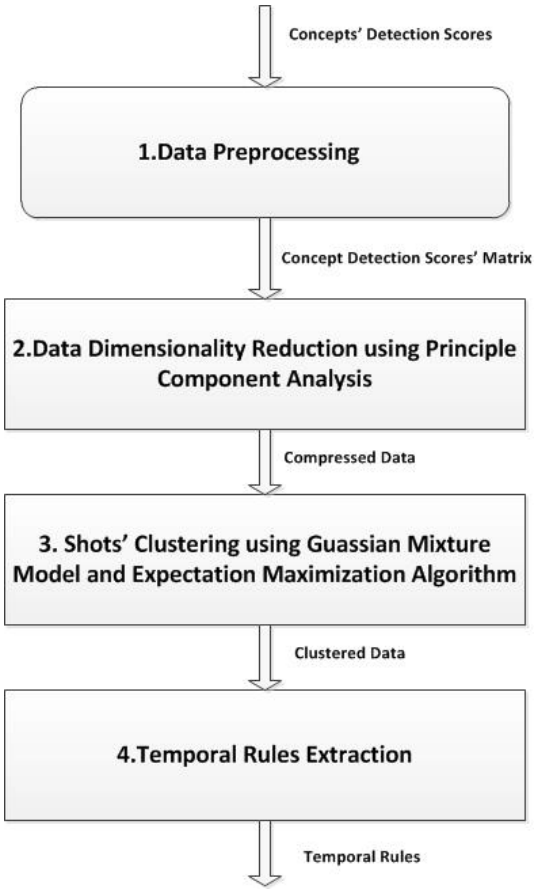


Figure 1: Framework components

minimal loss of information. PCA reduces the dimensionality of our dataset, which consists of a large number of interrelated variables (concepts), while retaining as much of the variation as possible. PCA projects/transforms our concept space of dimension N onto a new smaller subspace of uncorrelated principle component variables, which are constructed as linear combinations of the original concepts (variables), with dimension L, where L≤N [6].

C is the concepts' detection score matrix, M is the number of video shots, and N is the number of concepts, as shown in equation (1).

$$\begin{pmatrix} c_{1,1} & \cdots & c_{1,N} \\ \vdots & \ddots & \vdots \\ c_{M,1} & \cdots & c_{M,N} \end{pmatrix} \quad (1)$$

For i= 1,..,M shots, PCA transform j=1,..,N concepts $(c_1, c_2, ..., c_N)$ into K=1,..,P new uncorrelated variables $(Z_1, Z_2, , Z_P)$ called principle components, as shown in equation (2).

$$\begin{aligned} Z_1 &= e_{11}C_1 + e_{12}C_2 + \ + e_{1P}C_P \\ Z_P &= e_P1C_1 + e_P2C_2 + \ + e_PPC_P \end{aligned} \quad (2)$$

where

Z_K : Value or score of principle component K (of reduced dimension).

C_j : Values of the original (j) concept, of the original dimension.

e_{ik} : Weights or coefficients that indicate how much each original concept contributes to the linear combination used to form principle component K.

The matrix notation is shown in equation (3).

$$Z_k = e_k^{\backslash} C^{\backslash} \quad (3)$$

Where

$e_k^{\backslash}$: The transposed eigenvector of the correlation matrix corresponding to its kth largest eigenvalue u_k.

$C^{\backslash}$:The transposed vector of p concepts.

The eigenvector gives a direction of the data and the corresponding eigenvalue is the variance of the data values in that direction. All the eigenvectors of our concept detection matrix are perpendicular. Thus, the eigenvectors will be ordered according to their eigenvalues, from highest to lowest. Then, we will represent the data according to the new axes (p eigenvectors) obtained in equation (3).

We then represent the data according to the selected components (new axes) by the following general formula in equation (4).

$$Z = e^{\backslash}C^{\backslash} \tag{4}$$

The correlation matrix (Cor) is calculated from the covariance matrix, where the correlation between c_x and c_y measures the strength and direction of the linear relationship between two numerical variables X and Y. The correlation equation is shown in equation (5).

$$Cor(X,Y) = Cov(X,Y)/\sigma x \sigma y \tag{5}$$

where:
Cor(X,Y) : The correlation between concept C_x and concept C_y.
Cov(X,Y) : The covariance between C_x and C_y.
σ_X : The standard deviation of concept C_x.
σ_Y : The standard deviation of concept C_y.

Cov(X,Y) : The covariance between c_x and c_y, which is calculated as shown in equation (6).

$$Cov(x,y) = \frac{\sum_{i=1}^{M}(x_i - \mu_X)(y_i - \mu_Y)}{M-1} \tag{6}$$

Where:
μ_X : The mean values for concept C_x.
μ_Y : The mean values for concept C_y.
X_i : The detection value of concept X for shot i.
Y_i : The detection value of concept y for shot i.
M : Number of video shots.
The standard deviation is calculated as shown in equation (7).

$$\sigma_X = \sqrt{\frac{\sum_{i=1}^{M}(x_i - \mu_X)^2}{M-1}} \tag{7}$$

Where:
σ_X : The standard deviation for concept X.
x_i: The detection score for shot i and concept X.
μ_X The mean value for concept X.
M: The number of shots.

The correlation coefficient has several advantages over the covariance for determining the strengths of relationships.

- The covariance can take any value, while the correlation is limited to values between -1 and +1.
- Because of its numerical limitations, the correlation is more useful for determining how strong the relationship is between two variables:
  - The correlation does not have units. The covariance always has units.
  - The correlation is not affected by changes in the centers (i.e., means) or scales of the variables.

3.3 Shots' Clustering using Gaussian Mixture Models and Expectation Maximization Algorithm

In this stage, the dimension-reduced data are clustered using Gaussian mixture models (GMM) [6] and EM algorithm for parameter estimation.

3.3.1 Guassian Mixture Models for Data Clustering

The dimension-reduced data that were obtained using PCA have many dimensions, the number of which may exceed 25, and most of the standard clustering algorithms may not work with high-dimensional data due to the curse of dimensionality [3], causing the distance measure to become meaningless. This problem led to new clustering algorithms for high-dimensional data, such as subspace- and model-based clustering algorithms.

The Gaussian distribution or normal distribution is one of the most important probability distributions for continuous variables. It estimates uncertainty and requires only two parameters, the mean and variance. Therefore, it is preferable to other distributions, and the symmetry of its bell shape makes it preferable to most of the popular models. The central limit theorem tells us that the expectation of the mean of any random variable converges to a Gaussian distribution [15].

GMM is a model-based clustering algorithm in which each cluster can be mathematically represented by a parametric Gaussian distribution. GMM is a parametric probability density function represented as a weighted sum of Gaussian component densities. GMM latent variables or parameters are estimated from training data using the iterative Expectation-Maximization (EM) algorithm or Maximum A Posteriori (MAP) estimation from a well-trained prior model.

The Gaussian probability density function of a single dimension (univariate) is shown in equation (8).

$$g(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{8}$$

Where:
μ: Mean or expected value of the distribution.
X: Random variable.

σ^2: Variance.
σ: Standard deviation.

The multivariate Gaussian probability density function is a generalization of the one-dimensional (univariate) normal distribution to higher dimensions , as shown in equation(9).

$$g(x|\mu_i, \sum_i) = \frac{1}{(2\pi)^{D/2}|\sum_i|^{1/2}} \exp\{-\frac{1}{2}(x-\mu_i)^T \sum_i^{-1}(x-\mu_i)\} \quad (9)$$

Where:
x: D-dimensional continuous-valued data vector.
μ_i: D-dimensional mean vector .
$\sum_i$: D X D covariance matrix .
$|\sum_i|$:Determinant of $\sum_i$.
D: Number of dimensions.

As stated before, a Gaussian mixture model is a weighted sum of M component Gaussian densities, as given by the following equation.

$$p(x|\lambda) = \sum_{i=1}^{M} w_i g(x|\mu_i, \sum_i) \quad (10)$$

Where:
λ : GMM variants w_i ,μ_i ,$\sum_i$ i = 1,. . ,M.
W_i: Mixture weights for i = 1,. . ,M.
$\sum_i$: Covariance matrix.
μ_i: Mean value of concept i.
$g(x|\mu_i, \sum_i)$: Component Gaussian densities, for i = 1. . . M.

Each component Gaussian density is a D-variate (multivariate) Gaussian function. The mixture weights satisfy the constraint $\sum_{i=1}^{M} W_i = 1$.

3.3.2 Expectation Maximization Algorithm

There are many latent parameters variables, such as mean vectors, covariance matrices and mixture weights from all component densities, in the Gaussian mixture model . These parameters are collectively represented by λ as shown in equation(10).

The expectation maximization (EM) algorithm is used for estimating the parameters in equation(10). The EM algorithm is an powerful method for finding maximum likelihood solutions for models with latent variables.The EM algorithm is an iterative method to find maximum likelihood or maximum a posteriori (MAP) estimates of parameters in statistical models. The EM iteration alternates between performing an expectation (E) step, and a maximization (M) step. The basic idea of the EM algorithm is, beginning with an initial model, to estimate a new model . The new model then becomes the initial model for the next iteration, and the process is repeated until some convergence threshold is reached. During each EM iteration, there are set of re-estimation formulas are used, which guarantee a monotonic increase in the model likelihood values, as found in [11].

3.4 Temporal Rules Extraction

In the final stage, the temporal rules are extracted from the stream of cluster numbers that resulted from the Gaussian mixture model clustering algorithm being applied to the data that were dimension reduced using PCA.

The SPADE algorithm is used in this stage. The SPADE (Sequential Pattern Discovery using Equivalence classes) algorithm is one of the Sequential Pattern mining algorithms. The sequential pattern mining problem was first addressed in [19].

The SPADE algorithm uses a vertical id-list database format, in which we associate with each sequence a list of objects in which it occurs. Then, frequent sequences can be found efficiently using intersections on id lists. The method also reduces the number of database scans and therefore also reduces the execution time.

The first step of SPADE is to compute the frequencies of 1-sequences, which are sequences with only one item. This is done in a single database scan. The second step consists of counting 2-sequences. This is done by transforming the vertical representation into a horizontal representation in memory and counting the number of sequences for each pair of items using a dimensional matrix. Therefore, this step can also be executed in only one scan. Subsequent n-sequences can then be formed by joining (n-1)-sequences using their id lists. The size of an id list is the number of sequences in which an item appears. If this number is greater than minsup, the sequence is a frequent one. The algorithm stops when no more frequent sequences can be found. The algorithm can use either a breadth-first or a depth-first search method for finding new sequences [19]

4 Experimental Results and Discussion

4.1 Experimental setup

The proposed framework is performed on an Intel core(TM) i7-2630 QM CPU @ 2.00 GHZ 2.00 GHZ processor with 6 gigabyte RAM on a 64-bit operating system (Windows 7).

Table 1: "Sample Data of cu_vireo_TV10 sorted according to video and shot number"

	video	shot	Actor	Adult	Airplane	Airplane_Flying	Anchorperson	Animal	Asian_People	Athlete	Basketball	Beach	Beards	Bicycles	B
1	3174	1	0.02955	0.08133	0.00680	0.00514	0.00707	0.02422	0.01219	0.07361	0.00633	0.05799	0.04970	0.01519	
2	3175	1	0.07044	0.15214	0.00391	0.00307	0.06606	0.02248	0.01936	0.00781	0.00281	0.02278	0.07685	0.01305	
3	3175	2	0.03322	0.11896	0.01114	0.00871	0.04732	0.01824	0.01687	0.03193	0.00327	0.03660	0.06727	0.03760	
4	3175	3	0.06507	0.13614	0.00901	0.00698	0.02053	0.01811	0.01425	0.04160	0.00529	0.05281	0.08734	0.05856	
5	3175	4	0.05800	0.19939	0.00516	0.00397	0.05233	0.01920	0.01918	0.00796	0.00273	0.01790	0.07223	0.01750	
6	3175	5	0.04340	0.13499	0.00262	0.00196	0.01503	0.01768	0.01904	0.03080	0.00249	0.03363	0.04265	0.01642	
7	3175	6	0.03776	0.07470	0.00330	0.00250	0.01019	0.01780	0.01230	0.04484	0.00292	0.03776	0.03500	0.00688	
8	3175	7	0.02566	0.05156	0.00290	0.00219	0.01220	0.04312	0.00815	0.02012	0.00210	0.07666	0.02160	0.00473	
9	3175	8	0.06333	0.17284	0.00564	0.00436	0.05661	0.02146	0.02137	0.00716	0.00285	0.01782	0.10654	0.01611	
10	3175	9	0.04936	0.07285	0.00161	0.00127	0.01454	0.01197	0.01117	0.00373	0.00063	0.00527	0.03822	0.00802	
11	3176	1	0.03111	0.05220	0.00898	0.00680	0.00761	0.00698	0.01114	0.00140	0.00117	0.00230	0.03395	0.01838	
12	3176	2	0.04519	0.04666	0.01405	0.01094	0.00977	0.02020	0.01176	0.00259	0.00203	0.00315	0.03742	0.02938	

Showing 1 to 13 of 144,988 entries

All our proposed framework components are implemented using R [16].

4.2 Dataset

The dataset used in our proposed framework is the CU_VIREO_ TV10 set of detection scores[11]. It contains 130 concepts, detected for 150,000 video shots; table 1 contains a sample of these data, sorted according to video number and shot number. The CUVIREO TV10TV10 detection score dataset consists of the latest detection scores provided by cu-vireo374. This dataset is based on models retrained on the TRECVID 2010 development set. The annual NIST TRECVID video retrieval benchmarking event provides benchmark datasets for performing system evaluation. It uses multiple bag-of-visual-words local features computed from various spatial partitions, and it incorporates the DASD algorithm [10].

4.3 The used dataset versus other datasets

The detection score datasets can be obtained from Mediamill-101, Columbia374, Vireo374, or Cu-Vireo 374. However, Media Mill-101 includes 101 more concept detectors than TRECVID 2005/2006. Columbia374 and Vireo374 include 374 detectors for 374 semantic concepts selected from the LSCOM ontology [14]. Columbia374 depends on three types of global features, and Vireo374 emphasizes the use of local key point features. As they work using on the same concepts, their output format is unified and the detection scores of both detector sets are fused to generate the cu-vireo374 detection scores [11]. Cu-vireo is the most suitable dataset for our framework because it detects up to 300 concepts for a huge number of video shots (up to 175,000 video shots).

4.4 Compressed dataset

The Cu-vireo_tv_10 dataset contains the detection scores for 130 concepts for 150,000 video shots. This dataset is loaded into a matrix of size 150000x132. The entries of the additional two columns contain the video number and the shot number in the specified video. These columns are very important for temporal rule detection in the final step. The allocated memory for the original dataset matrix is 161,23,5784 bytes and contains 19,138,416 elements. This matrix is unsuitable for use with sequential pattern mining algorithms such as the SPADE algorithm. Thus, we have to compress this dataset without losing the relationships between concepts.

Table 2: "the first 25 principle components"

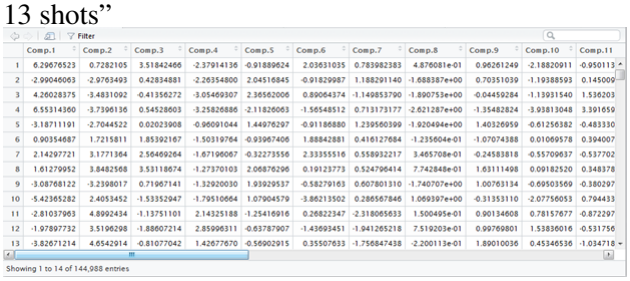
	Comp.1	Comp.2	Comp.3	Comp.4
Standard deviation	5.9556479	4.7657131	3.08260205	2.69879263
Proportion of Variance	0.2728442	0.1747079	0.07309566	0.05602678
Cumulative Proportion	0.2728442	0.4475520	0.52064768	0.57667446
	Comp.5	Comp.6	Comp.7	Comp.8
Standard deviation	2.42476814	2.18681185	2.14533063	1.98397435
Proportion of Variance	0.04522693	0.03678574	0.03540341	0.03027811
Cumulative Proportion	0.62190139	0.65868712	0.69409054	0.72436865
	Comp.9	Comp.10	Comp.11	Comp.12
Standard deviation	1.92894165	1.85834513	1.76393512	1.63666450
Proportion of Variance	0.02862166	0.02656497	0.02393436	0.02060516
Cumulative Proportion	0.75299031	0.77955528	0.80348964	0.82409480
	Comp.13	Comp.14	Comp.15	Comp.16
Standard deviation	1.40381365	1.38713404	1.23909619	1.120956637
Proportion of Variance	0.01515918	0.01480108	0.01181046	0.009665721
Cumulative Proportion	0.83925398	0.85405506	0.86586552	0.875531238
	Comp.17	Comp.18	Comp.19	Comp.20
Standard deviation	1.077136187	1.005920733	0.97157600	0.893705087
Proportion of Variance	0.008924787	0.007783666	0.00726123	0.006143914
Cumulative Proportion	0.884456026	0.892239691	0.89950092	0.905644835
	Comp.21	Comp.22	Comp.23	Comp.24
Standard deviation	0.847579087	0.823567938	0.783055566	0.747212736
Proportion of Variance	0.005526079	0.005217417	0.004716739	0.004294822
Cumulative Proportion	0.911170914	0.916388331	0.921105069	0.925399892
	Comp.25	Comp.26	Comp.27	Comp.28
Standard deviation	0.720906721	0.669854703	0.662860615	0.647337260
Proportion of Variance	0.003997742	0.003451579	0.003379878	0.003223427
Cumulative Proportion	0.929397634	0.932849213	0.936229092	0.939452519

Therefore, we transform the Cu-vireo_tv_10 dataset into a compressed dataset using principle component analysis.

Principle component analysis reduces the dimensionality of the Cu-vireo_tv_10 data, which contain a large number of concepts, by representing them with a small selected number of variables without losing the important data. Principle component analysis represents our data with new dimensions, called principle components. The number of produced principle components is equal to the original number of concepts. These principle components are sorted according to the variance of the data. Thus, the first set of components contains the most important information about our data. In our implementation, we select the first 25 principle components, which contain 92% of the variance of our data, as shown in table 2.

Our new compressed dataset is represented using the first 25 principle components, as shown in table 3. Table 3 shows the first 11 PCs for the first 13 shots. The size of the new compressed matrix is 150,000x25, and it consists of 3,750,000 elements and allocates 37,118,776 bytes.

Table 3: ”the first 11 principle components of the first 13 shots”



	Comp.1	Comp.2	Comp.3	Comp.4	Comp.5	Comp.6	Comp.7	Comp.8	Comp.9	Comp.10	Comp.11
1	6.29676523	0.7282105	3.51842466	-2.37914136	-0.91889624	2.03631035	0.783982383	4.876081e-01	0.96261249	-2.18820911	-0.950113
2	-2.99046063	-2.9763493	0.42834881	-2.26354800	2.04516845	-0.91829987	1.188291140	-1.688387e+00	0.70351039	-1.19388593	0.145009
3	4.26028375	-3.4831092	-0.41356272	-3.05469307	2.36562006	0.89064374	-1.149853790	-1.890753e+00	-0.04459284	-1.13931540	1.536203
4	6.55314360	-3.7396136	0.54528603	-3.25826886	-2.11826063	-1.56548512	0.713173177	-2.621287e+00	-1.35482824	-3.93813048	3.391659
5	-3.18711191	-2.7044522	0.02023908	-0.96091044	1.44976297	-0.91186880	1.239560399	-1.920494e+00	1.40326959	-0.61256382	-0.483330
6	0.90354687	1.7215811	1.85392167	-1.50319764	-0.93967406	1.88842881	0.416127684	-1.235604e-01	-1.07074388	0.01069578	0.394007
7	2.14297721	3.1771364	2.56469264	-1.67196067	-0.32273556	2.33355516	0.558932217	3.465708e-01	-0.24583818	-0.55709637	-0.537702
8	1.61279952	3.8482568	3.53118674	-1.27370103	2.06876296	0.19123773	0.524796414	7.742848e-01	1.63111498	0.09182520	0.348378
9	-3.08768122	-3.2398017	0.71967141	-1.32920030	1.93929537	-0.58279163	0.607801310	-1.740707e+00	1.00763134	-0.69503569	-0.380297
10	-5.42365282	2.4053452	-1.53352947	-1.79510664	1.07904579	-3.86213502	0.286567846	1.069397e+00	-0.31353110	-2.07756053	0.794433
11	-2.81037963	4.8992434	-1.13751101	2.14325188	-1.25416916	0.26822347	-2.318065633	1.500495e-01	0.90134608	0.78157677	-0.872297
12	-1.97897732	3.5196298	-1.88607214	2.85996311	-0.63787907	-1.43693451	-1.941265218	7.519203e-01	0.99769801	1.53836016	-0.531756
13	-3.82671214	4.6542914	-0.81077042	1.42677670	-0.56902915	0.35507633	-1.756847438	-2.200113e-01	1.89010036	0.45346536	-1.034718

Showing 1 to 14 of 144,988 entries

Table 4: ”clustered shots”

	Video_no	Shot_no	Cluster_no
1	3174	1	19
2	3175	1	7
3	3175	2	1
4	3175	3	1
5	3175	4	7
6	3175	5	18
7	3175	6	6
8	3175	7	6
9	3175	8	7
10	3175	9	7
11	3176	1	20
12	3176	2	20
13	3176	3	20
14	3176	4	7
15	3176	5	18
16	3176	6	20
17	3176	7	20
18	3176	8	20

Showing 1 to 18 of 144,988 entries

4.5 Clustered Data

Each video consists of a consistent set of shots, and each shot consists of a set of concepts; each concept is detected by a concept detector. Therefore, each shot is associated with a set of standardized concept detection scores. We cluster shots using a Gaussian mixture model clustering algorithm [4], and each shot is grouped into a cluster. The dimension reduced data will be categorized into 20 clusters using the Gaussian mixture model clustering algorithm. Each cluster represents the shots behaviour category. Finally, we obtain a stream of cluster numbers; see table 4.

4.6 Temporal rules

In this final step, we extract temporal rules from the clustered data. The SPADE algorithm is used to extract temporal rules . The SPADE algorithm parameters are support=0.09 and max window size=10. The matrix input into the SPADE algorithm is as shown in table 5. In table 5, sequence id represents the video number; event id represents the shot number in the current video; size represents the number of items; and items represents the cluster number of the current shot.The extracted temporal rules are shown in table 6. The first temporal rule is 20->16->20. this rule indicates that if we have two consecutive shots in the video , and their clusters numbers are as the following20,and 16 , then the fourth shot cluster is 20 . the temporal rules help in concludes the missing shot behavior by deducing its cluster number according to the suitable rule. then we take the cluster center values to be the missing shot PCs values.

Table 5: ”the prepared temporal data to SPADE algorithm”

	sequenceID	eventID	SIZE	items
1	3174	1	1	19
2	3175	1	1	7
3	3175	2	1	1
4	3175	3	1	1
5	3175	4	1	7
6	3175	5	1	18
7	3175	6	1	6
8	3175	7	1	6
9	3175	8	1	7
10	3175	9	1	7
11	3176	1	1	20

Showing 1 to 11 of 144,988 entries

Table 6: ”the generated temporal rules”

	sequence	support
106	<{20},{16},{20}>	0.12787785
107	<{16},{20},{20}>	0.13336514
108	<{20},{15},{20}>	0.13968746
109	<{15},{20},{20}>	0.14577120
110	<{20},{14},{20}>	0.18799952
111	<{14},{20},{20}>	0.19289037
112	<{20},{1},{20}>	0.10378146
113	<{1},{20},{20}>	0.10974591
114	<{8},{20},{20},{20}>	0.11046165
115	<{20},{8},{20},{20}>	0.09459621
116	<{7},{20},{20},{20}>	0.12143624
117	<{20},{7},{20},{20}>	0.11296672
118	<{20},{20},{7},{20},{20}>	0.09161398
119	<{5},{20},{20},{20}>	0.10580938
120	<{20},{5},{20},{20}>	0.09531194
121	<{20},{20},{20},{20}>	0.25002982
122	<{2},{20},{20},{20}>	0.09376118

Showing 105 to 123 of 354 entries

5 Conclusion and future work

The proposed framework aims to reduce the huge size of the concept detection score matrix without loss of concept relationships and to produce a helpful set of temporal rules for the shots. The resulting temporal rules aim to predict neighbouring shots, the number of which may be 10 or more, according to the maximum window size parameter value in the SPADE algorithm. Using the resulting temporal rules, we can predict the clusters values of future shots representing the shot behaviour. These rules refine our clustered dataset to be more accurate and helpful in semantic video retrieval. Additionally, they help in deducing missing shots. Although principle component analysis is efficient in reducing data dimensionality without loss of information on the relations between the variables in the dataset, the resulting principle components are incomprehensible to the normal user. Thus, in future work, we will use big data processing techniques to extract more comprehensible temporal rules that are more easily understood by the unqualified user.

References:

[1] Muhammad Nabeel Asghar, Fiaz Hussain, and Rob Manton. Video indexing: a survey. *framework*, 3(01), 2014.

[2] Lamberto Ballan, Marco Bertini, Alberto Del Bimbo, and Giuseppe Serra. Video annotation and retrieval using ontologies and rule learning. *IEEE MultiMedia*, 17(4):80–88, 2010.

[3] R Bellman. Dynamic programming princeton university press princeton. *New Jersey Google Scholar*, 1957.

[4] Bergé, Laurent and Bouveyron, Charles and Girard, Stéphane. Hdclassif: An r package for model-based clustering and discriminant analysis of high-dimensional data. *Journal of Statistical Software*, 46(6):1–29, 2012.

[5] Shripad A Bhat, Omkar V Sardessai, Preetesh P Kunde, and Sarvesh S Shirodkar. Overview of existing content based video retrieval systems. *International Journal of Advanced Engineering and Global Technology*, 2, 2014.

[6] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[7] Geng, Jie and Miao, Zhenjiang and Chi, Hai. Temporal-Spatial refinements for video concept fusion. In *Asian Conference on Computer Vision*, pages 547–559. Springer, 2012.

[8] Alexander Hauptmann, Rong Yan, and Wei-Hao Lin. How many high-level concepts will fill the semantic gap in news video retrieval? In *Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 627–634. ACM, 2007.

[9] Alexander Hauptmann, Rong Yan, Wei-Hao Lin, Michael Christel, and Howard Wactlar. Can high-level concepts fill the semantic gap in video retrieval? a case study with broadcast news. *IEEE transactions on multimedia*, 9(5):958–966, 2007.

[10] Yu-Gang Jiang, Qi Dai, Jun Wang, Chong-Wah Ngo, Xiangyang Xue, and Shih-Fu Chang. Fast semantic diffusion for large-scale context-based image and video annotation. *IEEE Transactions on Image Processing*, 21(6):3080–3091, 2012.

[11] Jiang, Yu-Gang and Yanagawa, Akira and Chang, Shih-Fu and Ngo, Chong-Wah. CU-VIREO374: Fusing Columbia374 and VIREO374 for large scale semantic concept detection. *Columbia University ADVENT Technical Report# 223–2008–1*, 2008.

[12] Lin Lin, Mei-Ling Shyu, and Shu-Ching Chen. Association rule mining with a correlation-based interestingness measure for video semantic concept detection. *International Journal of Information and Decision Sciences*, 4(2-3):199–216, 2012.

[13] Ken-Hao Liu, Ming-Fang Weng, Chi-Yao Tseng, Yung-Yu Chuang, and Ming-Syan Chen. Association and temporal rule mining for post-filtering of semantic concept detection in video. *IEEE Transactions on Multimedia*, 10(2):240–251, 2008.

[14] Milind Naphade, John R Smith, Jelena Tesic, Shih-Fu Chang, Winston Hsu, Lyndon Kennedy, Alexander Hauptmann, and Jon Curtis. Large-scale concept ontology for multimedia. *IEEE multimedia*, 13(3):86–91, 2006.

[15] John Rice. *Mathematical statistics and data analysis*. Nelson Education, 2006.

[16] R Core Team. R: A language and environment for statistical computing. r foundation for statistical computing, vienna, austria. 2013, 2014.

[17] Xiao-Yong Wei, Chong-Wah Ngo, and Yu-Gang Jiang. Selection of concept detectors for video search by ontology-enriched semantic spaces.

IEEE Transactions on Multimedia, 10(6):1085–1096, 2008.

[18] Jun Yang and Alexander G Hauptmann. Exploring temporal consistency for video analysis and retrieval. In *Proceedings of the 8th ACM international workshop on Multimedia information retrieval*, pages 33–42. ACM, 2006.

[19] Mohammed J Zaki. SPADE: An efficient algorithm for mining frequent sequences. *Machine learning*, 42(1-2):31–60, 2001.