CIS 519 Final Paper: Video Recommendations

Nazih Bissat
University of Pennsylvania
Systems Engineering
nbissat@seas.upenn.edu

Christian de Abreu
University of Pennsylvania
Bioengineering
dechrist@seas.upenn.edu

Ali Kozlu
University of Pennsylvania
Cognitive and Computer Science
akozlu@seas.upenn.edu

Bernardo Penteado
University of Pennsylvania
Mechanical Engineering & Wharton
pbern@seas.upenn.edu

Abstract

Recommendation Systems for video channels are highly investigated areas in the machine learning domain because of their importance in retaining viewers. This paper presents a comparison between two recommendation systems developed on the basis of a similarity index between a video and all other videos in Vimeo's staff picks database released in January of 2018. The first model extracts text, thumbnail, and numerical features individually and used categories as a proxy for accuracy of prediction of top k recommended clips. The second model puts these three features in one feature space through normalization and dimensionality reduction and returns a list of top ten similar video clips based on distance. It was found that Model 1 yielded total category matching of 37.1% with text, which consequently validated the proposal of Model 2 both. In the future, Model 2 would be further optimized by user input to weigh features based on similarity or creating a graph where a more robust algorithm such as Page Rank could more accurately offer recommendations, though not necessarily based on similarity.

1 Introduction

Various recommendation systems for media outlet channels such as Youtube, Netflix and Vimeo exist to improve user experience and increase user retention. These are mostly governed by machine learning techniques and tools to constantly learn user interactions and provide the most salient and "clickable" alternatives. The goal of this paper is to build a video recommendation algorithm that returns the 10 most similar Vimeo clips given a clip id based on number of comment, caption, title, number of likes, video thumbnail, and other features. Unlike Youtube's and Netflix's video recommendation engines, the model developed in this report does not make use of user meta data (click-through rate, directed graph of views). However, both recommendation systems were used as a starting point for the algorithms developed in this paper.

1.1 Youtube

In an effort to find personalized recommendations that help users find high quality videos relevant to their interest, Youtube uses a Top-N recommendation system instead of a predictor(Davidson et al., 2010). This is a two step process in which (1) similarities are collected between videos watched and (2) similarities are aggregated to compare those to other videos not yet watched(Adomavicius and Tuzhilin, 2005). Several other recommendation factors that go into consideration in this algorithm such as co-visitation score, and ranking once similarity is calculated. Ranking can be divided into 3 parts: video quality (views, ratings, likes, sharing), diversification(limiting recommendations repeatedly similar, diversifying video types), and user specificity (user history, profile) (Yan et al., 2015). In general, there are two broad classes of data in consideration: (1) Content, and (2) User Activity Data. For the purposes of this report, no data was provided on user activity, therefore, this investigation will be limited to evaluating similarity based on content, which includes video metadata like description, title, etc.

1.2 Netflix

The biggest difference between Netflix and YouTube is that Netflix videos are curated. Moreover, quantity, length, and how Netflix videos are

either episodic in nature or one-time productions differs when compared to YouTube videos. For these reasons, Netflix approached its recommendation algorithms differently.The idea they had was you can't suggest the person 500 videos because of the paradox of choice: having more options makes choices harder (Gomez Uribe and Hunt, 2015). Their algorithms include a Personalized Video Ranking Algorithm, Top-N Ranking Algorithm (like Youtube), Trending Now, and Because you Watched Recommender (which is like a history tracker).

2 Problem Definition

The dataset provided by Vimeo's Recommendation Channel contains no labels for comparing whether or not a video is similar to another. However Vimeo categories are provided as a feature and were used as a proxy for similarity. Although this is intrinsically not a measure of accuracy, Vimeo does categorize videos based on content similarity and thus category is suitable as a proxy. Additionally, once a final model that did return top ten most similar video clips was created, there was no way to tangibly assess its predictive power.

An extension of this investigation would need to take place where user generated input would provide weight updates for features to maximize click through-rate by suggesting optimal similar videos. Essentially, having user data could create a graph of connected nodes whose weights are also updated based on clicks, thus clustering similar videos through edge weights.

2.1 Task Definition and Algorithms

The main challenge of this project was defining similarity. For Youtube and Netflix, similarity is defined by the users (i.e. user activity on the platforms points to videos that could be deemed similar). The Vimeo database, however, did not contain data on user activity. More specifically, each video was represented by 3 categories of features: textual, numerical, and visual. The textual features included title and caption. The numerical features included date of creation, filesize, duration, total comments, total plays, and total likes. The visual feature was a thumbnail of the clip. Therefore, given that the project was limited to video metadata, the notion of similarity had to be objective. The problem boils down to creating a suitable similarity metric for the clips and evaluating each clip based on that metric. Given a target clip, the video recommendation system developed in this report returns the 10 most similar videos in the database based on that similarity metric.

In order to evaluate videos based on their similarity, two models were developed to assess similarity between clips. The first model consisted of creating a similarity score for each pair of clips by focusing on a single feature at a time. In other words, it assessed the suitability of each feature to predict similarity between clips. The second model consisted of representing a clip as a vector in a transformed feature space. This vector would include textual, numerical, and visual information about the clip.

2.2 Model 1: Score-Based Similarity

The main goal of this model was to validate whether or not the three feature sets provided textual, visual, numerical could be used to predict category, our proxy for label. If this could be confirmed, then based on our definition of similarity we could combine these features in Model 2 to calculate a similarity score.

Model 1 consisted of using inter-feature distances between clips to determine proximity and use the top 10 closest clips in n dimensional space and calculating a score dubbed Test Category Match (TCM) to estimate accuracy. This score is the fraction of top ten clips returned with the same category as the original clip. A purity score was also included in this analysis because even though the TCM captures whether or not distance between features are indicators of category, purity captures how many clips are clustered near each other. Therefore, high purity, even for those clips with low TCM, is a good indicator that similarly categorized clips are near each other in n-dimensional space.

2.2.1 Model 1: Numerical Features

Vimeo's database included features of integer nature including total comments, total plays, total likes, filsize, and duration. The only feature to be transformed was "date of creation" which was converted from a 1x6 vector of integers (6-dimensional) to an integer representing the number of days since creation of the clip in the database.

2.2.2 Model 1: Visual Features

In order to embed visual information of a clip's thumbnail in a vector, the team used the ResNet50 neural network, a Residual Neural Network that was pretrained to recognize image features on the ImageNet database, and represent each jpeg image as a 1x2048 vector of floating-point values (He et al., 2016). The images were first represented as a 1x438x780x3 tensor of RGB values and then preprocessed using the keras.applications.resnet50 library in order to be fed into the neural network, which would return the 1x2048 feature representation of the image. As described, this 1x2048 dimensional vector of features was then used to calculate inter-clip distances in 2048 dimensional space.

Note that for numerical and visual features, distance between clips was used to extract top 10 similar clips. For optimization, five different distance metrics were used, those being Euclidean, Manhattan, Chebyshev, Cosine, and Minkowsky.

2.2.3 Model 1: Textual Features

In order to analyze textual similarity between two documents, two different models were tested. The first model used Wordnet, which is a lexical database of English. Wordnet groups words with different position tags into cognitive synonyms (synsets) (pri). These sets are interlinked by means of conceptual-semantic and lexical relations, resulting in a network structure. However, our dataset included a lot of video descriptions in foreign languages, undermining the accuracy of WordNet network. Additional research led us to GenSim, a collection of python scripts, also aimed at finding similarity between documents. GenSim is shown to be very effective for creating unsupervised semantic models from plain text. Our final metrics are based on similarity measurements calculated by using GenSim model, which yields tf-IDF.

To find top 10 similar captions/titles to a caption/title, our first step was to convert the one given to its sparse vector representation. In our case, the representation was a bag-of-words vector. After vectorizing the documents, its tf-idf score was calculated. The similarity between tf-IDF score of given caption/title and other documents in our corpora was extracted by Gensims similarity model, which took other factors than tf-idf score into account as well. The similarity model returned the difference between the scores of original documents and all other documents. We then were able to extract top 10 similar documents by tracking their similarity scores.

2.3 Model 2: Distance-Based Similarity

As briefly mentioned in the Task Definition Section, the main goal of this model was to represent each clip as a vector in an n-dimensional feature space. The vector would include textual, numerical, and visual information about the clip. The main challenge for this Distance-Based Similarity Model was, then, how to represent a clip as an n-dimensional vector. In other words, the issue consisted of embedding textual, numerical, and visual information on the same vector given that this information was provided in different formats (strings, numbers, and jpeg files).

In order to embed the textual, numerical, and visual characteristics of each clip in a vector, a significant amount of data-preprocessing and transformation was conducted. Below are the main steps that were undertaken.

2.3.1 Model 2: Embedding Numerical Information

Embedding numerical information from each clip in a vector was a straightforward undertaking. With the exception of the feature "date of creation", all features were represented as integers (filesize, duration, total comments, total plays, and total likes). The "date of creation" feature was transformed to represent the date of creation of the clip as the number of days since the creation of the first clip in the database. The numerical information from each clip was then represented as a 1x6 (6-dimensional) vector of integers.

2.3.2 Model 2: Embedding Textual Information

In order to embed textual information of a clip in a vector, Latent Semantic Analysis was conducted. First, the title and caption were individually represented as a vector of the TF IDF values of all words present in the title and caption respectively. The titles and captions for the entire dataset were then represented as two m-by-n matrices where m represents the number of clips in the dataset and n represents the number of distinct words in (1) all captions and (2) all titles.

The issue was that such representation (TF IDF) was extremely sparse. Since the purpose of this transformation was to represent textual information in the most concise way possible, a Truncated SVD was conducted in order to reduce the dimensionality of the TF IDF vectors for both title and caption. Such algorithm was conducted with n_components = 100, which is indicated as the ideal parameters for Latent Semantic Analysis. Such parameter explained 13.70% of the variance in the original TF IDF vectors. This value for the explained variance ratio was quite low, as expected, since there is inherently a very large amount of variance between TF IDF vectors for different captions and titles. Therefore, so each caption and text was represented as a 1x112 (112-dimensional) vector of floating-point values.

2.3.3 Model 2: Embedding Visual Information

Just as in Model 1, visual information was extracted by using the ResNet50 neural network to translate jpeg files into 1x2048 feature vectors.

In order to reduce the dimensionality of this feature vector and embed it in the textual and numerical, PCA with n_components = 112 was conducted in order to capture 60% of the variance in the original feature representation (60.03 % of the variance was explained by the model, to be more precise).

2.3.4 Model 2: Final Clip Representation and Similarity assignment

Following the steps highlighted above, each clip was represented as a 1x218 (6+112+100) array of floating point values. t-SNE was finally applied to the dataset in order to project clips on a 2-dimensional space. In this way, clips that had similar features were placed close to each other with very high probability.

In summary, the Distance-Based similarity model consisted of building a feature transformation on a particular dataset that would place similar videos next to one another in the resulting feature space. Given a target video, the recommendation system would simply fetch the 10 closest points in such space and return them as the most similar clips, and thus, the final recommendation.

3 Experimental Evaluation

3.1 Model 1

3.1.1 Model 1: Methodology

To calculate TCM and purity of textual, visual, and numerical features extracted, all clips were compared to each other to calculate different distance and the ten neighbor clips with shortest distance were returned with their respective categories. Thus, TCM was calculated by finding the number of instances the testing clip's category appeared in the top 10. Purity was more simply calculated by finding the category that most appeared in the top 10. As a result, purity is the upper bound to TCM.

3.1.2 Model 1: Results

To maximize TCM, the different distance methods were iterated on when comparing features of two clips yielding the results seen on Figures 1 and 2.

Using distance as a hyper parameter, it was found that the methods that maximized TCM were Minkowsky for Visual Features and Cosine for simpler numerical features. Because text similarity was calculated differently than the rest, GenSim similarities were not tuned. Overall, it was found that Textual representation of the dataset holds the highest predictive power of similarity between clips with a TCM of 37.71%, while the additional numerical dataset holds the highest purity at 49.94%. With these results, it was concluded through Model 1 that our feature set can in fact separate data based on categories and could thus hold the power to separate data based on similarity since these are correlated.

Since the only features provided in the dataset are content-based, this portion of the investigation showed that the most valuable content in identifying similarity is text in the form of captions and titles.

Table 1: Model 1 Results

	TCM Score	Purity	Distance
Visual	16.15%	39.15%	Minkowsky
Numerical	19.18%	49.93%	Cosine
Textual	37.71%	40.81%	NA

3.1.3 Model 1: Discussion

While Euclidean distance is the natural interpretation of distance (also known as the L2 Norm), using four others helped navigate through the n

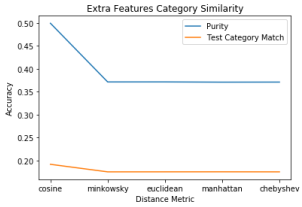


Figure 1: Purity and TCM for Distance Metrics on Numeric Data

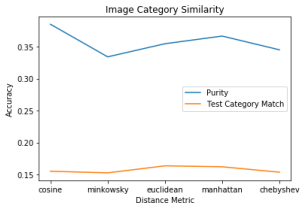


Figure 2: Purity and TCM for Distance Metrics on Thumbnail Data

dimensional feature space with greater curiosity. The reason behind different purities and TCM values is because these metrics are inherently different. Manhattan distance is the L1 norm and in its absolute value lies its unique capability of driving distances to 0, similar to how Lasso Regression works. The Minkowski distance is fact the general L_p norm, where the team chose $p = 3$. A cosine distance was interesting because it does not take into consideration scale, rather, angle between vectors to identify similarity. Therefore it was surprising to see the cosine distance envelop the greater number of clips in the top 10, so there is more to the weights of the features than could be explored.

The team had initially decided that a TCM of over 20% would be sufficient to prove that a set of features could predict a category, our label proxy. The reasoning for 20% was because the dataset provided was limited in size and it made sense to assume that likes and number of comments and a thumbnail would be sufficient to powerfully predict similarity. As can be seen in Table 1, text yielded the highest TCM, which is sound because of the high correlation of caption and title with category.

3.2 Model 2

3.2.1 Model 2: Methodology

In order to validate the Distance-Based Similarity model, two procedures were conducted. First, the exploded feature space was plotted and tested for different parameters of the t-SNE model (perplexity and number of iterations). Second, the same procedure used for validating model one (calculating purity and test category matches) was conducted for model 2.

3.2.2 Model 2: Results

In order to make sure that the output of the t-SNE model made sense, that is, similar points would be placed next to one another and reasonable clusters would be formed, different values were tested for the perplexity and iteration parameters of the model.

First, $p \in \{1, 10, 30, 50, 100, 200, 500, 1000\}$ were the values tested for the perplexity of the model. These tests were run with iterations set to 300. Figure 3 shows the results of such model. As can be seen, with p=1 (a), the t-SNE model focuses too much on local structure. As a consequence, the points are consider relatively distinct one from the other and there is no clear formation of clusters. At the other end of the spectrum, with p=1000, the model focuses too much on global structure, all points are deemed similar, and only a single cluster is formed. At p=30, it was found that the model provided a good balance between focus on both local and global structure.

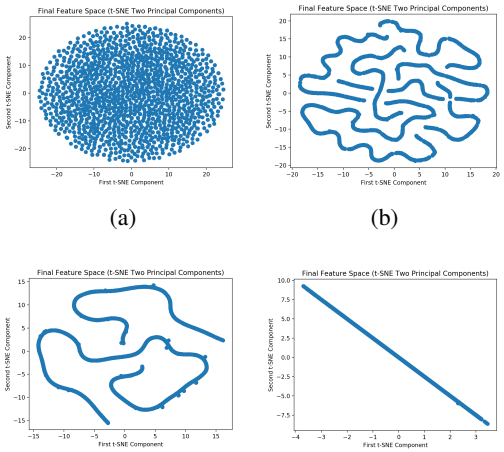
Second, $i \in 300, 600, 1000, 2000, 5000$ were the values tested for the iteration parameter. By varying the number of iterations, the goal was to identify at what point the t-SNE model would reach equilibrium. Figure 4 shows the results for different iteration parameters tested. As can be seen, the model only reached equilibrium with i = 5000.
Finally, the same purity and test category match (TCM) procedure highlighted for Model 1 was conducted for Model 2. The results can be seen in Table 2.

Table 2: Model 2 Results

TCM Score	Purity Score	Distance Metric
25.87%	37.15%	Euclidean

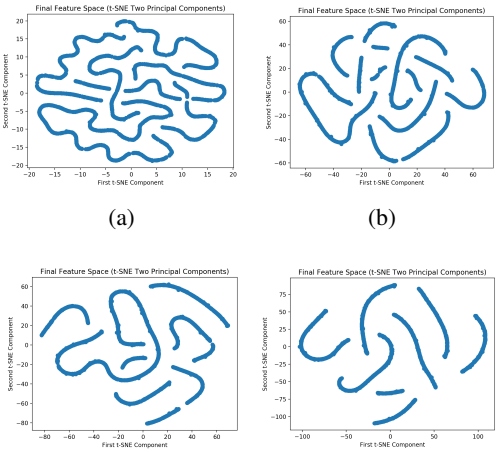
3.2.3 Model 2: Discussion

Based on the use of categories as proxies for labels, or ideal similarity, it is interesting to note that the number of clusters outputted by the t-SNE model in Figure 4(d) (clusters = 8) is very similar to the number of main categories in the dataset (categories = 10). Although the model was not



(a) (b) (c) (d)

Figure 3: t-SNE Feature Space for 300 iterations at different perplexities (a) perplexity = 1 (b) perplexity = 30 (c) perplexity = 100 (d) perplexity = 1000



(a) (b) (c) (d)

Figure 4: t-SNE Feature Space for perplexity = 30 at different iterations (a) perplexity = 1 (b) perplexity = 30 (c) perplexity = 100 (d) perplexity = 1000

trained to match categories of target and recommended clips, this is a good indication that the model is narrowing down its definition of similarity. It is important to remember here that the concept of similarity is vague. Since this project only works with clip metadata, similarity could mean anything from similar number of likes to similar date of creation, to similar caption text. The bottom line is that the transformations applied to the original feature set were adequate to group videos that had similar features together.

4 Future Work

An important future direction is to not only map full sentences and whole images into the same space, but also to map phrases within sentences and salient regions within images to a shared visual-textual space to achieve a better discriminative ability for image/text representations. This representation is less likely to miss details such as color and environment, which might convey critical visual cues for matching images and texts. In a similar fashion, mapping not only the full sentences but also the phrases within the sentences into a common space can yield to a better representation. Inspired by recent advances in multi modal learning, some recent work (Niu et al., 2017) has focused on multimodal neural network models to exploits the hierarchical relations between sentences and phrases, and between whole images and image regions Another recent paper uses the multimodal representation to create a novel language model for decoding distributed representations Dual-path Convolutional Networks are also being used to extract more meaningful image and text representations We believe that building representations for all phrases and image regions, and exploiting their hierarchical relations can produce the general performance of image-text similarity matching problem, thus improving our similarity index of TCM.

5 Conclusion

This report proposed a framework to estimate similarity between video clips based solely on clip metadata. The score-based model assessed similarity based on a single defining feature. As expected, text data (caption and title) was the most accurate feature in assessing similarity between clips. The distance-based model adopted a holistic approach to assessing similarity by transforming the original feature space into a 2-dimensional feature space where similar videos would be placed closely together with high probability. The second model performed slightly worse than the first, although improvements could be made in terms of generating similarity features for textual data (TF-IDF representation was used instead of the GenSins framework). Going forward, the integration of user data would bolster the effectiveness of the clip recommendation system. The problem would then become a supervised one and similar-

ity would be defined by the users themselves.

References

Gediminas Adomavicius and Alexander Tuzhilin. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowl. and Data Eng.*, pages 734–749.

James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Michel Lambert, Blake Livingston, and Dasarathi Sampath. 2010. The youtube video recommendation system. pages 293–296.

Carlos A. Gomez Uribe and Neil Hunt. 2015. The netflix recommender system: Algorithms, business value, and innovation. *ACM Trans. Manage. Inf. Syst.*

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhenxing Niu, Mo Zhou, Le Wang, Xinbo Gao, and Gang Hua. 2017. Hierarchical multimodal lstm for dense visual-semantic embedding. *2017 IEEE International Conference on Computer Vision (ICCV)*.

Ming Yan, Jitao Sang, and Changsheng Xu. 2015. Unified youtube video recommendation via cross-network collaboration. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, ICMR '15, pages 19–26. ACM.