Categorization of YouTube Videos

Nitesh Gupta

Computer Science

Iowa State University

nkgupta@iastate.edu

Tanmay Gore

Computer Science

Iowa State University

tgore03@iastate.edu

**Abstract**

Classification of web-based videos is an important task in video search and ads targeting applications. The proposed method categorizes YouTube videos in different genres like Comedy, Horror, Romance, Sports, Technology in a supervised manner. It maps the different features of a video to TF-IDF vectors in a sparse zero matrix. Dimensionality reduction is performed using PCA on the obtained features. Fully connected feed forward neural networks and k nearest neighbor algorithms are used for the training purpose. The method then computes a new TF-IDF vector array for the new videos and classifies it to one of the categories obtained. The results obtained from both the models are compared on the basis of running time and classification accuracy.

**Keywords:** TF-IDF, PCA, YouTube, Neural Networks, k Nearest Neighbors, Classification.

1. **Introduction**

Categorization of videos is an increasingly prominent area of research, rising with the quantity of videos shared through online platforms such as YouTube. Its applications are of paramount importance to video search and website monetization through advertisements targeting. However, the classification of videos to various category poses a great challenge. An accurate classification of videos help user to reach to the desired video without wasting a lot of time searching. It also provides the monetary befit to ads targeting applications.

In this paper we evaluate the efficacy of two different machine learning models for video categorization. This project includes a categorization on the basis of few selected features of a particular video. The selected features are tags, video title and video description.

This project explores four main ideas – TF-IDF, PCA, Artificial Neural Networks and k Nearest Neighbor. TF-IDF stands for *term frequency-inverse document frequency*, and the TF-IDF weight is often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. (http://www.tfidf.com/ n.d.) The importance of a word is proportional to the number of times a word appears in the document but is offset by the frequency of the word in the corpus.

The following are the equations for TF and IDF respectively:

… (1)

… (2)

The TF-IDF is the product of equation (1) and (2).

The second concept that we explored is Principal Component Analysis (PCA). PCA is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. This transformation is defined in such a way that first principal component has the largest possible variance and each succeeding component has the highest possible variance under the constraint that it is orthogonal to the preceding components. A data point can have thousands of features. The running time for any machine learning algorithm depends on the number of data points in the dataset and also the dimensionality of a data point. Through PCA it is possible to achieve the faster running time for the dataset by reducing the dimensions of a data point while preserving most of its value. Thus, PCA plays an important role in the situation when there is a need to perform various algorithms on a particular dataset to obtain the comparison results.

Artificial Neural Network(ANN) are one of the important tools used in machine learning. These are the brain inspired systems that are intended to replicate the ways the humans learn. An ANN is based on a collection of connected units called artificial neurons. Each connection between neurons transmits a signal from one to another. The output of each artificial neuron is calculated by a nonlinear function of the sum of its input. These connections also have weights that adjusts the learning process. Neural Networks consists of the input layer, hidden layers and output layer. Hidden layers play the role of transforming the input into the form desired in the output. These are excellent tools for finding computationally intensive patterns underlying the data.

k Nearest Neighbor is an algorithm used for classification and regression in the field of pattern recognition. The input consists of k closest training examples in the feature space. In k-NN classification, the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors. (Reference: Wikipedia)

1. Methodologies

To categorize a video, we perform three major steps: (1) Extract 20000 videos from YouTube API (2) Computing TF\_IDF to convert the string features into equivalent numeric features. (3) Applying PCA for the dimensionality reduction. (4) Train the machine learning model using Feed Forward Fully Connected Neural Networks and get the running time, test accuracy and confusion matrix. (5) Apply k Nearest Neighbor for the classification of the data among k categories and get the running time, test accuracy and confusion matrix.

* 1. Extraction of the YouTube videos

YouTube provides two APIs for retrieving videos information. The Streaming API provides methods to retrieve videos in bulks and filter them out based on location, keywords etc. We used the Streaming API for collecting 20000 US videos and store it in a file in JSON format. The REST API provides methods to retrieve video-specific information such as video title, description, tags information etc. We use this API to get most trending US videos and to classify them to various categories.

* 1. Preprocessing and Dimensionality Reduction for the YouTube Videos

For categorizing YouTube videos, we had to first assign a TF-IDF score for every word in video tags. This would enable us to have a vector representation of every feature that was obtained. Once the TF-IDF scores were obtained, a set containing every word was made. Let’s call this set “*word\_set*”. Every word in *word\_set* was assigned a unique index (*index\_word*). Next, a sparse zeros array containing dimensions [number of tags, length of *wordSet*] was generated. We iterated through the TF-IDF scores for every word in a tag and added them to the specific index as determined by *index\_word*. This was done for every video in the sample space.

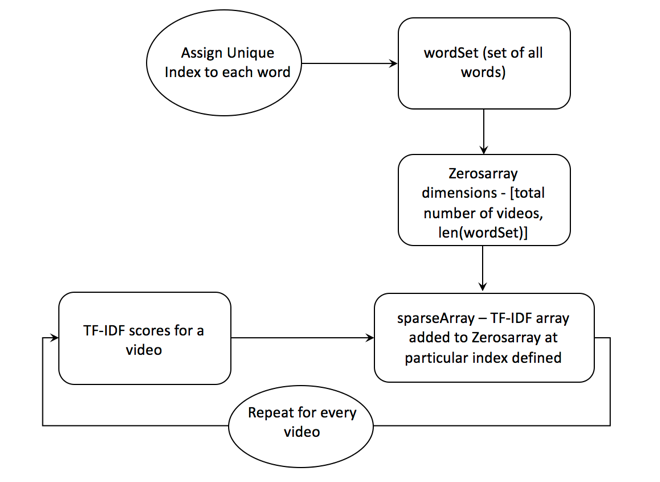


Figure 1: Block diagram illustrating clustering phase

The dataset of the videos contained 16 different categories. This conclusion was reached after computing the total distinct categories in the extracted data.

PCA is applied to the obtained matrix to reduce the dimensionality of the data points. This step reduces the number of the features significantly while preserving the information lies in the data.

* 1. Classification Methodology

This phase of the algorithm deals with classifying the new videos to the specific categories. The classification is obtained using two methods, Feed Forward Fully Connected Neural Networks and k Nearest Neighbors. New videos are obtained using YouTube API. We first filter out the stop words from the tags of this video. Next, we generate the TF-IDF value for each word obtained in the tags.

Using the TF-IDF values generated for the new set of videos, we populate the sparse zero array with these values using the same methodology described in section 2.2. The sparse array of zeroes specific to the new dataset is computed and this will have dimensions [1, length of *wordSet*]. We use this matrix to assist us in classifying the new videos into various categories. The test data set also contains the actual category of the videos which is then used to calculate the classification accuracy and to obtain the confusion matrix.

The results obtained from Neural Networks and k Nearest Neighbor are compared on the basis of the accuracy on the test data set and the running time. A comparison chart is also made.

1. Experiments

In this section, we experiment with different set of features, used to train our machine learning models. The set of total features contains a) Video Title 2) Video Description 3) Tags for the video. The experiment is also performed with various values of parameters for the machine learning models.

* 1. Neural Networks

Neural Network used for the training purpose is Feed Forward Full Connected Neural Network. Softmax function is used for the output layer and 1 of c encoding is used. Early stopping technique is used to stop the training. Experiments are performed using following values for different parameters:

1. Error Function: Sum of Square Error, Cross Entropy Error
2. Hidden Units**:** 
   1. Number of Neuron in the hidden unit: 100, 200, 300
   2. Number of hidden layers: 2, 3
   3. Type of hidden Units: ReLU, tanh
3. Learning Rates
4. Momentum Rate
5. Input Scaling
6. Different set of features for training of the model

K fold Cross validation is also performed for the training of the model. Once the neural network mdoel is trained it is run for the test data set to obtain the classification accuracy.

* 1. k Nearest Neighbors

The experiment is done with different values of k to obtain the category for a particular video. Also, various set of features are used for the training purpose to obtain the best classification model.

A comparison of the results obtained from both the models is made on the basis classification accuracy. Runtime for various algorithms for the different set of experiments is also compared.

1. Results & Discussions

We found that the neural network performed a better job of classifying the YouTube videos than kNN when the input features are more. The best neural network model obtained was with Cross Entropy Error Function, ReLU hidden layers with 3 layers and 100 neurons in each layer. The kNN performed good when run for single features (only tags or video description) but gives less accuracy when the multiple features are combined as compare to ANN. We were able to achieve an accuracy of 70 percent on our test dataset with the ANN while kNN gives the best accuracy of around 65 percent. kNN best perform with the value of k as 10.

The dimensionality reduction with PCA decreases the runtime with a significant amount with very less loss in the underlying information of the features. Our reduction preservers the 95 percent of the information and reduces the dimension of data point from 18000 to 3000 which is a significant reduction. This results in a decrease in run time from 600 second to 150 seconds.

We started with only one feature(tags) to obtain the training dataset and subsequently included all the three features. There is an increase in the accuracy of the ANN model form 60 percent to 70 percent. The below output shows the best parameters obtained for both ANN and kNN:

**Table 1: Best Parameter Values obtained for ANN**

|  |  |
| --- | --- |
| Feed Forward Fully Connected Neural Networks | |
| Error Function | Cross Entropy |
| Hidden Units | ReLU |
| Number of Hidden Units | 3 |
| Number of Neurons in each hidden unit | 100,200,100 |
| Learning Rate | 0.9 |
| Momentum Rate | 0.01 |
| Features Used | Tags, Title, Description |
| Accuracy | 71.08 percent |

**Table 2: Best Parameter Values obtained for kNN**

|  |  |
| --- | --- |
| k Nearest Neighbors | |
| k | 10 |
| Features Used | Tags |
| Accuracy | 65.66 percent |

1. **Conclusion**

This project focused primarily on ANN, kNN and categorizing similar videos together. The novelty came in the fact that we were using the most recent videos and categorizing it into similar groups such it can be an advantage to the user in order to search or get the monetary benefit from it through ads. In addition, we also explored how Principal Component Analysis can be used for the dimensionality reduction when input data pits have a large number of dimensions. It provided the approximate results with a significant improvement in the running time.

In the end, we were able to develop an efficient classification system that would correctly group a video into a category containing similar videos.

1. **References**

https://en.wikipedia.org/wiki/Artificial\_neural\_network

<https://en.wikipedia.org/wiki/Principal_component_analysis>

<https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm>

https://www.kaggle.com/datasets