UNIFESP at MediaEval 2016: Predicting Media Interestingness Task

Jurandy Almeida
GIBIS Lab, Institute of Science and Technology, Federal University of São Paulo – UNIFESP
12247-014, São José dos Campos, SP – Brazil
jurandy.almeida@unifesp.br

ABSTRACT

This paper describes the approach proposed by UNIFESP for the MediaEval 2016 Predicting Media Interestingness Task and for its video subtask only. The proposed approach is based on combining learning-to-rank algorithms for predicting the interestingness of videos by their visual content.

1. INTRODUCTION

Current solutions to predict the interestingness of video data are usually based on learning-to-rank strategies. Most of those research works have focused on using a single machine-learned ranker. Recently, combining individual predictions from a set of machine-learned rankers has been established as an effective way to improve classification performance [9].

This paper presents an approach for predicting the interestingness of videos that relies on different learning-to-rank strategies for processing visual contents. For that, a simple, yet effective, *histogram of motion patterns* (HMP) [1] is used for processing visual information. Then, a simple *majority voting* scheme [8] is used for combining machine-learned rankers and predicting the interestingness of videos.

This work is developed in the MediaEval 2016 Predicting Media Interestingness Task and for its video subtask only, whose goal is to automatically select the most interesting video segments according to a common viewer by using features derived from audio-visual content or associated textual information. Details about data, task, and evaluation are described in [4].

2. PROPOSED APPROACH

Measuring the degree of interestingness of a video is a challenging task. For that, the strategy proposed by Jiang et al. [6] was adopted. It relies on training a model to compare the interestingness of video pairs. Thus, given two videos to the system, it indicates the more interesting one.

Roughly speaking, the basic idea is to use machine learning algorithms to learn a ranking function based on features extracted from training data, and then apply it to features extracted from testing data.

The proposed approach predicts the interestingness of videos based on combining learning-to-rank algorithms and exploiting only visual information.

MediaEval 2016 Workshop, Oct. 20-21, 2016, Hilversum, Netherlands.

2.1 Visual Features

Instead of using any keyframe visual features provided by the organizers, a simple and fast algorithm was used to encode visual properties, known as *histogram of motion patterns* (HMP) [1]. It considers the video movement by the transitions between frames. For each frame of an input sequence, motion features are extracted from the video stream. After that, each feature is encoded as a unique pattern, representing its spatio-temporal configuration. Finally, those pattens are accumulated to form a normalized histogram.

2.2 Learning to Rank Strategies

In this work, the extracted features were classified with the following methods:

Ranking SVM [7]. It is a pairwise ranking method that uses the traditional SVM classifier to learn a ranking function. For that, each query and its possible results are mapped to a feature space. Next, a given rank is associated to each point in this space. Finally, a SVM classifier is used to find an optimal separating hyperplane between those points based on their ranks.

RankNet [2]. It is a pairwise ranking method that relies on a probabilistic model. For that, pairwise rankings are transformed into probability distributions, enabling the use of probability distribution metrics as cost functions. Thus, optimization algorithms can be used to minimize a cost function to perform pairwise rankings. The authors formulate this cost function using a neural network in which the learning rate is controlled with gradient descent steps.

RankBoost [5]. It is a pairwise ranking method that relies on boosting algorithms. Initially, each possible result for a given query is mapped to a feature space, in which each dimension indicates the relative ranking of individual pairs of results, i.e., whether one result is ranked below or above the other. Thus, the ranking problem is formulated as a binary classification problem. Next, a set of weak rankers are trained iteratively. At each iteration, the resulting pairs are re-weighted so that the weight of pairs ranked wrongly is increased whereas the weight of pairs ranked correctly is decreased. Finally, all the weak rankers are combined as a final ranking function.

ListNet [3]. It is an extension of RankNet that, instead of using pairwise rankings, considers all possible results for a given query as a single instance, enabling to capture and exploit the instrinsic structure of the data.

Majority Voting [8]. It is the simplest method for combining the output of a set of classifiers. It relies on assigning the class of a given result by the most common class assigned by all the classifiers.

3. EXPERIMENTS & RESULTS

Five different runs were submitted for the video sub-task. These runs were configured as shown in Table 1. As the proposed approach relies on combining different learning-to-rank algorithms, one of the runs considers a fusion of machine-learned rankers. For comparison purposes, the use of each machine-learned ranker in isolation was evaluated in the other runs. All those approaches were calibrated through a 4-fold cross validation on the development data.

Table 1: Configurations of Runs

Run	Learning-to-Rank Strategy
1	Ranking SVM
2	RankNet
3	RankBoost
4	ListNet
5	Majority Voting

The development data was used for training and is composed by 5,054 video segments from 52 movie trailers. Each video segment was represented by a HMP. Notice that only the visual content was considered, ignoring audio information and textual metadata. Then, the extracted features were used as input to train the aforementioned machine-learned rankers. The SVM^{rank} package[1] [7] was used for running Ranking SVM. The RankLib package[2] was used for running RankNet, RankBoost, and ListNet. Ranking SVM was configured with a linear kernel. RankNet, RankBoost, and ListNet were configured with their default parameter settings. Next, the trained rankers were used to predict the rankings of test video segments. The rankings associated with the video segments of a same movie trailer were normalized using a z-score normalization. After that, a thresholding method was applied to transform the normalized rankings into binary decisions. It was found empirically that better results were obtained when a video segment is classified as interesting if its normalized rank is greater than 0.7; otherwise, it is classified as non interesting. Finally, the binary decisions of all the rankers are combined using a *majority voting* scheme, producing the final classification. The effectiveness of each strategy was assessed using Mean Average Precision (MAP).

Table 2 presents the results obtained on the development data. Observe that the performance of the different learning-to-rank algorithms in isolation is similar, with a small advantage to Ranking SVM. By analyzing the confidence intervals, it can be noticed that the results achieved by the fusion of all the machine-learned rankers seem promising.

Table 2: Results obtained on the development data.

Run	Avg.	Conf. Interval (95%)	
		min.	*max.*
1	15.19	13.99	16.38
2	13.82	12.09	15.55
3	14.67	12.93	16.42
4	13.32	12.06	14.57
5	14.71	12.69	16.73

[1] `https://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html` (As of September 2016)

[2] `https://sourceforge.net/p/lemur/wiki/RankLib/` (As of September 2016)

Table 3 presents the results obtained on the official submission runs for 2,342 video segments from 26 movie trailers of the test data. Observe that the best results were achieved by a learning-to-rank algorithm in isolation, more specifically, Ranking SVM. It can be noticed that the fusion of all the machine-learned rankers did not improved the overall performance. One of the reasons for those results is the strategy adopted for combining learning-to-rank algorithms, in which all of them are treated equally.

Table 3: Results of the official submitted runs.

Run	Learning-to-Rank Strategy	MAP (%)
1	Ranking SVM	18.15
2	RankNet	16.17
3	RankBoost	16.17
4	ListNet	16.56
5	Majority Voting	16.53

Figure 1 presents the Average Precision (AP) per movie trailer achieved in each of the submitted runs. Although the MAP obtained for the fusion of all the machine-learned rankers is not superior to each of them in isolation, the obtained results show the potential of the idea. Notice that Ranking SVM provides the best results for 8 movie trailers, RankNet was the best for 8 movie trailers, and RankBoost performs better than both of them in 7 movie trailers. This clearly indicates that those learning-to-rank algorithms provide complementary information that can be combined by fusion techniques aiming at producing better results.

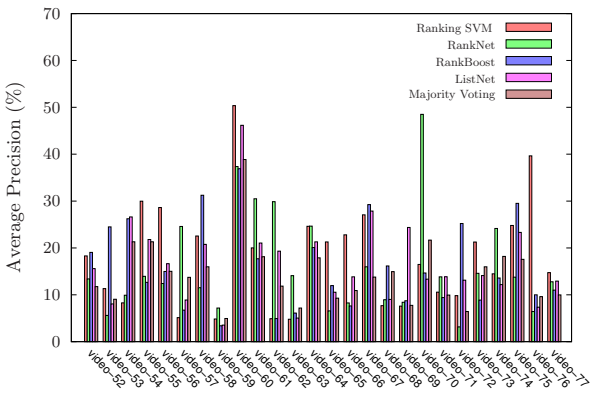

Figure 1: AP per movie trailer achieved in each run.

4. CONCLUSIONS

The proposed approach has explored only visual properties. Different learning-to-rank strategies were considered, including a fusion of all of them. Obtained results demonstrate that the proposed approach is promising. Future works include the investigation of a smarter strategy for combining learning-to-rank algorithms and considering other information sources to include more features semantically related to visual content.

5. ACKNOWLEDGMENTS

The author would like to thank CAPES, CNPq, and FAPESP (grant #2016/06441-7) for funding.

6. REFERENCES

[1] J. Almeida, N. J. Leite, and R. S. Torres. Comparison of video sequences with histograms of motion patterns. In *ICIP*, pages 3673–3676, 2011.

[2] C. J. C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. N. Hullender. Learning to rank using gradient descent. In *ICML*, pages 89–96, 2005.

[3] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li. Learning to rank: from pairwise approach to listwise approach. In *ICML*, pages 129–136, 2007.

[4] C.-H. Demarty, M. Sjöberg, B. Ionescu, T.-T. Do, H. Wang, N. Q. K. Duong, and F. Lefebvre. Mediaeval 2016 predicting media interestingness task. In *Proc. of the MediaEval 2016 Workshop*, Hilversum, Netherlands, Oct. 20–21 2016.

[5] Y. Freund, R. D. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4:933–969, 2003.

[6] Y.-G. Jiang, Y. Wang, R. Feng, X. Xue, Y. Zheng, and H. Yang. Understanding and predicting interestingness of videos. In *AAAI*, pages 1113–1119, 2013.

[7] T. Joachims. Training linear svms in linear time. In *ACM SIGKDD*, pages 217–226, 2006.

[8] L. Lam and C. Y. Suen. Application of majority voting to pattern recognition: an analysis of its behavior and performance. *IEEE Trans. Systems, Man, and Cybernetics, Part A*, 27(5):553–568, 1997.

[9] L. T. Li, D. C. G. Pedronette, J. Almeida, O. A. B. Penatti, R. T. Calumby, and R. S. Torres. A rank aggregation framework for video multimodal geocoding. *Multimedia Tools and Applications*, 73(3):1323–1359, 2014.