Stochastic Models of Video Structure for Program Genre Detection

Cuneyt M. Taskiran, Ilya Pollak, Charles A. Bouman, and Edward J. Delp

School of Electrical and Computer Engineering, Purdue University,
West Lafayette, IN 47907-1285
{taskiran,pollak,bouman,ace}@ecn.purdue.edu

Abstract. In this paper we introduce stochastic models that characterize the structure of typical television program genres. We show how video sequences can be represented using discrete-symbol sequences derived from shot features. We then use these sequences to build HMM and hybrid HMM-SCFG models which are used to automatically classify the sequences into genres. In contrast to previous methods for using SCGFs for video processing, we use unsupervised training without an a priori grammar.

1 Introduction

In this paper we investigate the problem of building stochastic models that characterize the structure of various types of television programs. Our models are based on the idea of segmenting the video sequence into shots and labeling each shot using a discrete-valued label. These labels are then used to build video structure models based on hidden Markov models (HMMs) and stochastic context-free grammars (SCFGs).

We present the application of these models to the task of automatically classifying a given program to one of a specified set of program genres. However, we believe that the video sequence analysis paradigm we have developed will have applicability to a much wider range of video analysis problems, such as video sequence matching and generation of table-of-content views for programs. Furthermore, in addition to being useful in solving these problems, the models themselves can provide us with valuable insight about common characteristics of the sequences within the same program genre.

HMMs and SCFGs have previously been applied to a number of video analysis problems, mainly for gesture recognition and event detection. Most of these techniques use a hand-designed state topology or grammar, which works well for the problem at hand but is hard to generalize to different application domains. For example, it is not obvious how a grammar can be designed for programs like sitcoms or soap operas. We propose an unsupervised approach where the grammar is automatically learned from training data.

The paper is organized as follows: In Section 2 we describe the features extracted from video and how they are used to derive shot labels. Section 3

briefly establishes HMMs, SCFGs, and related terminology. In Section 4 we introduce our model for video analysis which is a hybrid HMM-SCFG model. Finally, in Section 5 we present results of our experiments, and discuss possible further applications of our model in Section 6.

2 Shot Feature Extraction and Labeling

In this section we describe how we generate discrete shot labels which are used in building stochastic models for video sequences, as will be discussed in later sections. Our goal is to derive shot labels that correlate well with the semantic content of the shots and are easily derivable from the compressed video stream with reasonable computational burden.

The first processing step in obtaining shot labels is determining the shot boundaries in the given video sequence. For this paper we have used ground truth shot boundary locations determined by a human operator although robust methods exist [1] to perform this task automatically with high accuracy. After shot boundary locations are determined, a number of features are extracted from each frame in the video sequence. These features are then aggregated to obtain a feature vector for each shot. Finally, clustering is used to derive the shot labels.

2.1 Shot feature extraction

We extract a feature vector from each shot containing features that represent the editing pattern and the motion, color, and texture content of the shot. The distribution of shot lengths is an important indicator of the genre and the tempo of the video program [2] so shot length in frames was chosen as a feature.

The amount of object or camera motion also provides important clues about the semantic content of the shot. Shot length and some measure of average shot activity have been shown to be useful features in classifying movie sequences to different genres [3, 2]. In order to derive the shot motion feature, we first compute the following motion feature for each frame in the sequence

$$\frac{1}{\#\text{blocks with MVs}} \sum_{\text{blocks with MVs}} (MV_x)^2 + (MV_y)^2$$

where MV_x and MV_y are the horizontal and vertical components, respectively, of the motion vector for each macroblock in the frame. The motion feature for the shot is then computed by averaging these values over the length of the shot.

We enhance these two basic shot features by three additional features based on the color and texture of the shot frames. The color features are obtained by averaging the pixel luminance and chrominance values within each frame and over the shot. The texture feature is calculated by averaging the variance of pixel luminance values for each macroblock within each frame and averaging these values for the shot.

At the end of the shot feature extraction process each video sequence is represented by a sequence of shot feature vectors $\{\mathbf{G}_j\}$ (we use $\{\mathbf{G}_j\}$ to denote

random vectors and $\{\mathbf{g}_j\}$ for their realizations) where each shot feature vector $\mathbf{G}_j$ has a dimensionality of $n = 5$.

2.2 Shot feature vector clustering and generation of shot labels

After the shot feature vectors are extracted from shots for all the video sequences in our training data set, they are modelled using a Gaussian mixture model. We use the Expectation-Maximization (EM) algorithm to estimate the parameters of the mixture model and agglomerative clustering to estimate the number of clusters from training data. In this approach the component mixtures are viewed as clusters, and starting with a large number clusters, we merge two clusters at each step until one cluster remains. The number of clusters which maximizes a goodness-of-fit measure is chosen as the final model order.

We collect the shot feature vectors from all video sequences in the training set and number them consecutively, obtaining the collection $\{\mathbf{G}_j\}_{j=1}^{N}$. We assume that the probability density function (pdf), p_k, for each cluster k is multivariate Gaussian with parameters $\boldsymbol{\theta}_k = (\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, where $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ are the the mean vector and the covariance matrix of the cluster, respectively. Then, assuming we have K clusters in the mixture and that the shot feature vectors are iid, we can write the log-likelihood for the whole collection as

$$L(\boldsymbol{\Psi}) = \sum_{i=1}^{N} \log \left(\sum_{k=1}^{K} \pi_k p_k(\mathbf{g}_i;\, \boldsymbol{\theta}_k) \right) \tag{1}$$

where $\boldsymbol{\Psi} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K, \pi_1, \ldots, \pi_{K-1})$ is the complete set of parameters specifying the model and π_k is the probability that $\mathbf{G}_j$ belongs to cluster k, subject to the constraint $\sum_{k=1}^{K} \pi_k = 1$.

We then use a EM-based approach to find a local maximum of the likelihood function to obtain the maximum likelihood estimate (MLE) of the parameter vector, $\hat{\boldsymbol{\Psi}}_{ML}$. Note that in the above formula we assumed that the number of clusters were known, but this number also has to estimated. Unfortunately, the MLE for the number of clusters, $\hat{K}_{ML}$, is not well-defined, since $L(\boldsymbol{\Psi})$ can always be increased by increasing the number of clusters for $\hat{K} \leq N$. Methods for estimating model order generally require the addition of an extra term to the log-likelihood of Equation 1 that penalizes higher order models. We have used the minimum description length (MDL) criterion [4], which is defined as

$$MDL(K, \boldsymbol{\Psi}) = -L(\boldsymbol{\Psi}) + \frac{1}{2} R \log(Nn) \tag{2}$$

where R is the number of real-valued numbers required to specify the parameters of the model and n is the dimensionality of the feature vectors. In our case we have

$$R = K \left(1 + n + \frac{n(n+1)}{2} \right) - 1 \tag{3}$$

and $n = 5$. The minimization of the above criterion is performed iteratively using the EM algorithm. We start with a high number of initial clusters, usually

2-3 times the anticipated number of clusters, and at each step merge the two clusters which cause the maximum decrease in the MDL criterion. This process is continued until only one cluster is left. Then, the number of clusters for which the minimum value of MDL was achieved is chosen as the estimate of the number of clusters for the model, $\hat{K}$ [1].

The mixture model estimated using the above procedure is then used to obtain a discrete label for each shot feature vector. The label for each shot is determined by the cluster number that the shot feature vector is most likely to belong to, that is, given the shot feature vector, $\mathbf{G}_j$, we determine the corresponding shot label symbol, t_j, using

$$t_j = \arg \max_{k \in \{1,\ldots,\hat{K}\}} p_k(\mathbf{g}_j; \boldsymbol{\theta}_k) \tag{4}$$

where the shot label v_j is an integer in the range $\{1, \ldots, \hat{K}\}$.

3 Hidden Markov Models and Stochastic Context-Free Grammars

3.1 Hidden Markov Models

Hidden Markov models (HMMs) have been applied to various video analysis tasks such as classifying programs into genres using audio [5], dialog detection [6], and event detection [7, 8].

A HMM, λ, with N states and M output symbols is a 5-element structure $\langle \mathcal{S}, \mathcal{T}, \mathbf{A}, \mathbf{B}, \boldsymbol{\pi} \rangle$ where $S = \{s_1, \ldots, s_N\}$ is the set of states, $T = \{t_1, \ldots, t_M\}$ is the set of output symbols, $\mathbf{A}$ is the $N \times N$ state transition probability matrix, $\mathbf{B}$ is the $N \times M$ observation symbol probability distribution matrix, and $\boldsymbol{\pi}$ is the $N \times 1$ initial state distribution vector. Once the initial state is chosen using $\boldsymbol{\pi}$, at each value of the discrete time t, the HMM emits a symbol according to the symbol probability distribution in current state, chooses another state according to the state transition probability distribution for the current state, and moves onto that state. The sequence of states that produce the output are not observable and form a Markov chain.

In our approach the observations are discrete-valued shot labels that are derived from each shot in the video sequence using Equation 4. We have used *ergodic* or fully connected HMM topology, for which $a_{ij} > 0$, $\forall i, j$, that is every state can be reached from any other.

Let there be L program genres that we want to use for classification. In order to perform genre detection using HMMs, we train a HMM for each program genre using the shot label sequences for the training video sequences. The standard Baum-Welch algorithm is used in the training [9]. We then use these L HMMs as a standard maximum a posteriori (MAP) classifier and classify a new sequence

[1] The cluster software and further details about the implementation are available at http://www.ece.purdue.edu/~bouman/software/cluster/manual.pdf.

to the genre with the highest a posteriori probability. Assuming all genres are equally likely, the classification of a given video sequence V is performed using the equation

$$\text{genre of } V = \max_{k \in \{1,\ldots,L\}} P(V \mid \lambda_k) \tag{5}$$

where the probability $P(S \mid \lambda_k)$ is obtained using the forward algorithm [9].

3.2 Stochastic Context-Free Grammars

Most video programs have a hierarchical structure where shots may be grouped into scenes and scenes may be grouped into larger segments. Such a hierarchical model for video suggests that shots that are far apart in the program may actually be semantically related. Linear models, such as HMMs, fail to model such long-range dependencies within sequences. Therefore, hierarchical language models such as stochastic context-free grammars (SCFGs) may be more appropriate for modelling video structure. In this section we present a brief introduction to these models, for further details see [10]. SCFGs have been widely used in natural-language processing but have not been used as often as HMMs for video sequence analysis, except for some studies in video event recognition [11, 12].

Suppose we have a sets of symbols, $\mathcal{I} = \{I_1, \ldots, I_N\}$, called nonterminal symbols. We define a production rule to be either a binary or unary mapping of the form

$$I_i \rightarrow I_j I_k \quad \text{or} \quad I_i \rightarrow t_l, \qquad I_i, I_j, I_k \in \mathcal{I},\, t_l \in \mathcal{T} \tag{6}$$

where $\mathcal{T}$ is the set of terminal symbols, which is equivalent to the set of output symbols used in the definition of an HMM. A SCFG, γ, is then specified as a 5-element structure $\langle \mathcal{I}, \mathcal{T}, \mathcal{R}, \mathcal{P}, \boldsymbol{\pi}_{root} \rangle$ where $\mathcal{R}$ is the set of all unary and binary rules of the form given in Equation 6, $\mathcal{P}$ is a set of probabilities associated with each rule in $\mathcal{R}$, and $\boldsymbol{\pi}_{root}$ is the initial probability distribution which determines which nonterminal is chosen as the first state, which is called the root state [2] The rule probabilities in $\mathcal{P}$ are chosen so that they obey the constraints

$$\sum_j \sum_k P(I_i \rightarrow I_j I_k) + \sum_j P(I_i \rightarrow v_j) = 1, \quad i = 1, \ldots, N.$$

After the root nonterminal is chosen using $\boldsymbol{\pi}_{root}$, at each value of the discrete time t, the SCFG chooses one of the rules originating from the current nonterminal and replaces the current node with the symbols on the right side of the rule. This process is continued in a recursive fashion until there are no more nonterminal symbols to be expanded, producing a tree structure which is called a parse tree. Given a string, the probability assigned to it by the SCFG is the

[2] The type of SCFG defined here is actually based on a special case of context-free grammars called the Chomsky normal form. However, there is no loss of generality since it can be shown that any SCFG can be transformed into an identical grammar in the Chomsky normal form in the sense that the languages produced by the two grammars will be identical.

sum of the probabilities for all the parse trees that could have produced the given string.

One problem with SCFGs is that, compared with linear models like HMMs, their training is slow. For each training sequence each iteration takes $O(N^3|V|^3)$ computations, where N is the number of nonterminals in the grammar and $|V|$ is the number of shots for the video sequence [13]. This makes training for longer video programs impractical and makes using a MAP-based approach similar to the one used for HMMs hard. In the next section we discuss our hybrid HMM-SCFG approach which solves this problem.

4 The Hybrid HMM-SCFG Approach

In order to be able to train genre SCFGs in reasonable time, we propose a hybrid approach. In this approach we train SCFGs for genres as follows: Let there be L genres that we want to classify sequences into. We first train a HMM for each genre using the sequences in the training set, thereby obtaining L HMMs, $\lambda_1, \ldots, \lambda_L$. We then divide all the sequences in the training set into 10 pieces, $\mathbf{x}_j, j = 1, \ldots, 10$. This is done in order alleviate the problem of the sequences being nonstationary over long intervals. For each sequence, we run each of the L HMMs on each of the 10 pieces and obtain the log-likelihood value $\log P(\mathbf{x}_j \mid \lambda_l)$ for each piece which are then arranged in a $L \times 10$ matrix of log-likelihood values. t the end of this step each shot label sequence in the training set is represented as a matrix of log-likelihood values obtained using HMMs.

Instead of training SCFGs directly on the shot label sequences, we use the log-likelihood matrices obtained from the above step. In this way, the computation is reduced from $O(N^3|V|^3)$ computations to $O(N^3 10^3)$ computations which brings about significant savings in training time, since usually we have $|V|^3 \gg 10^3$. In order perform the grammar training in our approach, we introduce a new type of nonterminal denoted by $\tilde{I}^l$, which can only appear on the right side of a rule, and change the form of the unary rules defined in Equation 6 to $P(I^j \rightarrow \tilde{I}^l)$. The special nonterminal $\tilde{I}^l$ takes on values in the range $[1, L]$ and indicates the particular HMM whose log-likelihood value will to be used to for that piece. This implies that instead of the rule probability $P(I^j \rightarrow t_l)$ we have the probability $\sum_l P(I^j \rightarrow \tilde{I}^l) P(\tilde{I}^l \rightarrow \mathbf{x}_k)$. The probabilities $P(\tilde{I}^l \rightarrow \mathbf{x}_k) = P(\mathbf{x}_k \mid \lambda_l)$ are obtained from the HMM log-likelihood matrices, whereas the probabilities $P(I^j \rightarrow \tilde{I}^l)$ have to be estimated along with binary rule probabilities. We have modified the standard SCFG training algorithm, called the inside-outside algorithm [10], so that these probabilities can be estimated from the input HMM log-likelihood matrices.

5 Experimental Results

We selected four program genres,soap operas, sitcoms, C-SPAN programs, and sports programs for our experiments, and selected a total of 23 video sequences from our video database that we believe represented the given genres. These

sequences were digitized at a rate of 2 Mbits/sec in SIF (352×240) format. Commercials and credits in the sequences, if they exist, were edited out. The locations and types of all the shot transitions in these sequences were recorded by a human operator. Detailed information about the sequences are given in Table 1. All the sequences in the *soap* and *comedy* genres contain complete programs, some of the sequences for other genres contain only parts of programs.

Table 1. Statistical information about the sequences used in the experiments.

genre	# sequences	avg length (minutes)	avg number of shots/seq
soap	11	14.3	140.1
comedy	11	20.2	264.4
cspan	14	28.1	59.5
sports	14	12.3	84.1

The sequences in each genre were divided into sets containing roughly the same number of sequences. One of these sets were used as the training set, the other as the test set for the algorithms. We clustered the shot feature vectors obtained from the sequences in the training set, using the method described in Section 2.2. The cluster parameters so obtained were then used to label the shots of the sequences in both the training and test sets. We used six clusters, so the number of terminal symbols, $M = 6$.

We performed two genre classification experiments. In Experiment I a HMM for each genre was trained using the the training set and then used these HMMs to classify the sequences in the test set where the genre of each sequence was determined using Equation 5. The number of states of each HMM was set to four. All the sequences in the training set were correctly classified. The results for the test set are given in Table 2.

In Experiment II we used the same HMMs that were used for Experiment I but we now used the hybrid SCFG-HMM model that was described in Section 4. The number of terminal nodes of the SCFG was set to four. Again, all the sequences in the training set were correctly classified. The results for the test set are shown in Table 3.

6 Conclusions

In this paper we have examined the problem of unsupervised training of stochastic models that characterize the structure of typical television program genres. We showed how the computational complexity of training a SCFG may be greatly reduced using a hybrid HMM-SCFG model and compared the results obtained with this model and HMMs for the program genre classification task. For this task, our model gave slightly better results than HMMs.

Table 2. HMM genre classification confusion matrix. HMMs of order 6 were used.

	Classifier Output			
True Label	soap	comedy	cspan	sports
soap	4	1	0	0
comedy	0	5	0	0
cspan	0	1	6	0
sports	0	0	0	6

Table 3. SCFG-HMM genre classification confusion matrix. The same HMMs as the ones provided the results in Table 2 were used with a SCFG of 4 nonterminal nodes.

	Classifier Output			
True Label	soap	comedy	cspan	sports
soap	5	0	0	0
comedy	0	5	0	0
cspan	1	0	6	0
sports	0	0	0	6

As pointed in the introduction, the applicability of the shot label sequence representation and our hybrid HMM-SCFG stochastic model go far beyond the genre classification problem. The shot label sequences may be used to very efficiently search video databases for sequences similar to a query sequence. This may be done using dynamic programming based on the sequence edit distance or by using profile HMMs, such as the ones used for searching biological sequence databases.

References

1. Taskiran, C., Bouman, C., Delp, E.J.: The ViBE video database system: An update and further studies. In: Proceedings of the SPIE/IS&T Conference on Storage and Retrieval for Media Databases 2000, San Jose, CA (2000) 199–207
2. Adams, B., Dorai, C., Venkatesh, S.: Study of shot length and motion as contributing factors to movie tempo. In: Proceedings of the ACM International Conference on Multimedia, Los Angeles, CA (2000) 353–355
3. Vasconcelos, N., Lippman, A.: Statistical models of video structure for content analysis and characterization. IEEE Transactions in Image Processing **9** (2000) 3–19
4. Rissanen, J.: A universal prior for integers and estimation by minimum description length. The Annals of Statistics **11** (1983) 417–431
5. Liu, Z., Huang, J., Wang, Y.: Classification of TV programs based on audio information using hidden Markov model. In: IEEE Second Workshop on Multimedia Signal Processing, Redondo Beach, CA (1998) 27–32
6. Alatan, A.A., Akansu, A.N., Wolf, W.: Multi-modal dialog scene detection using hidden Markov models for content-based multimedia indexing. Multimedia Tools and Applications **14** (2001) 137–151

7. Brand, M., Kettnaker, V.: Discovery and segmentation of activities in video. IEEE Transactions on Pattern Analysis and Machine Intelligence **22** (2000) 844–851
8. Xie, L., Chang, S.F., Divakaran, A., Sun, H.: Structure analysis of soccer video with hidden Markov models. In: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Orlando, Fl (2002)
9. Rabiner, L.R.: A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE **77** (1989) 257–285
10. Manning, C.D., Schutze, H.: Foundations of Statistical Natural Language Processing. MIT Press, Cambridge, MA (1999)
11. Ivanov, Y.A., Bobick, A.: Recogition of visual activities and interactions by stochastic parsing. IEEE Transactions on Pattern Analysis and Machine Intelligence **22** (2000) 852–872
12. Moore, D., Essa, I.: Recognizing multitasked activities from video using stochastic context-free grammar. In: Workshop on Models versus Exemplars in Computer Vision in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii (2001)
13. Lari, K., Young, S.J.: The estimation of stochastic context-free grammars using the inside-outside algorithm. Computer Speech and Language **4** (1990) 35–56