Learning Video Preferences from Video Content

Darin Brezeale
Department of Computer Science and Engineering
The University of Texas at Arlington
Box 19015, Arlington, TX 76019, USA
darin.brezeale@uta.edu

Diane J. Cook
School of Electrical Engineering and Computer Science
Washington State University
Pullman, WA 99164-2752, USA
cook@eecs.wsu.edu

ABSTRACT

Viewers of video now have more choices than ever. As the number of choices increases, the task of searching through these choices to locate video of interest is becoming more difficult. Current methods for learning a viewer's preferences in order to automate the search process rely either on video having content descriptions or on having been rated by other viewers identified as being similar. However, much video exists that does not meet these requirements. To address this need, we use hidden Markov models to learn the preferences of a viewer by combining visual features and closed captions. Results are provided from some initial experiments using this approach.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing; I.2.6 [**Artificial Intelligence**]: Learning

Keywords

video preferences, user modeling, closed captions

1. INTRODUCTION

People today have access to more video than at any time in history. Sources of video include television broadcasts, movie theaters, movie rentals, video databases, and the Internet. While many video choices come from the entertainment domain, other types of video are becoming more common, such as educational lectures at universities and conferences [34].

As the number of video choices increases, the task of searching for video of interest is becoming more difficult. One approach that viewers take is to search for video within specific genre. In the case of entertainment video, the genre of the video is provided when the video is released. However, there is much video that is unclassified. This has led to research in automatically classifying video by genre. While knowing the genre of video is helpful, the large amounts of video choices within many genre still make finding video of interest a time-consuming process. In addition, this problem is even greater for people who enjoy video from a variety of genre, which seems likely for most people. For these reasons, systems have been developed that can learn a particular person's preferences and make recommendations given these preferences.

There have traditionally been two approaches to identifying video of interest to a viewer. The first is the case-based approach, which utilizes descriptions of the video content. In the case of entertainment video, the description might include the genre of the video, director, actors, and a brief summary of the video. The second is collaborative filtering, which attempts to identify viewers that are considered similar by some measure. Recommendations for the current viewer will be drawn from the positively rated video of these similar viewers.

The major strength of the case-based approach is that it relies strictly on the viewer's profile. Once a viewer's preferences are known, it is a simple task to match these up with video content descriptions. There are, however, several weaknesses to the case-based approach. One is that it takes some effort to produce content descriptions. While this is typically not a problem when dealing with entertainment video such as television or movies, there is much video in video databases and on the Internet for which there are no content descriptions. Another weakness is that the viewer must initially seed the system with some preference information. A viewer may not wish to devote the time and effort to provide enough preference details for the system to perform well. The third weakness is that recommendations will be very similar to previously rated video.

Collaborative filtering does not require the content descriptions used by the case-based approach. Also, unlike the case-based approach, video recommendations are not restricted to video similar to that previously rated by the user if the group the viewer is assigned to has a greater variety of interests. However, it does take some effort to gather enough information about other viewers in order to determine who is similar to the current viewer. A second weakness of collaborative filtering is the latency of a new video spreading. A video can't be recommended if no one has seen and rated it yet.

There are many videos that lack the content descriptions required by the case-based approach and do not have the ratings of other viewers required by collaborative filtering. The approach we have chosen is to extract visual features and

MDM/KDD'07, August 12, 2007, San Jose, California, USA.

closed captions from video in order to learn a viewer's preferences. The visual features and closed captions are combined to produce observation symbols for training hidden Markov models (HMM). A video is a collection of features in which the order that the features appear is important, which suggests that an HMM might be appropriate for classification. We believe that visual features and closed captions are complementary. Visual features represent what is being seen, but miss much of the social interaction. Video dialogue typically doesn't describe what is being seen, but represents the social interaction.

The rest of the paper is organized as follows. Our approach is to learn video preferences using methods more commonly associated with classification of video by genre, so in Section 2 we discuss related work from the video classification field as well as other approaches to video recommendation. Section 3 provides background on the features, clustering, and classification methods necessary to understand our approach. In Section 4 we explain our overall methodology. Section 5 provides details of our initial experiments. Conclusions and suggestions for future work are provided in Section 6.

2. RELATED WORK

Research related to this work falls into two categories. The first is the previously mentioned video recommendation systems. The second is automatic classification of video by genre.

2.1 Video Recommendation

Ardissono et al. [1] combine three user models. The Explicit user model is constructed from a form the viewer fills out requesting demographic information, general interest in topics such as books and politics, and TV program preferences. The Stereotypical user model uses information obtained in constructing the Explicit user model, which in turn is used to determine how well the viewer matches a number of pre-existing categories, or stereotypes, of TV viewers. Finally, the Dynamic user model is constructed by observing the viewer's TV viewing habits. In particular, the day and time of TV programs is monitored as well as the types of TV programs watched. The preferences derived from these three user models are combined using a weighted sum. A precision of 0.8 and a mean absolute error rate of 0.3 were achieved.

The fusion of these three user models has several strengths. The input of the user used to construct the Explicit user model allows the system to begin making recommendations immediately as well as to match the user to existing viewer profiles (the Stereotypical user model). The Dynamic user model allows the system to learn new preferences over time. However, this approach has several weaknesses as well. The user may not wish to spend the time necessary to provide the initial preference information. Without much initial preference values, the Explicit and Stereotypical user models will be limited in their usefulness. The Dynamic user model allows the system to learn over time, but it considers the time and day of viewing as important features. We hypothesize that digital video recorders, which make recording television programs for later viewing easy, will reduce the importance of time and day as features since a viewer's choices will not be limited to what is on at a certain time of the day.

Basu et al. [3] identify the content features associated with specific genre of movies that a user liked and then perform classification using inductive learning. This approach achieved precision and recall values of 83% and 34%, respectively.

The strengths of the case-based and collaborative filtering approaches tend to offset the weaknesses of the other. As a result, some authors have combined the two approaches to recommendation. Smyth and Cotter [32] require that a user initially provide information about the types of television programs that they like and dislike. This information is used to find viewers with similar interests. Some recommendations are derived from the content information provided at registration while other recommendations are based on the preferences of the similar viewers. Smyth and Cotter measured performance by the percentage of users who received N or more good recommendations per day, where $N = 1, 2, 3$. The collaborative filtering recommendations produced one or more good recommendations per day for 96% of users while the case-based recommendations produced one or more good recommendations for 78% of users.

Brezeale and Cook [6] learn video preferences using closed captions and discrete cosine transform (DCT) coefficients separately. All of the closed captions from each of 81 movies were used. After representing the closed captions using the bag-of-words model, classification was performed using a support vector machine. To learn preferences using visual features, video clips in the MPEG-1 format were segmented into shots using color histograms in the RGB color space, after which each shot was represented by a keyframe of the DC terms of the DCT coefficients. The keyframes were clustered using a k-means algorithm so that similar shots were grouped together; this allowed each video clip to be represented as a feature vector whose elements are a count of how many of each type of shot are present in the video clip. Classification was performed using a support vector machine. The classification accuracy using closed captions was 64% while the classification accuracy using DCT coefficients and a cluster size of 20 was 59%.

The approach described in this paper is most beneficial in situations in which neither case-based nor collaborative filtering approaches are applicable and the only choice is to analyze the video itself. It does not require that a viewer provide any information about his preferences other than a rating for a video that he has viewed. This saves time as well as avoids poor recommendations that might occur due to omissions in the preference description. It is also unnecessary to identify similar viewers.

Our approach does have some known disadvantages. One is that some video may not have closed captions nor may it be possible to automatically generate a transcript using speech recognition. Another is that initially the system would ignorant of the viewer's preferences and would require that he locate enough video of interest to learn preferences. This last limitation could be overcome by combining our approach with case-based or collaborative filtering approaches when the features required by those approaches are available.

2.2 Classification of Video by Genre

Approaches to classification of video by genre use features from three modalities: audio, visual, or text. We only discuss approaches that used visual or text features because of their relationship to this work.

Zhu et al. [35] classify news stories using features obtained from closed captions. News video is segmented into stories using the topic change marks (explained in Section 3) inserted by the closed caption annotator. A natural language parser is used to identify keywords within a news segment and the first 20 unique keywords are kept. A weighted voting scheme involving the conditional probabilities of classes and keywords is used to classify the news segment.

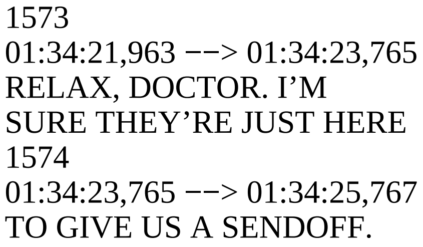
Lin and Hauptmann [21] combine classifiers of visual and text features. A video is divided into shots and a keyframe is extracted from each. Each keyframe is represented by a vector of the color histogram values in the *RGB* color space. A support vector machine (SVM) is trained on these features. For each shot, the closed captions are extracted and represented as a vector. For these vectors, another SVM is trained. Two methods for combining classifiers are investigated. The first method is based on Bayes' theorem and uses the product of the posterior probabilities of all classifiers. Performance is improved by assuming equal prior probabilities. The second method uses an SVM as a meta-classifier for combining the results of the other two SVMs. Both methods had similar recall, but the SVM meta-classifier had statistically significant higher precision.

Hidden Markov models are a popular method for classifying video by genre. The typical approach is to train one HMM for each class.

Dimitrova et al. [8] detect and track faces and text. Counts of the number of faces and text are used for labeling each frame of a video clip. An HMM is trained for each class using the frame labels as the observations.

Lu et al. [23] classify a video by first summarizing it. The color channel bands of each frame are normalized and then moved into a chromaticity color space. After more processing including both wavelet and discrete cosine transforms, each frame is now in the same lighting conditions [9]. A set of twelve basis vectors determined from training data can now be used to represent each frame. A hierarchical clustering algorithm segments the video into scenes; the keyframes from the scenes represent the summarized video. One HMM is trained for each video genre with the keyframes as the observation symbols.

Huang et al. [17] combine audio and visual features. The audio features produced are as described in [22]. The visual features are dominant color, dominant motion vectors, and the mean and variance of the motion vectors. Four ways of using these features are investigated. In the first method, the audio and visual features are determined for each clip and concatenated into a single vector. The features vectors for sequences of 20 clips are the input to HMMs, one for each video class. In the second method, audio, color, and motion features are produced for each video frame and a separate HMM is trained for each. The product of the observation probabilities for each these three types of features is used for classification. The third method uses two stages of HMMs. In the first stage, audio features are used to train HMMs for distinguishing between commercials, football or basketball games, and news reports or weather forecasts. In the second stage, visual features are used to train HMMs to distinguish football games from basketball games and news reports from weather forecasts. For the fourth method, for each of the three types of features (audio, color, motion), an HMM is trained for each class. The output from these HMMs becomes the input to a three layer perceptron neural network. The product HMM gave the best average classification accuracy.



```
1573
01:34:21,963 --> 01:34:23,765
RELAX, DOCTOR. I'M
SURE THEY'RE JUST HERE
1574
01:34:23,765 --> 01:34:25,767
TO GIVE US A SENDOFF.
```

Figure 1: Example of two closed captions sets from Star Trek: Close Contact

Gibert et al. [15] use motion and color to classify sports video. Motion vectors from MPEG video clips are used to assign a motion direction symbol to each video frame. Color symbols are assigned to each pixel of each frame. A symbol for the most prevalent color is assigned to the entire frame. Unlike most other applications of HMMs for video classification, the authors train two HMMs for each video class: one for the frame color symbols and the other for the motion direction symbols. The output probability for each class is calculated by taking the product of the color and motion output probabilities for that class.

3. BACKGROUND

3.1 Closed Captions

Closed captioning is a method of letting hearing-impaired people know what is being said in a video by displaying text of the speech on the screen. Closed captions are found in Line 21 of the vertical blanking interval of a television transmission and require a decoder to be seen on a television [30]. On a DVD the closed captions are stored in sets with display times. Figure 1 shows the 1573^{rd} and 1574^{th} closed captions sets for the movie *Star Trek: Close Contact*.

While not all television shows have closed captions, that is changing. The Telecommunications Act of 1996, which took effect in 1998, placed closed captioning requirements on television shows broadcast in the United States. With some exceptions, the law required that broadcasters begin providing closed captions on their broadcasts with a goal of 100% of all broadcast hours of new (first broadcast in 1998 or later) television shows by 2006 and 75% of older (first broadcast prior to 1998) television shows by 2008.

In addition to representing the dialog occurring in the video, closed captioning also displays information about other types of sounds such as onomatopoeias (e.g., grrrr), sound effects (e.g., [BEAR GROWLS]), and music lyrics (enclosed in music note symbols, ♪). At times, the closed captions may also include the marks >> to indicate a change of speaker or >>> to indicate a change of topic [14].

One advantage of text-based approaches is that they can utilize the large body of research conducted on document text classification [31]. Another advantage is that the relationship between the features (i.e., words) and specific genre is easy for humans to understand. For example, few people would be surprised to find the words 'stadium', 'umpire', and 'shortstop' in a transcript from a baseball game.

However, using closed captions does have some disadvantages. One is that the text available in closed captions is largely dialog; there is little need to describe what is being seen. For this reason closed captions do not capture much of what is occurring in a video. A second is that not all video has closed captions nor can closed captions be generated for video without dialog. A third is that while extracting closed captions is not computationally expensive, generating the feature vectors of terms and learning from them can be computationally expensive since the feature vectors can have tens of thousands of terms.

A common method for representing text features is to construct a feature vector using the bag-of-words model [12]. In the bag-of-words model, each feature vector has a dimensionality equal to the number of unique words present in all sample documents (or closed caption transcripts) with each term in the vector representing one of those words. Each term in a feature vector for a document will have a value equal to the number of times the word represented by that term appears in the document. One potential drawback of the bag-of-words model is that information about word order is not kept.

Representing a transcript may require a feature vector with dimensions in the tens of thousands if every unique word is included. To reduce the dimensionality, stop lists and stemming are often applied prior to constructing a term feature vector. A stop list is a set of common words such as ‘and’ and ‘the’ [13]. Such words are unlikely to have much distinguishing power and are therefore removed from the master list of words prior to constructing the term feature vectors. Stemming removes the suffixes from words leaving the root. For example, the words ‘independence’ and ‘independent’ both have ‘indepen’ as their root. The stemmed words are used to generate the feature vectors instead of the original words. One of the more common methods for stemming is using Porter’s stemming algorithm [27].

3.2 Visual Features

A variety of features can be obtained from the visual part of a video, as demonstrated by the video retrieval and classification fields [2], [4]. Some choices of features are color, texture, objects, and motion. We will focus on color-based features because of their relevance to this work.

Many methods for representing a video extract features on a per frame or per shot basis. A video is a collection of images known as frames. All of the frames within a single camera action are called a shot. To reduce the amount of information that must be worked with, a shot is often represented by a single frame known as the keyframe. The task of automatically detecting shots is difficult, in part because of the various ways of making transitions from one shot to the next. Lienhart [20] states that some video editing systems provide more than 100 different types of edits and no current method can correctly identify all types.

A video frame is composed of a set of dots known as pixels and the color of each pixel is represented by a set of values from a color space [28]. Many color spaces exist for representing the colors in a frame. Two of the most popular are the red-green-blue (RGB) and hue-saturation-value (HSV) color spaces. In the RGB color space, the color of each pixel is represented by some combination of the individual colors red, green and blue. In the HSV color space, colors are represented by hue (i.e., the wavelength of the color percept), saturation (i.e., the amount of white light present in the color), and value (also known as the brightness, value is the intensity of the color) [4].

The distribution of colors in a video frame is often represented using a color histogram, that is, a count of how many pixels in the frame exist for each possible color. Color histograms are often used for comparing two frames with the assumption that similar frames will have similar counts even though object motion or camera motion will mean that they don’t match on a per pixel basis. One disadvantage of color histograms is that spatial information is lost.

A disadvantage of color-based features is that the images represented in frames may have been produced under different lighting conditions and therefore comparisons of frames may not be correct. The solution proposed by Drew and Au [9] is to normalize the color channel bands of each frame and then move them into a chromaticity color space.

One difficulty in using visual features is the huge amount of potential data. This problem can be alleviated by using keyframes to represent shots or with dimensionality reduction techniques, such as the application of wavelet transforms.

3.3 Hierarchical Clustering

Hierarchical clustering methods are of two types: agglomerative or divisive. Agglomerative clustering begins with all individuals separate. The two nearest individuals are joined into a group. Then the next two nearest individuals (or individual and group) are joined until all individuals have been joined to a group. Divisive clustering begins with a single group and divides it into smaller groups until eventually each member has been separated out. In both cases ultimately a tree, often represented with a dendrogram, is formed which makes hierarchical clustering particularly popular for exploring relationships in the data [24].

There are also several ways of calculating the similarity of two groups, or a group and an individual (a group with one member) [10]. The more common ways are single linkage, complete linkage, and average linkage. The single linkage method determines the similarity of two groups by calculating the distance between the nearest members of each group. The complete linkage method determines the similarity of two groups by calculating the distance between the farthest members of each group and then choosing the minimum of these. The similarity of two groups when average linkage is used is determined by calculating the average distance from each member of one group to each member of the other group. Average linkage is more robust to outliers [10] than single or complete linkage. Assigning individuals to a group requires a distance calculation where the distance between two individuals can be any of the standard distance measures, such as the Euclidean distance.

An advantage of hierarchical clustering over some other forms of clustering, such as k-means, is that it is unnecessary to know the number of clusters in advance. However, it is still necessary to decide where to prune the dendrogram in order to determine the clusters. A disadvantage of hierarchical clustering is that once an individual has been merged into a group (agglomerative method) or separated from a group (divisive method), it can not be undone. Another disadvantage is that the data may not fit a hierarchy.

3.4 Hidden Markov Models

The hidden Markov model (HMM) is widely used for classifying sequential data. An HMM represents a set of states

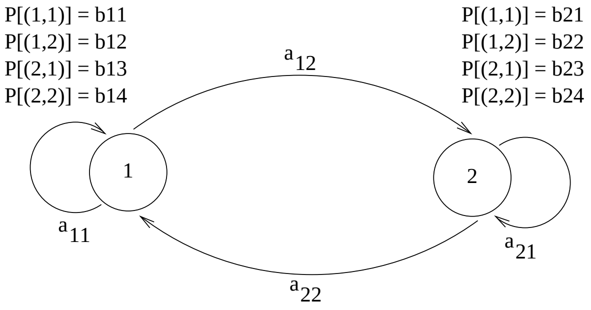


Figure 2: Example of hidden Markov model

and the probabilities of making a transition from one state to another state [29]. While in each state, an observation symbol can be generated with some probability. More specifically, an HMM is represented by $Q = 1, \dots, N$ states, each generating $V = 1, \dots, M$ observation symbols, an $N \times N$ matrix A of transition probabilities where a_{ij} is the probability of moving to state j while in state i, an $N \times M$ matrix B of observation (or emission) probabilities where b_{ik} is the probability of generating symbol v_k while in state i, and a $1, \dots, N$ vector π of starting probabilities where i is the probability of beginning in state q_i.

The model is 'hidden' because the true number of states and which state the model is in are unknown; only the observation symbols being generated are known with certainty. Figure 2 shows an example of a two-state HMM with four observation symbols $\{(1,1),(1,2),(2,1),(2,2)\}$ and the probabilities b_{ik} that each will be generated.

4. METHODOLOGY

Our proposed methodology begins by extracting the closed captions sets from the training videos that the viewer has rated with the intent of using the times that the closed captions sets are displayed as the mechanism for segmenting the video. This allows us to avoid the error-prone task of automatically detecting shot boundaries.

For each closed captions set, we extract the first video frame to represent the entire time span that the closed captions set is displayed. Visual features are extracted from this video frame.

Some methods for representing text or images suffer from a lack of context. The bag-of-words model, which is a common method for representing documents, does not maintain word order and as a result two documents with essentially the same words but different word order can have different meanings but appear similar when comparing their term-feature representations. Likewise, two different images may appear similar when represented as color histograms. By combining text and visual features, we believe that these limitations can be lessened.

The process for combining the visual features and closed captions begins by clustering the visual features and closed captions separately. The cluster assignment of a closed captions set and the cluster assignment of it corresponding video frame are combined in the form (closed captions set cluster number, video frame cluster number), which becomes an observation symbol for training the HMM An example is shown in Figure 3. In this example, the hierarchy produced from the closed captions sets has been pruned at a level that produces two clusters, numbered from left to right as 1 and 2. The video frame hierarchy has been pruned at a level that produces three clusters, numbered from left to right as 1, 2, and 3. Closed captions set #1 (the first set chronologically) is in cluster 1 while its corresponding frame is in cluster 2, so the observation symbol will be (1,2). Closed captions set #2 (the second set chronologically) is in cluster 1 while its corresponding frame is cluster 1, so it is represented by the observation symbol (1,1). This is repeated for all of the closed captions sets and frames extracted from a video clip to generate a sequence of observation symbols that represent a video.

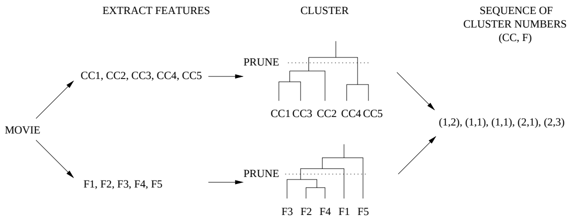


Figure 3: Example of observation symbol production

The sequence of symbols for each video rated as 'liked' by the viewer are used to train an HMM. This entire process is repeated for those movies rated by the viewer as 'disliked'. Test video sequences are classified according to the HMM that has the highest probability of generating the sequence.

5. EXPERIMENTS

In order to validate our approach, we chose to obtain real-world data. For this purpose, we used the MovieLens collaborative filtering data set as a starting point [16]. This data set was produced by 6,040 viewers who had rated movies from a set of 3,883 possible movies for a total of more than 1 million ratings. The demographic information provided of the users is gender, age (grouped into 7 age ranges), occupation (consisting of 21 choices), and zip code. The movie data includes the title and one or more genre. We chose to use video from the entertainment domain due to the fact that it has the closed captions required by our approach and the availability of ratings. We acquired the DVD version of 90 of the movies represented in the MovieLens data set. These movies were from a variety of entertainment genre (e.g., sci-fi, drama, and so forth) with many classified in multiple genre.

Our data set consisted of 357 viewers who had rated at least 20 of these movies on a 1–5 scale, for a total of 9,708 ratings. The number of movies rated by each viewer ranged from 20 to 69 with a mean of 27. For each viewer we split the ratings into two groups: movies with ratings of 4 or 5 were considered 'liked' while the remainder were considered 'disliked'. For the entire data set, there were 4,771 (49%) disliked movies and 4,937 (51%) liked movies although this is not guaranteed for any particular viewer. Two-thirds of the liked and disliked sets were used for training.

It was not computationally feasible for us to extract and work with features from the full length of each movie, therefore we chose to extract features from only a five minute portion of each movie. In particular, we extracted features from minutes 5 to 10 of each movie. The reason for using this time period as opposed to the first five minutes of each movie is that the very beginning of a movie is often used for displaying credits and therefore may not be representative of the movie as a whole. The visual features might differ as well as there are fewer closed captions. Limiting our method to just a five minute period does have the potential drawback that this time period may not capture what is important to a user. For example, a viewer may prefer movies in which story and character development is drawn out over a long period. Another viewer may enjoy action scenes, which typically don't occur at the very beginning of movies.

The closed captions sets for this time period were extracted and represented using the bag-of-words model. The number of closed captions sets for this five minute period ranged from 32–162, with a mean value of 86. The term-feature vectors representing the closed captions had 4,003 terms. We found that clustering the vectors representing the closed captions was too time-intensive due to the high number of dimensions. To alleviate this problem, we used random projection to reduce the dimensions of the vectors from 4,003 terms to 400 terms.

The idea of random projection is to project a set of points in a high-dimensional space to a randomly selected lower-dimensional subspace [7]. The application of random projection is simple: Given an input matrix X with dimensions $N \times d$ where N is the number of samples and d is the dimensionality of each sample, we can transform this matrix to a new matrix X' with dimensions $N \times k$ by multiplying X by a random matrix R with dimensions $d \times k$ such that $X' = XR$. Papadimitriou et al. [26] show that there is a high probability that pairwise Euclidean distances are kept in the projected subspace.

Several ways of generating the transformation matrix R have been proposed [5]. We chose to generate a matrix in which each element is drawn from a standard normal distribution, $N(0, 1)$. Then each column of this matrix is normalized to one [11].

An advantage of random projection over principal component analysis (PCA), another dimensionality reduction method, is that PCA is very computationally expensive while generating and applying random projections is not. Also, because the application of random projections consists of matrix multiplication, the input matrix X can be partitioned and the matrix R applied to the individual partitions with the results combined if the original matrix X is too large to work with in memory.

To produce the visual features, the first frame from each of the time periods that the closed captions sets were displayed was extracted. Representing each frame by concatenating the RGB values of the pixels would have produced vectors with 253,440 terms. Instead, five levels of a 2D Daubechies 4 wavelet were applied separately to the R, G, and B components, with the results concatenated to form vectors of 363 terms. In addition to reducing the dimensionality, applications of wavelets to images have been shown to improve matching in image retrieval [18].

The visual features and closed captions were clustered separately using agglomerative hierarchical clustering. We performed exploratory data analysis on a subset of our training data to determine which linkage method and distance measure produced the most balanced hierarchies. We found that in general complete linkage with the Euclidean distance produced balanced hierarchies for the wavelet-transformed pixel values. For the closed captions transformed by random projections, none of the combinations of linkage methods and distance measures produced well-balanced hierarchies which suggests that hierarchical clustering may not be the most appropriate form of clustering for closed captions. For both the visual features and closed captions we performed clustering using complete linkage with the Euclidean distance.

When using hierarchical clustering, it is necessary to prune the hierarchy in order to determine the clusters. Many methods have been proposed for determining the true number of clusters in a data set [25]. We chose to partition the hierarchy into clusters using the method proposed by Krzanowski and Lai [19]:

$$\mathrm{DIFF}(g) = (g-1)^{2/p}\ \mathrm{tr}(W_{g-1}) - g^{2/p}\ \mathrm{tr}(W_g)$$

where g is the number of groups, p is the number of dimensions, and W_g is the within-group-sum-of-squares covariance matrix for group g. The number of groups is the g that maximizes

$$\mathrm{KL}(g) = \left| \frac{\mathrm{DIFF}(g)}{\mathrm{DIFF}(g+1)} \right|$$

One disadvantage of this method is that it is unable to determine if the data only represents a single cluster [33]. Calculating the within-group-sum-of-squares is very computationally intensive and therefore we limited our search for the correct number of clusters to cluster sizes of 2–10. Once the clustering was complete, the cluster numbers for the closed captions sets and the corresponding visual features were combined to form the observation symbols for training the HMMs.

The cluster assignments for test samples were determined by finding the training cluster that the test sample would have been assigned to by the complete linkage method using the Euclidean distance as the distance measure.

The 'liked' and 'disliked' HMMs each had two states with equal probability of starting in that state. The initial elements of the transition matrix were all set to have equal likelihood as were the elements of the observation matrix. In constructing the 'liked' and 'disliked' HMMs, we investigated models with 2–10 states with the same number of states in each model. We found that the accuracy, precision, and recall were essentially the same regardless of the number of states. The test samples produced observation symbols that were not present in the training samples, which made it impossible to calculate the log-likelihood of the test sequences. To overcome this, the emission probabilities with values of zero were changed to 10^{-6}.

Our results from combining closed captions and visual features are shown in Table 1. We also generated observation symbols from each type of feature alone in order to determine if the combination of features was an improvement. Our combination approach had an average classification accuracy over the 357 viewers of 54.6%, which is only slightly better than what would be expected if the movies were picked at random. The average classification accuracy when using closed captions or visual features alone was 51.4% and 52.7%, respectively. While the mean value of our combination approach is larger than the mean values of using either type of feature separately, the confidence intervals overlap and we therefore can't state that the results are significantly

Features	Mean	95% CI
CC + Visual	54.6%	(52.7, 56.6)
CC only	51.4%	(49.4, 53.5)
Visual only	52.7%	(50.7, 54.7)

Table 1: Comparison of Features

Number Rated	# users	mean	95% CI
$20 \leq$ movies rated < 30	253	54.2	(51.8, 56.5)
$30 \leq$ movies rated < 40	78	54.0	(50.1, 57.8)
$40 \leq$ movies rated ≤ 69	26	61.5	(56.3, 66.6)

Table 2: Results per number of movies rated

different.

We believe there are several possible reasons for the poor performance of our approach. The first is that out of the 357 users for which we had preference information, 253 of them had rated less than thirty movies. In fact, forty-eight had only rated twenty movies. This is unlikely to be a sufficient amount of data for our approach to effectively learn preferences. When we look at the average results for the viewers who had rated forty or more movies, the mean classification accuracy improves to 61.5%. See Table 2 for the results by number of movies rated.

Another possible problem is that when generating the test observation symbols by combining the cluster numbers for the visual features and closed captions, it's possible to generate observation symbols that never occurred in the training data. Many test sequences contained symbols that never occurred in the training, so even though we gave these symbols an emission probability of 10^{-6} to make it possible to calculate the log-likelihood for the sequence, each of these symbols essentially contributed nothing to that calculation. The remaining symbols may not have been enough to effectively learn preferences. A possible solution to this that we can investigate in a future work is to assign observation symbols to test samples by finding the nearest training combination of closed caption and visual features. This will avoid generating observation symbols that never occurred in the training data. It is not possible to generate unseen observation symbols for the test samples when using only a single type of feature, so this can't account for the poor performance of using either type of feature alone.

In an earlier work [6], we investigated the use of closed captions and DCT coefficients separately. We felt that this work suffered from two problems. First, it did not attempt to combine the text and visual features with the resultant gain in performance that one would expect from such a combination. Second, no consideration was given to the order in which features appear. However, this earlier work does help us to establish a baseline for comparison with the current method in order to determine whether the use of HMMs has improved performance.

The use of closed captions alone in the earlier work had a classification accuracy of 64.04% with a 95% confidence interval of (63.02, 65.05). This exceeds the classification accuracy of 51.4% that we achieved when using closed captions alone in our current work. In our earlier work, all of the closed captions for an entire movie were represented in a single feature vector. In the current work, prior to dimensionality reduction each closed captions set is represented by a feature vector with 4,003 terms for the 4,003 unique words present in the entire data set. However, the maximum number of words that any closed captions set had was eleven which means that more than 99% of the terms were zero. The unlikely overlap of many of the feature vectors makes learning difficult. Using closed captions sets as the segmentation mechanism may not be appropriate since it appears to be over-segmenting the video, both when closed captions are used alone or combined with visual features. A better approach may be to segment by combining several closed captions sets, such as considering every ten sets of closed captions a segment.

The visual features in the previous work were the DC terms of the discrete cosine transform coefficients and the classification accuracy using these features alone was 59.23% with a 95% confidence interval of (58.28, 60.19). The visual features and and methodology were different in the previous work, so it is difficult to make a direct comparison of the effectiveness of different types of visual features.

The results of much of the other work in video recommendation is reported in terms of precision and recall. The precision and recall for our approach that combined closed captions and visual features was 57% and 51%, respectively. Ardissono et al. [1] achieved a precision of 80% and a mean absolute error rate of 30% in their case-based approach to recommendation. The precision of this approach is significantly better than what our approach achieved. Basu et al. [3] achieved precision and recall values of 83% and 34%, respectively. Our approach had worse precision but better recall than this approach.

6. CONCLUSIONS

Traditional approaches to video recommendation have been shown to have relatively good performance. However, for reasons described previously, these approaches are not always applicable. To address this need, we have explored the use of visual features and closed captions extracted from video for learning a viewer's preferences. Overall, however, our results were not good enough to clearly show our approach to be a viable alternative to traditional approaches.

While our results were not very encouraging, we still believe that it is possible to learn preferences from video features and that our work can be extended.

We only applied our method to minutes 5–10 of each movie. Other time periods as well as longer or multiple time periods should be investigated.

It can be applied to the task of classifying video by genre. While there has already been much research in this area, there is still room for improvement. It could also be applied at the shot or scene level. Applications include content filtering, such as identifying violent scenes in movies, or the identification of scenes important to the user. Video summarization can also be performed by finding scenes important to many users. Of the research that has been performed in automatically classifying video by genre, very little has attempted to subdivide genre, such as finding action movies that include car chases or separating romantic comedies from dark comedies.

Much of the research in learning video preferences and classifying video by genre has focused primarily on the entertainment video domain. Other domains, such as education, should be explored more to determine how well our approach would apply to them.

Another area where our work could be applied is video learning. As more and more educational video becomes available, students will have a variety of video choices for learning a particular topic. If the student's performance on a test covering a specific topic is tracked, which should be possible with online courses, then video can be recommended that is similar to those that resulted in the best performance by the student.

7. REFERENCES

[1] L. Ardissono, C. Gena, P. Torasso, F. Bellifemine, A. Difino, and B. Negro. User modeling and recommendation techniques for personalized electronic program guides. In L. Ardissono, A. Kobsa, and M. Maybury, editors, *Personalized Digital Television: Targeting Programs to Individual Viewers*. Kluwer, 2004.

[2] Y. A. Aslandogan and C. T. Yu. Techniques and systems for image and video retrieval. In *IEEE TKDE Special Issue on Multimedia Retrieval*, 1999.

[3] C. Basu, H. Hirsh, and W. Cohen. Recommendation as classification: Using social and content-based information in recommendation. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI98)*, pages 714–720, 1998.

[4] A. D. Bimbo. *Visual Information Retrieval*. Morgan Kaufman, San Francisco, CA, 1999.

[5] E. Bingham and H. Mannila. Random projection in dimensionality reduction: applications to image and text data. *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 245–250, 2001.

[6] D. Brezeale and D. J. Cook. Using closed captions and visual features to classify movies by genre. In *Poster session of the Seventh International Workshop on Multimedia Data Mining (MDM/KDD2006)*, 2006.

[7] S. Dasgupta. Experiments with random projection. In *Sixteenth Conference on Uncertainty in Artificial Intelligence (UAI)*, 2000.

[8] N. Dimitrova, L. Agnihotri, and G. Wei. Video classification based on HMM using text and faces. In *European Signal Processing Conference (EUSIPCO2000)*, 2000.

[9] M. S. Drew and J. Au. Video keyframe production by efficient clustering of compressed chromaticity signatures. In *Poster session of the eighth ACM international conference on Multimedia (MULTIMEDIA '00)*, pages 365–367, 2000.

[10] B. S. Everitt, S. Landau, and M. Leese. *Cluster Analysis*. Oxford University Press, New York, NY, 4th edition, 2001.

[11] X. Z. Fern and C. E. Brodley. Random projection for high dimensional data clustering: A cluster ensemble approach. In *Proceedings of 20th International Conference on Machine learning (ICML2003)*, 2003.

[12] G. Forman. An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, 3:1289–1305, 2003.

[13] W. B. Frakes and R. Baeza-Yates. *Information Retrieval: Data Structures and Algorithms*. Prentice Hall, Englewood Cliffs, New Jersey, 1992.

[14] S. Gauch, J. M. Gauch, and K. M. Pua. The VISION Digital Video Library Project. In A. Kent, editor, *The Encyclopedia of Library and Information Science, Vol. 68, Supplement 31*. Marcel Dekker, August 2000.

[15] X. Gibert, H. Li, and D. Doermann. Sports video classification using HMMs. In *International Conference on Multimedia and Expo (ICME '03)*, volume 2, pages II–345–348, 2003.

[16] GroupLens Research, University of Minnesota. One Million Ratings MovieLens Dataset, URL: http://www.cs.umn.edu/Research/GroupLens, 2005.

[17] J. Huang, Z. Liu, Y. Wang, Y. Chen, and E. K. Wong. Integration of multimodal features for video scene classification based on HMM. In *Third IEEE Workshop on Multimedia Signal Processing*, pages 53–58, 1999.

[18] C. E. Jacobs, A. Finkelstein, and D. H. Salesin. Fast multiresolution image querying. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 277–286, 1995.

[19] W. Krzanowski and Y. Lai. A criterion for determining the number of groups in a dataset using sum of squares clustering. *Biometrics*, 44:23–34, 1988.

[20] R. Lienhart. Comparison of automatic shot boundary detection algorithms. In *In SPIE Conference on Storage and Retrieval for Image and Video Databases VII*, volume 3656, pages 290–301, 1999.

[21] W.-H. Lin and A. Hauptmann. News video classification using SVM-based multimodal classifiers and combination strategies. In *ACM Multimedia*, 2002.

[22] Z. Liu, J. Huang, and Y. Wang. Classification of TV programs based on audio information using hidden markov model. In *IEEE Signal Processing Society Workshop on Multimedia Signal Processing*, pages 27–32, 1998.

[23] C. Lu, M. S. Drew, and J. Au. Classification of summarized videos using hidden markov models on compressed chromaticity signatures. In *MULTIMEDIA '01: Proceedings of the ninth ACM international conference on Multimedia*, pages 479–482, 2001.

[24] W. L. Martinez and A. R. Martinez. *Exploratory Data Analysis with MATLAB*. Chapman & Hall/CRC, Boca Raton, FL, 2004.

[25] G. W. Milligan and M. C. Cooper. An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50(2):159–179, 1985.

[26] C. H. Papadimitriou, H. Tamaki, P. Raghavan, and S. Vempala. Latent semantic indexing: a probabilistic analysis. In *PODS '98: Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pages 159–168, 1997.

[27] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[28] C. Poynton. *A Technical Introduction to Digital Video*. John Wiley & Sons, New York, NY, 1996.

[29] L. R. Rabiner and B.-H. Juang. An introduction to hidden markov models. *IEEE ASSP Magazine*, 3(1):4–16, 1986.

[30] G. D. Robson. *The Closed Captioning Handbook*.

Focal Press, Burlington, MA, 2004.

[31] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[32] B. Smyth and P. Cotter. Surfing the digital wave: Generating personalised television guides using collaborative, case-based recommendation. In *Proceedings of the Third International Conference on Case-based Reasoning*, 1999.

[33] R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a dataset via the gap statistic. *Journal of the Royal Statistical Society, Series B*, 63(2):411–423, 2001.

[34] videolectures.net. URL: http://www.videolectures.net, 2007.

[35] W. Zhu, C. Toklu, and S.-P. Liou. Automatic news video segmentation and categorization based on closed-captioned text. In *IEEE International Conference on Multimedia and Expo (ICME 2001)*, pages 829–832, 2001.