Automated Movie Genre Classification with LDA-based Topic Modeling

Brandon Chao
Duke University

Ankit Sirmorya
Universty of Florida

ABSTRACT

Movie genre classification is a challenging problem with many potential applications. Whereas many prior approaches rely on image, audio, or motion features to classify movies, we consider using textual content analysis instead, which is a comparatively less computationally expensive and time consuming process. In this paper, we present a novel system for movie genre classification that uses probabilistic topic modeling of the movie's script as its main component. Our approach uses latent Dirichlet allocation, a topic modeling algorithm, to train our model and discover common themes present in movie scripts of the same genre. We then compute the cosine similarity of the feature vectors from our trained and test models and use this value to identify the movies' genres.

General Terms

Movie Genre Classification, Linguistic Feature Extraction, Probabilistic Topic Modeling

Keywords

Video Genre Identification, Latent Dirichlet Allocation, LDA

1. INTRODUCTION

With the spread of the internet, the amount of available video data has expanded significantly, creating the need for methods than can automatically analyze and classify large collections of digital media. As a result, the automated Video Genre Identification (VGI) problem has motivated several recent studies and challenges such as the TrecVid evaluation campaign and the ACM Multimedia Grand Challenge by Google in an effort to discover efficient ways to categorize videos into separate genres.

Much of the current work in VGI depends upon features extracted from image and cinematic analysis. Zhou et al. [1] achieved an accuracy of 71.58% in movie genre identification by using scene categorization from movie trailers. Truong et al. [2] extracted computable features from video data by encoding it in MPEG-1 format and using a C4.5 decision tree classifier for genre labeling. Rasheed et al. [3] established a classification error rate of 17% by utilizing low-level visual features along with cinematic principles.

Several experiments have also been carried out exploring audio-based approaches to organize videos by genre. In [4], the authors evaluated a mel-frequency cepstral coefficients (MFCC) and neural network system for VGI and produced a correct classification rate of 51% in a 5-genre task. Additionally, Jasinschi et al. [5] used sets of relative probabilities and mid-level audio categories to achieve automated classification of TV program genres at a precision of 65.2% and a recall of 89.2%.

However, relatively few studies have focused on using text based approaches [6, 7, 8], such as applying text-based classification methods on closed captions or Teletext streams, to solve the VGI problem. Despite the fact that many distinctive features of video genres can be extracted from linguistic content, one of the major concerns in utilizing text for video genre classification is the lack of textual information associated with videos. However, since we are primarily concerned with classifying movies where textual information in the form of movie scripts is readily available, this is not a major issue for our applications. For movies without available scripts, another option is to use Audio Speech Recognition (ASR) techniques to transcribe movies and subsequently utilize that data for linguistic content extraction.

In this paper, we suggest a text-based solution to video genre identification where a classifier is built on top of a topic model. This approach is based on training a probabilistic topic model of relevant topics specific to each movie genre using latent Dirichlet allocation (LDA). The genre of a new movie is then identified by generating a feature vector representing the movie's most relevant topics and computing the cosine similarity between this vector and our model to determine the genres that most closely represent the movie. The results of this method prove that analyzing textual information alone has great potential in identifying movie genres as compared to other methods that incorporate video and audio features.

The remainder of this paper is organized as follows. In Section 2, we review the concept of LDA and its application to topic models. In Section 3, we describe our novel approach to the movie genre identification problem as well as how we evaluate it. Section 4 details the results of our methodology, and we finish with our conclusions in Section 5.

2. PROBABILISTIC TOPIC MODELING WITH LATENT DIRICHLET ALLOCATION

Topic Models Probabilistic topic models are used to analyze the contents of documents and reveal the meaning of words [9, 10]. They are based on the fundamental idea that a document is a mixture of topics. Accordingly, a document can be decomposed into a set of weighted topics, where each topic consists of a cluster of words that occur frequently together.

In this paper, we denote $P(z)$ as the probability distribution over some topic z in a document and $P(w|z)$ as the probability distribution over the words w of the document given topic z. We indicate w_i as the ith word token which is selected via a generative process where a topic is first sampled from the document's topic distribution, and then a word is chosen from that topic's word distribution. We denote the probability that the jth topic is sampled for the ith word token by $P(z_i = j)$, and the probability that word w_i is under topic j by $P(w_i|z_i = j)$. The following distribution specifies the topic model within a document where T is the total number of topics:

$$P(w_i) = \sum_{j=1}^{T} P(w_i|z_i = j)P(z_i = j) \tag{1}$$

$P(w_i)$ is the probability that a particular word w_i belongs to any of the T topic baskets in the document. The global maxima of the Gaussian curve obtained from this distribution thus represents the most relevant topic to the word.

Latent Dirichlet allocation LDA is an example of topic modeling which essentially comprises of three components: the observed variables which are the words in the documents, the hidden variable which is the underlying topic structure, and the generative process which defines the joint probability distribution over these variables. This joint probability is then used to compute the conditional probability distribution of the hidden variables given the observed variables. In other words, LDA allows us to discover the hidden thematic topic structure of a document, given the words in the document.

The LDA process is formally described with the following notation. $\beta_{1:K}$ denotes the topics where each β_k is a distribution over the words in the document. θ_d represents the topic proportions for document d, where $\theta_{d,k}$ is the topic proportion for topic k in document d. z_d represents the topic assignments in document d, where $z_{d,n}$ is the topic assignment for the nth word in document d. Finally, the observed words for document d are w_d, where $w_{d,n}$ is the nth word in document d. Using these notations, the generative process for LDA corresponds to the following joint distribution of the hidden and observed variables:

$$p(\beta_{1:K}, \theta_{1:D}, z_{1:D}, w_{1:D}) = \prod_{i=1}^{K} p(\beta_i) \prod_{d=1}^{D} p(\theta_d)(\prod_{n=1}^{N} p(z_{d,n}|\theta_d)p(w_{d,n}|\beta_{1:K}, z_{d,n})) \tag{2}$$

Thereafter, we use the joint probability to compute the conditional distribution of the topic structure given the observed documents. The posterior probability can be defined as:

$$p(\beta_{1:K}, \theta_{1:D}, z_{1:D}|w_{1:D}) = \frac{p(\beta_{1:K}, \theta_{1:D}, z_{1:D}, w_{1:D})}{p(w_{1:D})} \tag{3}$$

In Equation 3, the numerator represents the joint distribution of all the random variables and the denominator represents the probability of seeing the observed corpus under any topic model. While the joint distribution in the numerator can be easily calculated for any combination of the random variables, the number of possible topic structures in the denominator is exponentially large and is impractical to compute. However, this portion of the posterior can be efficiently approximated via various sampling algorithms, of which the most common is Gibbs sampling.

Dirichlet distribution and hyperparameter optimization As presented by Blei et al. [11], LDA is a graphical model [12] for topic discovery using the Dirichlet prior. The probability density of a T-dimensional Dirichlet distribution over the multinomial distribution $p = (p_1, \ldots, p_T)$ is defined by:

$$Dir(\alpha_1, \ldots, \alpha_T) = \frac{\Gamma(\sum_j \alpha_j)}{\prod_j \Gamma(\alpha_j)} \prod_{j=1}^{T} p_j^{\alpha_j - 1} \tag{4}$$

In Equation 4, α_j denotes the hyperparameter which can be interpreted as a prior observation for the number of times a topic j is sampled in a document. A high α value implies that every document is prone to contain a blend of the majority of the topics, and not one particularly dominant topic. Conversely, a low α value implies that it is probable that a record is composed of only a couple, or even one, of the topics.

3. APPROACH

For the automated movie genre classification task, we classify the movies based on a list of the 9 most common movie categories in our dataset - Action, Adventure, Comedy, Crime, Drama, Horror, Mystery, Romance, and Sci-Fi. The following sections describe our approach. Our method involves cleaning the raw data into a format that could be fed into the modeler (Section 3.1), creating a topic model via LDA from a corpus of movie scripts and converting each topic to a standard feature vector (Section 3.2), using a novel method of categorizing new movies through topic modeling and comparing the feature vector to those of each genre using their cosine similarity (Section 3.3), and evaluating our model based on the F-score and other metrics (Section 3.4).

3.1 Corpus and data cleaning

Our corpus consists of 1094 movie scripts downloaded from the Internet Movie Script Database (IMSDB) in HTML format. The movie scripts in this dataset are American Hollywood movies released from 1935 to 2015. The distribution of the genres of the movie in our corpus is shown in Figure 1. Note that the total count of the movie genres exceeds the number of movies in our training set since some movies belong to multiple genres. On average, each movie is classified in 2.88 unique genre categories and the maximum number of genres a movie is classified in is 5.

We cleaned this dataset using jsoup, a Java HTML parsing library, to remove the HTML tags from the raw files and extracted the movie script contained in the 'srctext' CSS class. We then converted the files to plain text and used the data in this format to train and test our model. For the purposes of the model, the words were tokenized from the scripts by whitespace.

From our dataset, we randomly allocated 80% of the movie scripts to the training set and the remaining 20% to the test set. We ran 5 rounds of experiments and report the average performance as evaluated by the metrics we describe in Section 3.4 below.

3.2 Creating the model

We created our model using MALLET [13], an open source, Java-based package from the University of Massachusetts Amherst. This

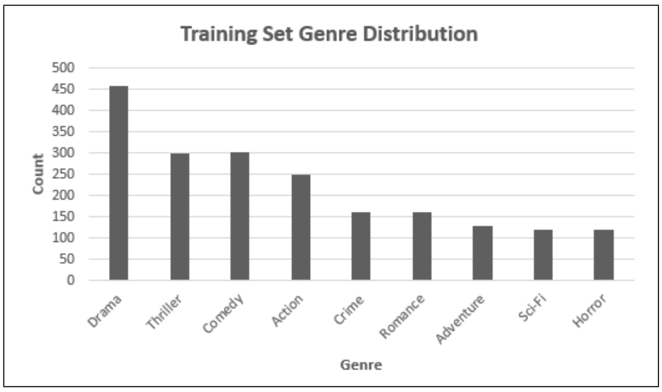


Fig. 1. Distribution of distinct movie genres in training set

package has several functionalities, one of which is a topic modeling toolkit that contains a Gibbs sampling-based implementation of LDA. One of the features of this implementation allows us to optimize the hyperparameters of LDA with the *optimize-interval* input in order to more accurately categorize our data. For our experiments, we used this toolkit, run with an alpha value of 20, to create the topic models in our experiments. The alpha value was chosen due to the broad nature of our data. As noted in Section 2, a high alpha implies that each movie script contains a blend of the majority of the topics, which makes sense intuitively since most words are not exclusive to only one or a few movie genres.

Topic discovery for each genre The movie scripts in our training set were classified into separate directories based on their listed genres. Movies categorized in multiple genres were included in each of the directories for its genres. We ran the LDA topic modeler on each genre directory, which produced a collection of T topics for that genre. This variable is preset in the modeler and for our experiments, we used $T = 100$, which was determined based on the number of training movies in each genre category. Each topic is characterized by a weight and a list of words within the topic as demonstrated in Figure 2.

For each row of the output, the first value is the serial number, the second value is the weight associated with the topic, and the remaining words are the words that make up that topic.

Through this process, we generated 9 topic models that correspond to the movie genres we had decided on previously. Finally, we converted each topic in the 9 genre-based topic models into a corresponding feature vector using the bag-of-words model. We denote $\theta(i, j)$ as the key-value pair where the key is the jth word in the ith row and the value is the weight of the ith row. For example, $\theta(0, 0)$ in Figure 2 above would be (matrix, 0.00835). The feature vector for the ith row in the trained topic model for each genre is defined as $\vec{L_i}$, which is a list of the key-value pairs defined above by $\theta(i, j)$. These feature vectors quantify the occurrences of the most common words in the movie scripts of each genre and are subsequently used to identify the most closely matching genres for the movies in our test data set.

3.3 Testing the model

We benchmarked our model on the remaining 20% of our dataset. To start, we ran the LDA topic modeler on an individual movie, which yielded a weighted distribution of the most relevant topics for that movie. This was returned in a similar format as Figure 2 shown previously.

Table 1. F-measure for various k values

k	F-measure
150	0.438
160	0.425
170	0.468
180	0.433
190	0.482
200	0.466
210	0.434
220	0.445
250	0.468
300	0.460
350	0.434
400	0.464
450	0.434
800	0.457

We denote $\phi(i, j)$ as the key-value pair from the test model where the key is the jth word in the ith row and the value is the weight of the ith row. We define $\vec{T_k}$ as the top-k $\phi(i, j)$ key-value pairs based on the value. For each new movie in the test dataset, we use $\vec{T_k}$ as the feature vector to represent the movie. The value of k was empirically determined at a point which maximized the f-measure of our movie-genre classifier. For our dataset of movie scripts, 190 was established as the optimal value of k. This is shown in Table 1.

Our next task is to quantitatively compare the feature vector generated from the individual movie in the test set with the genre feature vectors trained from our model. This is accomplished by computing the cosine similarity between each of the $\vec{L_j}$ feature vectors of a genre in the trained model and the $\vec{T_k}$ feature vector of the movie in test dataset. The similarity score Δ_{mn} of the mth genre with the nth test movie in our training model is as follows:

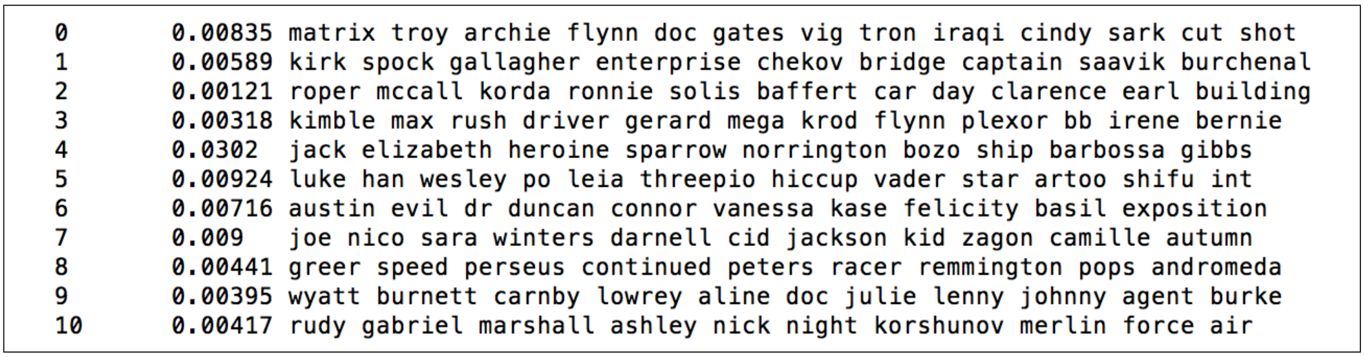
$$\Delta_{mn} = \vee_{j=1}^{100}\left(\frac{\overline{\vec{T_{kn}}} \cdot \overline{\vec{L_{jm}}}}{|\vec{T_{kn}}||\vec{L_{jm}}|}\right) \qquad (5)$$

As Equation 5 shows, the maximum cosine similarity between the $\vec{T_k}$ feature vector with each of the $\vec{L_j}$ feature vectors is taken as the final similarity score representing how closely the new movie matches a particular genre. The process is repeated for all 9 genres in our trained model.

The final genre prediction from our model for a test movie can be represented as:

$$\text{Genre Predictions} = \text{top-X}(\arg\max_{m} \Delta_{mn}) \qquad (6)$$

The above equation represents the top-X movie genres with the highest similarity scores Δ_{mn}, which are ultimately chosen by our model as the genres for the movie. Since the movies in our dataset can belong in up to five genres, we have conducted our experiments using values of $X = 5$ and $X = 6$ in order to fully represent the movies from the test set. We refer to these two algorithms as LDA5 and LDA6 respectively.



```
 0      0.00835 matrix troy archie flynn doc gates vig tron iraqi cindy sark cut shot
 1      0.00589 kirk spock gallagher enterprise chekov bridge captain saavik burchenal
 2      0.00121 roper mccall korda ronnie solis baffert car day clarence earl building
 3      0.00318 kimble max rush driver gerard mega krod flynn plexor bb irene bernie
 4      0.0302  jack elizabeth heroine sparrow norrington bozo ship barbossa gibbs
 5      0.00924 luke han wesley po leia threepio hiccup vader star artoo shifu int
 6      0.00716 austin evil dr duncan connor vanessa kase felicity basil exposition
 7      0.009   joe nico sara winters darnell cid jackson kid zagon camille autumn
 8      0.00441 greer speed perseus continued peters racer remmington pops andromeda
 9      0.00395 wyatt burnett carnby lowrey aline doc julie lenny johnny agent burke
 10     0.00417 rudy gabriel marshall ashley nick night korshunov merlin force air
```

Fig. 2. MALLET output of trained models

3.4 Evaluation

Baselines We use several baselines for the evaluation of our model. The first baseline is a majority labeling algorithm. It operates by finding the top N genres from the movies in the training set. Then, it predicts the same top N genres for each test instance found in the training set. Since we chose to report results for the top 5 and top 6 approach of the LDA method, we will compare them with two analogous versions of the majority labeling algorithm - one selecting the top 5 genres and another selecting the top 6. For evaluation purposes, we will call these algorithms ML5 and ML6.

The second baseline is linear SVM using a one-vs-the-rest scheme. To run SVM, we first converted the dataset into a TF-IDF vector where the vector space is defined by all of the instances in the training corpus. (Note that when preparing the text vectors, we excluded all English stop words.) We tested multiple variations of TF-IDF with different smoothing functions and n-grams by using bigrams in our TF-IDF vectors. However, since using bigrams and different smoothing functions did not show any significant deviation in the results of the evaluation metric, we will only show results using unigrams and Laplace smoothing. For the SVM itself, we used a *square hinge* loss function with a penalty parameter of 1 and a maximum of 1000 iterations.

The third baseline is the random forest classifier. For random forest, we used 5 estimators with a maximum split of 50. This method includes the same preprocessing steps as SVM described above.

Evaluation metrics We consider average F-measure as our primary evaluation metric and compute it with the formulas below.

$$P_{avg} = \frac{TP_{avg}}{TP_{avg} + FP_{avg}} \tag{7}$$

$$R_{avg} = \frac{TP_{avg}}{TP_{avg} + FN_{avg}} \tag{8}$$

$$F_{avg} = 2\frac{P_{avg} * R_{avg}}{P_{avg} + R_{avg}} \tag{9}$$

To compute the number of TP, FP, and FN, we followed the following process. Consider a predicted vector and a ground truth vector for a movie with genres $[A, B, C, D, E, F]$:

Predicted vector = $[A, B, C, D, E]$
Ground truth = $[B, C, D, F]$

Table 2. Performance of various video genre classification methods

Algorithm	Precision	Recall	F-measure
ML5	0.3242	0.6672	0.4167
ML6	0.3066	0.7346	0.4160
SVM	0.5028	0.3028	0.3516
Random Forest	0.5563	0.3530	0.4068
LDA5	0.3183	0.7206	0.4415
LDA6	0.3549	0.8036	0.4923

Table 3. F-measure for different iterations of ML5 and ML6

Iteration	ML5 F-measure	ML6 F-measure
1	0.41	0.32
2	0.39	0.38
3	0.42	0.35
4	0.34	0.40
5	0.33	0.39

In this example above, TP = 3, since we correctly predict 3 genres (B, C, D). FP = 2, since two genres are incorrectly predicted (A, E). And, FN = 1 since one genre is not predicted but is in the ground truth (F).

For every movie in the test set, we computed the TP, FP and FN. The final evaluation metric is generated by using the mean TP, FP, and FN to compute P_{avg} and R_{avg}.

4. RESULTS

In this section, we compare our methods to the baselines using the stated evaluation metrics. Table 1 describes the average precision, recall, and F-measure for all of the models.

The first baseline, ML5 and ML6, performed relatively well on precision, recall, and F-measure. For a thorough comparison of our algorithms with ML5 and ML6, we ran ML5 and ML6 on random selections of training and test instances. We found that this baseline fluctuates a lot with changes in the distribution of the training data, which is to be expected since these two algorithms rely solely on the distribution of the training instances. As such, these baseline algorithms are less robust than LDA5 and LDA6. That is, their result is totally random and does not generalize as a solution for the problem. The following table shows the F-measure varying significantly as we run ML5 and ML6 on randomly generated test and training sets.

The next baseline is SVM. SVM performs better overall in precision but worse in recall. This is because SVM penalizes false positives more severely, and as a result, produces an increased number of false negatives thus decreasing the recall. This directly implies that for any given test instance, SVM will correctly predict a subset of the correct genres but will not predict all of them. This is confirmed by the high precision value of 50%, and low recall value of 30%, suggesting that the ability to predict a high number of correct genres is not great. This produces an F-measure of 35%. LDA5 has a higher F-measure of around 44% and LDA6 performs even better with F-measure of 49%.

The final baseline is the random forest algorithm. Random forest is able to generate highly precise genres for a given test instance but misses out on many genres that are present in the ground truth. Random forest performs better than SVM in all three metrics because of more robustness when dealing with skewed datasets. However, LDA5 and LDA6 have higher F-measure values than random forest because of their ability to produce significantly higher recall while not compensating as much on precision.

5. CONCLUSIONS

Textual analysis techniques such as topic modeling have the potential to make significant contributions in the area of video genre identification. Our approach of using an LDA-based topic modeler as the main component to classify movies based on their genre is just one application of a category of algorithms that can serve as alternative or supplementary methods to video and audio analysis of digital content.

Certainly, we have only scratched the surface of this approach. In our experiments, despite only using the movies' textual features for predictions, we were able to classify the movies genres at a reasonably efficient rate. We plan to further investigate how incorporating other parameters in our model can improve its efficiency. Examples of features that we could factor into the model in the future include analyzing the semantic context of the words to generate the topic models, or considering user comments and reviews as inputs to the model. Other options include incorporating visual cues from the movies' video features such as facial expressions or background scenery to supplement our model. We also wish to extend the applications of our model to unscripted videos by extracting text from the videos using ASR techniques, and then applying our model to the text to identify the video's genres.

6. REFERENCES

[1] H. Zhou, T. Hermans, A. V. Karandikar and J. M. Rehg, *Movie Genre Classification via Scene Categorization*, in 18th ACM International Conference on Multimedia, 2010.

[2] B. T. Truong, S. Venkatesh and C. Dorai, *Automatic Genre Identification for Content-Based Video Categorization*, IEEE International Conference on Pattern Recognition, 2000.

[3] Z. Rasheed, Y. Sheikh, and M. Shah, *On the Use of Computable Features for Film Classification*, in IEEE Transactions on Circuit and Systems for Video Technology, 2001.

[4] M. Roach, L. Q. Xu, and J. Mason, *Classification of non-edited broadcast video using holistic low-level features*, in IWDC, 2002.

[5] R. S. Jasinschi and J. Louie, *Automatic TV program genre classification based on audio patterns*, in Euromicro Conference, 2001.

[6] W. Zhu, C. Toklu, and S. P. Liou, *Automatic news video segmentation and categorization based on closed-captioned text*, in Multimedia and Expo, ICME, 2001.

[7] S. Oger, M. Rouvier and G. Linares, *Transcription Based Video Genre Classification*, IEEE International Conference on Acoustics, Speech, and Signal Processing, 2010.

[8] M. Blosseville, G Hebrail, M. Monteil, N. Penot, *Automatic Document Classification: Natural Language Processing, Statistical Analysis, and Expert System Techniques Used Together*, Sigir Forum (Acm Special Interest Group on Information Retrieval), 1992.

[9] M. Steyvers, and T. Griffiths. *Probabilistic topic models*, Handbook of latent semantic analysis 427.7 (2007): 424-440.

[10] D. M. Blei, T. L. Griffiths , M. I . Jordan and J. B. Tenenbaum (2004), *Hierarchical topic models and the nested Chinese restaurant process*, Advances in Neural Information Processing Systems, 2004.

[11] D. M. Blei, A. Y. Ng, and M. I . Jordan, *Latent Dirichlet allocation*, Journal of Machine Learning Research, 2003.

[12] T. Hofmann, *Probabilistic Latent Semantic Analysis*, In Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, 1999.

[13] McCallum, Andrew Kachites. *MALLET: A Machine Learning for Language Toolkit.* http://mallet.cs.umass.edu. 2002.