Using Closed Captions and Visual Features to Classify Movies by Genre

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

Diane J. Cook
Department of Computer Science and Engineering
The University of Texas at Arlington
Box 19015, Arlington, TX 76019, USA
cook@cse.uta.edu

ABSTRACT

We investigate closed captions and discrete cosine transform coefficients individually as features for classifying movies by genre and learning user preferences. Using a support vector machine as the classifier, we find that these features work very well for classification by genre but the results are less satisfactory when learning user preferences.

Categories and Subject Descriptors

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

Keywords

closed captions, discrete cosine transform, video genre, classification

1. INTRODUCTION

Many consumers today in the USA have access to hundreds of television channels, not to mention the video available on the Internet and in video stores. While this provides consumers with a variety of options of what to watch, the huge number of choices makes it difficult for consumers to find the video that matches their interests.

One method that consumers use to narrow down the choices is to look for entertainment video, such as television shows or movies, that is in a particular genre. As a result, research has begun on automatically classifying video by genre. Classifying video by genre is useful for recommending entertainment video to a user, but if many video choices are in the same genre then the user must still filter out what they think they will like from the list of possibilities. The existing methods for recommending entertainment video to a user typically use information retrieval techniques that rely on text-based information about the video (e.g., genre, actors, description) or they use collaborative filtering, which makes recommendations based upon the preferences of other users thought to be similar.

These two approaches have shown to be successful, but they do have some drawbacks. In order to use text-based information retrieval techniques, then the text describing the video must exist. Currently this requires a human to prepare this information, at least to do it well. The problem with the collaborative filtering approach is that video that hasn't been seen by similar users can't be recommended. One solution to resolving the problems of these two approaches is to combine them [17].

The approach we have chosen is to extract information from the video itself. In this paper we investigate two different features for their applicability in automatically classifying entertainment video by genre and for learning user preferences: closed captions and discrete cosine transform coefficients. One of the benefits of using closed captions is that words have meaning to humans and it is possible to see how some words tend to be associated with certain genre (e.g., 'stadium' is likely to occur in the sports genre). It is not always as easy for humans to recognize how certain low-level visual and audio features are associated with certain genre. Another benefit is that by utilizing a lexicon such as WordNet [16], it might be possible to perform concept learning, although we don't pursue that in this paper. A third is that extracting text is less computationally expensive than performing image processing.

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.

Many methods exist for representing video, but discrete cosine transform coefficients have the advantage of already being present in MPEG-1 videos as well as some other image and video formats. The discrete cosine transform concentrates much of the energy in an image into a few coefficients [9]. By just using these few coefficients, the visual aspects of the video can be represented.

We believe that these features can also be used for purposes other than identifying entertainment video of inter-

MDM/KDD'06, August 20, 2006, Philadelphia, Pennsylvania, USA.
Copyright 2006 ACM 1-59593-446-4 ...$5.00.

est to a user. Some other uses for automatically classifying video by genre are:

- indexing multimedia databases—help search for particular types of video clips
- learn video preferences
- user modeling
- Internet-based agents to notify a user of video that they might find of interest
- determine genre for scenes within a video—for filtering or summarization

2. RELATED WORK

Bacher [3] designed the Monologue Dissector, which used closed captions to identify jokes within a monologue. Certain words and phrases were hard-coded into his system in order to identify where jokes began and ended. This allowed a user to search for jokes containing words of interest and then playing the video of that joke. Bacher also attempted to perform content analysis on the jokes, but he was never able to produce satisfactory results due to the limited number of words associated with each joke.

Roach and Mason [14] used the audio, in particular mel-frequency cepstral coefficients (MFCC), from video for genre classification. This approach was chosen because of its success in automatic speech recognition. A Gaussian mixture model was used because of its popularity in speaker recognition. The genre studied were sports (specifically fast-moving types), cartoons, news, commercials, and music. A classification accuracy of approximately 80% was achieved when using test sequences of 25 seconds.

Dinh et al. [5] used Daub4 wavelets of the audio from video clips to classify by genre. An advantage of only analyzing audio is that it takes much less computation than analyzing the image properties of a video. Wavelets were compared to features from Fourier and time analysis. Seven sub-bands of the audio were used in the study. The genres studied were news, commercials, vocal music shows, concerts, motor racing sports, and cartoons. Tests were conducted using the C4.5 decision tree, kNN with $k = 6$, and support vector machines with linear kernels. The results for wavelets were comparable to those for features from Fourier and time analysis. kNN was better than C4.5 and support vector machines in all cases. While clips of duration 0.5s, 1.0s, 1.5s, and 2.0s were tested, the duration didn't appear to cause any significant difference in the performance of the classifiers.

Fischer et al. [6] used a three-step process to classify video clips by genre. The genre studied were news, car racing (sports), tennis (sports), commercials, and cartoons. In the first step they extract syntactic properties: color statistics, cuts (or shots), motion vectors, identification of some simple objects, and audio features. In the second step they derive style attributes using information found in step 1. This consists of dividing the video into scenes, using motion information to distinguish between motion due to the camera panning or zooming and object motion, object segmentation, and distinguishing between the sounds of speech, music, and noise. In the third step, modules for each of the style attributes estimates what genre the clip belongs to. A weighted average of the estimates is used to produce a final decision.

Rasheed et al. [13] used low-level visual features to classify movie previews by genre. The genre studied were action, comedy, drama, and horror. The features used were average shot length, shot motion content, lighting key and color variance, with the intent of capturing cinematic principles. Clustering was performed using mean shift clustering. This method was chosen because it can automatically detect the number of clusters and it is non-parametric, so it was unnecessary to make assumptions about the underlying structure.

3. CLOSED CAPTIONING

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 [15]. In addition to representing the dialog occurring in the video, closed captioning also displays information about other types of sounds such as sound effects (e.g., [BEAR GROWLS]), onomatopoeias (e.g., grrrr), and music lyrics (enclosed in music note symbols, ♪). Because closed captioning is not part of the video, it is possible for the viewer to turn them on and off. This also allows them to be extracted from the transmission of the video.

In addition to closed captioning, text can be placed on the television screen with open captioning or subtitling. Open captioning serves the same purpose as closed captioning, but the text is actually part of the video and would need to be extracted with a character recognition program in order to be used for our purpose. Subtitles are also part of the video in television broadcasts although this isn't the case for DVDs. However, subtitles are intended for people who can hear the audio of a video but can't understand it because it is in another language or because the audio is unclear and therefore typically won't include references to non-dialog sounds.

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.

For video that contains human speech but is not closed captioned, speech recognition programs could be used to generate closed captions and thus make it possible to use closed captioning for classification.

4. DISCRETE COSINE TRANSFORM

During the encoding of MPEG-1 video, each pixel in each frame is transformed from the RGB color space to the YC_bC_r color space, which consists of one luminance (Y) and two chrominance (C_b and C_r) values. The values in the new color space are then transformed in blocks of 8×8 pixels using the discrete cosine transform (DCT). Much of the MPEG-1 encoding process deals with macroblocks (MB), which consist of four blocks of 8×8 pixels arranged in a 2×2 pattern. Because the human eye is less sensitive to the chrominance components, these are sampled less frequently than the luminance component. Therefore, each block within a macroblock has DCT coefficients for the luminance component but the same chrominance DCT coefficients are used for all blocks within the macroblock. This results in six sets of 64 DCT coefficients for each macroblock.

The DCT used in the MPEG-1 standard is

$$F(u,v) = \frac{1}{4} C_u C_v \sum_{x=0}^{7} \sum_{y=0}^{7} f(x,y) cos\ \alpha_x\ cos\ \alpha_y$$

where

$$\begin{aligned}
f(x,y) &= \text{value of the original block at coordinates } (x,y) \\
u &= 0,1,\ldots,7 \\
v &= 0,1,\ldots,7 \\
\alpha_x &= \frac{(2x+1)u\pi}{16} \\
\alpha_y &= \frac{(2y+1)v\pi}{16} \\
C_u &= \begin{cases} \frac{1}{\sqrt{2}} & \text{if } u = 0 \\ 1 & \text{otherwise} \end{cases} \\
C_v &= \begin{cases} \frac{1}{\sqrt{2}} & \text{if } v = 0 \\ 1 & \text{otherwise} \end{cases}
\end{aligned}$$

The upper left corner of the block of DCT coefficients has coordinates $(0,0)$ and the lower right corner has coordinates $(7,7)$. It can be seen from the equation that for coordinates $(0,0)$, the DCT produces a value that is proportional to the average value. This value is known as the DC term while the other 63 values are known as the AC terms. While each block has 64 DCT coefficients, for natural images most of the energy of the block is concentrated in a few terms in the upper left corner. That is, most of the information needed to reconstruct the block is found in these terms. One of the ways that compression is achieved in MPEG-1 video is that the DCT coefficients with little energy are discarded [18].

5. DATA ACQUISITION AND PREPROCESSING

We chose 81 movies from the MovieLens project [10] that had been rated by at least 20 users and acquired the DVDs of them. The entire MovieLens dataset consists of 3,883 movies rated by 6040 users on a 1-5 scale for a total of 1,000,209 individual ratings. Each movie in the dataset also has one or more genre labels. Using this dataset allowed us to perform experiments on both classification by genre and learning of individual preferences.

5.1 Processing of Closed Captions

The closed captions were extracted from the DVDs in their entirety including any sound effects (e.g., [DOOR CREAKS]). The words found in sound effects could possibly be used to gain understanding of what is happening at that point in a video, but we did not pursue this in this work. Each movie's closed captions were converted to a feature vector using the bag-of-words model [7]. In the bag-of-words model, the vector for document j (in our study, the closed captions for television show j) contains an entry for each distinct word appearing in the collection of documents. The value of the i^{th} entry, term i, in vector j, is the number of times word i occurs in document j. One potential drawback of the bag-of-words model is that information about word order is not kept.

Next, a stop list [8] was applied to remove common words such as 'and' and 'the'. Such words are unlikely to have much distinguishing power and increase the computational requirements. Then each word was stemmed using Porter's stemming algorithm [12]. This removed the suffixes from words leaving the root. For example, the words 'independence' and 'independent' both have 'indepen' as their root. The stemmed words were used to generate the feature vectors instead of the original words.

5.2 Processing of Video Features

A movie is a collection of frames. Those consecutive frames that are produced by a single camera action are a shot. For our purposes we wished to represent a movie as a collection of shots with each shot being represented by video features found within the shot. To extract the video features that we desired, we modified `mpeg_java`, an MPEG-1 video player [1]. This required first converting each DVD to an MPEG-1 clip. The resolution of the frames in our video was 240×352.

Each frame in the MPEG-1 format is classified as either an I-frame, P-frame, or B-frame [1] depending on how it is encoded. I-frames contain all of the information needed to decode the frame while the other two make use of information found in an I-frame or P-frame.

A color histogram was generated for each I-frame. Shots were detected by comparing the color histograms of consecutive I-frames; if the differences between two of these frames exceeded some threshold, we assumed a shot change had occurred [2].

We extracted DCT coefficients from the first frame of each shot with the assumption that the first frame is representative of the entire shot. In many cases the frames within a single shot will be similar enough for this assumption to hold true. If two consecutive frames within a single true shot are significantly different, then it is likely that the shot detection method will falsely identify a shot at this point anyway and the DCT coefficients for this frame will be included in the collection of shots.

The next step was to represent the frame as a histogram of DCT coefficients. In order to reduce the amount of information needed to represent a frame, we chose to use only the DC term from each block. To see how much information is contained just in the DC term of each block, see Figures 1 and 2. Figure 1 shows a frame from the TV show Sliders that was reconstructed from DCT coefficients. Figure 2 represents the same frame, but the 63 AC terms were set to zero and then the inverse DCT was applied. Although Figure 2 is blocky, it is still possible to recognize it as representing the scene shown in Figure 1. The histograms for each of the three color components were concatenated to form a vector representing the shot in a manner similar to that described in [19]. The resulting vector had $3 \times 2041 = 6123$ terms since DC coefficients can range in value from 0 to 2040 [11].

Once all of the shots had been represented as a histogram of DC terms, we performed k-means clustering with the Euclidean distance as the similarity measure. After the clustering was complete, each movie was represented by a feature vector with a term for each of the k clusters. Movies with

[1] There is also a D-frame in which only the DC coefficients are stored.

Figure 1: Frame from Sliders.

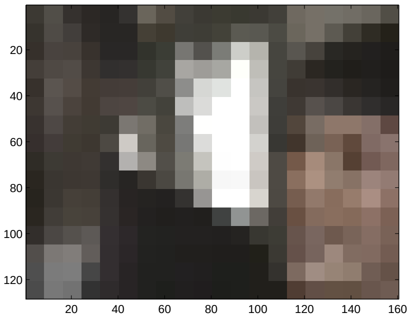

Figure 2: Frame from Sliders in which all values in a block use the DC term.

similar types of shots should have similar feature vectors.

6. EXPERIMENTS

For each of the features under consideration, we performed three types of experiments: classification by genre, classification by user rating, and classification by grouped user ratings. All tests were performed using the support vector machine classifier available in the Weka data mining software [20] with the default linear kernel. Support vector machines are well-suited to classification problems in which there are few training examples but the feature vectors have many terms [4]. Because we did not have much data, we performed 10-fold cross validation. There were 1,116 users who had rated at least 10 of these 81 movies. For each type of experiment the mean classification accuracy was calculated.

Experiment	Class. Accuracy	95% Confidence Interval
CC, by Genre	89.71%	(84.34, 95.09)
CC, Individual Ratings	38.45%	(37.40, 39.50)
CC, Grouped Ratings	64.04%	(63.02, 65.05)
DC Terms, by Genre (20)	88.48%	(82.66, 94.30)
DC Terms, Ind. Rat. (20)	33.26%	(32.33, 34.19)
DC Terms, Gr. Rat. (20)	59.23%	(58.28, 60.19)
DC Terms, by Genre (40)	87.24%	(81.17, 93.31)
DC Terms, Ind. Rat. (40)	32.54%	(31.63, 33.45)
DC Terms, Gr. Rat. (40)	58.76%	(57.83, 59.69)

Table 1: Summary of results.

Some of the 81 movies had more than one genre label in the MovieLens dataset. There were 18 unique genre labels. To classify by genre, we created a separate test file for each genre with each movie being marked as either being in that genre or not.

To classify by user rating, we created a test file for each of the 1,116 users with the movies that user had rated. The label for each movie was the rating that user had given the movie on a 1-5 scale.

To classify by grouped user ratings, we created a test file for each of the 1,116 users with the movies that user had rated. The ratings were grouped: a movie with a rating of 4 or 5 was labeled as ‘liked’ while a movie with a rating of 1-3 was labeled as ‘disliked’.

When classifying by genre using closed captions, feature vectors for all 81 movies were used. These feature vectors had 15,254 terms. When classifying using the individual ratings each user had assigned to the movies, the feature vectors ranged in size from 4401 to 13350 terms depending on the movies rated.

During the extraction of the DCT coefficients, our software failed prior to reaching the end of each movie. This resulted in an inconsistent number of minutes processed for each movie. While the total number of shots for all 81 movies was 46,311, we were only able to obtain a few shots for some movies while for others we obtained hundreds.

The experiments using DCT coefficients represented each movie by a histogram of k shot clusters. We initially set $k = 20$ for the three types of experiments. Then we set $k = 40$ to see if the number of shot clusters would affect the results.

The results for all of the experiments are shown in Table 1 with 95% confidence intervals. The results were virtually the same regardless of whether closed captions or DCT coefficients were used. In each case classification by genre had the best results while classification by individual ratings had the worst. We expected classification by genre of a movie to be easier than learning an individual’s preferences and so were not surprised by these results. We were surprised to find that when using DCT coefficients as the feature the results were very similar regardless of the cluster size. The previously mentioned problem in obtaining consistent data may have contributed to this. Another possible reason was that the threshold value that we used for shot detection may have been too conservative, that is, the amount of difference between two frames necessary to indicate that a shot had occurred may have been too high. This would result in some shots being undetected. In fact, we found that some movies were represented mainly by a few types of shots.

The results when learning preferences using individual ratings ranged from 32.5% to 38.4%. These values are better than the 20% accuracy one would expect to get if the ratings were chosen at random from a 1-5 scale, but there is still much room for improvement. It seems unlikely that users would be satisfied with a recommender system with classification accuracies this low. One reason for this poor performance could be that the number of training examples for each user was too small to learn a user’s preferences.

7. CONCLUSIONS

We have shown that when classifying movies by genre, both closed captions and DCT coefficients perform very well and that, at least when using the methods we employed, the results are essentially the same for both. When using these same methods to learn the video preferences of individuals, the results were better than one would expect to get if the movies were chosen at random but still well below 100% accuracy.

In the future we would like to combine closed captions with visual features to determine what relationship, if any, exists between closed captions and visual features. Our ultimate goal is to learn the preferences of users in order to make recommendations.

8. ACKNOWLEDGMENTS

We would like to thank Dr. Ramesh Yerraballi for suggesting the use of DCT coefficients and for improving our understanding of the MPEG-1 encoding process.

9. REFERENCES

[1] J. Anders. URL: http://rnvs.informatik.tu-chemnitz.de/ jan/MPEG/MPEG_Play.html, 2005.

[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] D. R. Bacher. Content-based indexing of captioned video. SB Thesis, Massachusetts Institute of Technology, May 1994.

[4] K. P. Bennett and C. Campbell. Support vector machines: Hype or hallelujah? *SIGKDD Explorations*, 2(2):1–13, 2000.

[5] P. Q. Dinh, C. Dorai, and S. Venkatesh. Video genre categorization using audio wavelet coefficients. In *Fifth Asian Conference on Computer Vision*, Melbourne, Australia, Jan 23-25 2002.

[6] S. Fischer, R. Lienhart, and W. Effelsberg. Automatic recognition of film genres. In *MULTIMEDIA '95: Proceedings of the third ACM international conference on Multimedia*, pages 295–304. ACM Press, 1995.

[7] G. Forman. An extensive empirical study of feature selection metrics for text classification. *J. Mach. Learn. Res.*, 3:1289–1305, 2003.

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

[9] M. Ghanbari. *Standard Codecs: Image Compression to Advanced Video Coding*. The Institution of Electrical Engineers, London, United Kingdom, 2003.

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

[11] MPEG-1. Coding of moving pictures and associated audio for digital storage media at up to about 1.5 mbit/s. ISO/IEC 11117-2: Video, November 1991.

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

[13] Z. Rasheed, Y. Sheikh, and M. Shah. Semantic film preview classification using low-level computable features. In *3rd International Workshop on Multimedia Data and Document Engineering (MDDE-2003)*, Berlin, Germany, Sep 8 2003.

[14] M. Roach and J. Mason. Classification of video genre using audio. *Eurospeech*, 4:2693–2696, 2001.

[15] G. D. Robson. *The Closed Captioning Handbook*. Focal Press, Burlington, MA, 2004.

[16] S. Scott and S. Matwin. Text classification using wordnet hypernyms. In *Proceedings of the COLING/ACL Workshop on Usage of WordNet in Natural Language Processing Systems*, Montreal, 1998.

[17] 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.

[18] P. Symes. *Digital Video Compression*. McGraw-Hill, New York, NY, 2004.

[19] H. Wang, A. Divakaran, A. Vetro, S.-F. Chang, and H. Sun. Survey of compressed-domain features used in audio-visual indexing and analysis. *Journal of Visual Communication and Image Representation*, 14(2):150–183, June 2003.

[20] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools with Java implementations*. Morgan Kaufmann, San Francisco, 2000.