Hidden Markov Model Parsing of Video Programs

Wayne Wolf
Department of Electrical Engineering
Princeton University

1 Introduction

This paper introduces statistical parsing of video programs using hidden Markov models (HMMs). The fundamental units of a video program are shots and transitions (fades, dissolves, etc.). Those units are in turn used to create more complex structures, such as scenes. Parsing a video allows us to recognize higher-level story abstractions—dialog sequences, transitional scenes, etc. These higher-level story elements can be used to create summarizations of the programs, to recognize the most important parts of a program, and many other purposes. Since the cinema is often referred to as a visual language[1], it is natural to use the statistical parsing methods developed for speech recognition[2] to help us understand visual languages.

There are two important reasons to use statistical parsing methods to understand video programs. First, there is some uncertainty in the classification of shots, similar to the uncertainty in classifying phonemes in speech. Second and most important, cinema, like any natural language has enough variability in its usage that ironclad rules cannot effectively capture the intent of a sequence of images. For example, a simple dialog scene would have alternating shots of two characters speaking. However, a director may choose to include a shot of something happening in the background, a flashback shot of an event remembered by one of the speakers, or some other variation on the basic sequence. The hidden Markov model allows us to recognize common storytelling structures in spite of the inevitable variations in their presentation.

In our experience, non-statistical parsing such as LR(k) grammars and NFA are not flexible enough to parse video programs into useful syntactic structures. Jain et al. developed a deterministic parser for news programs. Their parsing scheme identified only very basic structures, such as news stories, and did not identify any of the structure within the program. Their lexical analysis also identified features which were very specific to a particular program. Yeung et al.[4] developed a scene transition graph model. Clustering, which is used to create the nodes of a scene transition graph, does not provide the lexical information required to recognize many cinematic structures. The deterministic graph algorithms used to detect story structure also suffer problems recognizing the structure of complex video programs. Brooks[5] described a methodology for computational narrative which emphasizes the design and presentation of nonlinear stories, but this architecture does not directly

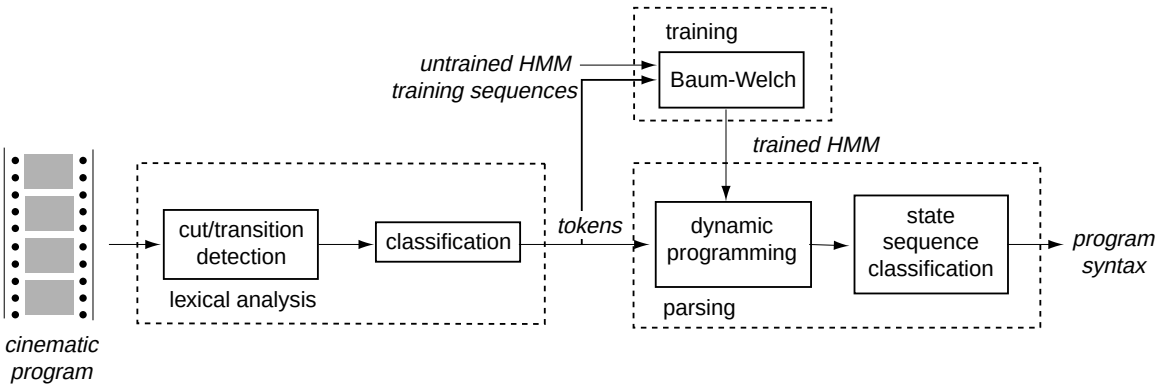

FIGURE 1. Lexical analysis and parsing of video by HMMs.

address the problem of algorithms for exacting structure from filmed stories. Zhang et al.[3] developed a hierarchical browser, which temporally samples the video and does not provide insight into the story structure of the program.

Figure 1 outlines the lexical analysis and parsing process for a cinematic program transferred to video. Lexical analysis selects key frames and identifies transitions, then classifies those units to create a stream of tokens which represent the program. Given a trained HMM, dynamic programming can select the most likely state sequence to have generated that sequence of tokens. The state sequence is classified to identify syntactic units in the program. The HMM is trained using a stream of tokens which represent training programs selected for their representation of important syntactic structures. We have implemented an HMM parsing system in C++. We have separately experimented with subject-based lexical analysis, as described below, but we have so far concentrated on HMM experiments using hand-coded token sequences.

2 Lexical Analysis

Tokenization of the video program into lexical units involves dividing the video into shots and transitions, selecting key frames to represent the shots, and classifying the key frames and transitions. A variety of algorithms can be used to detect cuts and transitions in videos (if the information is not available directly as an edit decision list), as well as selecting one or more key frames for each shot.

Classification of key frames and transitions is a critical step, since an inappropriate tokenization of the video makes it difficult to identify larger syntactical units. We use a classification scheme which is based on the terms used by directors and cinematographers to classify shots. The classification of transitions is directly computed by the transition detection algorithm, which recognizes a sequence of frames as a dissolve, fade, wipe, etc., which are all commonly used terms in cinema. We classify shots based upon the framing of their main subject. Since people are the subjects of many shots of most video programs, we can use face detection algorithms[6,7] to detect the persons in a shot. We then classify the shot based upon the size in the frame of the largest face. *Master shot* (a shot of all the characters in a scene), *medium shot*, and *close-up* are classifications of shots commonly made by filmmakers; all these classifications can be performed by this face detection and measurement scheme. *Establishing shots* which show the locale at which action takes place are also common features; detection of buildings and outdoor scenery can be used to help identify establishing shots.

3 Training the Parser

As in speech recognition, we must train the parser based on input sequences. We construct an initial HMM which is designed to recognize canonical versions of cinematic sequences of interest: establishing-master shot sequences, dialog sequences, etc. We currently use the Baum-Welch algorithm for training, though other training algorithms are also possible. Training starts with simplified HMMs constructed by hand. These machines can be constructed by using simplified, textbook definitions of video sequences. States in the HMM identify elements of a sequence. A state may correspond directly to lexical units if those units have unique purposes in a sequence; it may also correspond to a positional use of a lexical unit in a sequence. Training sequences are selected which exhibit instances of the sequences which the HMM is designed to be recognized. Training updates the HMM transition and output probabilities to reflect the structure of the training sequences. The trained HMMs can then be used to parse video sequences.

4 Parsing Video Sequences

The result of lexical analysis is a sequence of tokens which represent the program. Parsing extracts larger syntactic units from the sequence. The grammar is represented as an HMM obtained from the training phase. We use dynamic programming to determine the most likely sequence of states which generated the input token sequence. A separate classification step identifies the syntactic unit which corresponds to that state sequence—a dialog sequence, for example.

Figure 1 shows a HMM for a dialog sequence: state 0 represents the establishing shot; state 1 represents the dialog, which is primarily medium shots but may include close-ups; and state 2 represents the master shot of the scene, which is primarily long shots. The diagram shows the transition probabilities and the output probabilities for the three types of shots used: master (x), medium (m), and close-up (c). To illustrate the recognition process, we hand-coded the shots in one scene of a sketch from “The Tracey Ullman Show.” This scene portrayed a dialog between two characters during a press conference. The resulting sequence was xmmcmcmcmxmcmcmcmxmxmcmcmx, which represents dialog between the characters, close-ups to show reactions, and master shots including the reporters at the conference. The most likely state trajectory through the HMM as computed by dynamic programming was 0111111111211111112121111121. The long sequences of state 1s identify dialog sequences in the scene; the results match up well with the intent of the scene. The syntactic units identified by parsing can then be used to select one

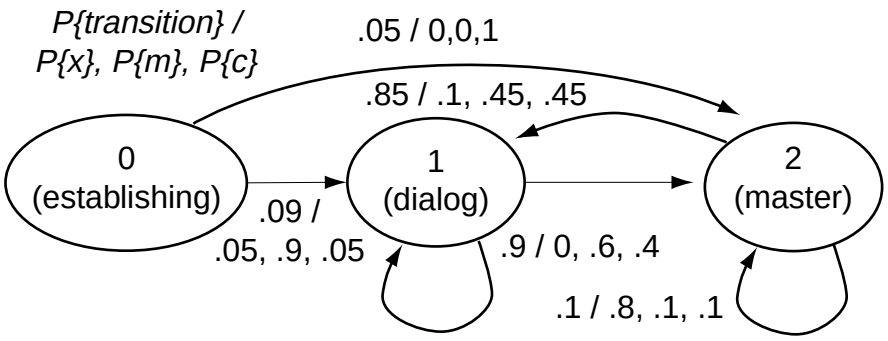

FIGURE 1. A HMM for a dialog sequence.

key frame which best identifies the syntactic unit, or for a variety of other purposes.

Acknowledgments

This work was supported by the ARPA CAETI Program under a contract monitored by NRAD.

References

[1] David Bordwell, *Narration in the Fiction Film*, University of Wisconsin Press, 1985.

[2] Frederick Jelinek, "Continuous speech recognition by statistical methods," *Proc. IEEE*, 64(4), April, 1976, pp. 532-556.

[3] HongJiang Zhang, Stephen W. Smoliar, and Jian Hua Wu, "Content-based video browsing tools," in *Proceedings, SPIE Conference on Multimedia Computing and Networking*, SPIE, vol. 2417, pp. 389-398, 1995.

[4] Minerva Yeung, Boon-Lock Yeo, Wayne Wolf, and Bede Liu, "Video browsing using clustering and scene transitions on compressed sequences," in *Proceedings, SPIE Conference on Multimedia Computing and Networking*, SPIE, vol. 2417, pp. 399-414, 1995.

[5] Kevin M. Brooks, "Do story agents use rocking chairs? The theory and implementation of one model of computational narrative," in *Proceedings, ACM Multimedia '96*, ACM Press, 1996, pp. 317-328.

[6] S.-H. Lin, S. Y. Kung, and L.-J. Lin, "Face recognition/ detection by probabilistic decision-based neural network," to appear in *IEEE Trans. Neural Networks*.

[7] H. A. Rowley, S. Baluja, and T. Kanade, "Human face detection in visual scenes," Carnegie Mellon Technical Report CMU-CS-95-158R, 1995.