Content-Based Video Relevance Prediction with Second-Order Relevance and Attention Modeling*

Xusong Chen, Rui Zhao, Shengjie Ma, Dong Liu, Zheng-Jun Zha
CAS Key Laboratory of Technology in Geo-Spatial Information Processing and Application System
University of Science and Technology of China
Hefei, China

ABSTRACT

This paper describes our proposed method for the Content-Based Video Relevance Prediction (CBVRP) challenge. Our method is based on deep learning, i.e. we train a deep network to predict the relevance between two video sequences from their features. We explore the usage of second-order relevance, both in preparing training data, and in extending the deep network. Second-order relevance refers to e.g. the relevance between x and z if x is relevant to y and y is relevant to z. In our proposed method, we use second-order relevance to increase positive samples and decrease negative samples, when preparing training data. We further extend the deep network with an attention module, where the attention mechanism is designed for second-order relevant video sequences. We verify the effectiveness of our method on the validation set of the CBVRP challenge.

CCS CONCEPTS

• **Information systems → Recommender systems**;

KEYWORDS

attention mechanism; content-based filtering; deep learning; video relevance prediction.

ACM Reference Format:
Xusong Chen, Rui Zhao, Shengjie Ma, Dong Liu, Zheng-Jun Zha. 2018. Content-Based Video Relevance Prediction with Second-Order Relevance and Attention Modeling. In *MM '18: 2018 ACM Multimedia Conference, Oct. 22–26, 2018, Seoul, Republic of Korea.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3240508.3266434

1 INTRODUCTION

With the rapid development of Internet, watching videos online has become one of the most indispensable entertainments in our daily life. This leads to a rapid growth of demand for online videos, and many online video streaming service, such as YouTube, Netflix, and Hulu, depend heavily on the video recommendation system to help its user discover videos they would enjoy. Video recommendation task is still a difficult task due to the huge gap between the tremendous and multifarious online videos and users' personalized interests. Most existing video recommendation methods usually fall into one of the three categories: collaborative filtering , content-based filtering, and hybrid methods. Collaborative filtering recommendation systems compute the video relevance based on users' explicit or implicit feedback, e.g. watch and search behaviors. The system analyze the user-to-video preference and compute the video-to-video relevance scores using collaborative filtering based methods [3, 8, 10–12]. However, collaborative filtering methods suffer from the *cold-start* problem. One promising approach to solve cold-start is exploiting video content for relevance prediction, i.e. we can predict the video relevance by analyzing the content of videos including image pixels, audios, subtitles and metadata. Since the content contains almost all the information about a video, ideally, we can have enough details to build the video relevance table only from video content.

Content-based methods focus on recommending items which have similar content characteristics to the items the user liked in the past. For most existing systems, the content features are associated with the items as structured metadata, e.g. movie/show genre, director/actors, description; Or other unstructured information from external sources, such as tags, and textual reviews. In contrast to these kinds of *explicit* features, there are also *implicit* content characteristics which can be exploited from the original movie/show video. Such characteristics could be visual features encoding low-level information like lighting, color, shape, motion, or high-level semantics like plot, mood, and artistic style.

Li *et al.* [5] represent a complicated TV-show from different modalities, i.e., audio, meta-data and vision features in Hulu-ICIP 2017 challenge. Mei *et al.* [7] proposed contextual video recommendation system, called VideoReach, based on multimodal content relevance and user feedback. They consider online video usually consists of different modalities (i.e. visual information (color, motion, shot tempo, concept), textual information (term frequency describe the weight of a word), acoustic information (strong beats)). Zhu *et al.* [13] proposed content-based recommendation framework, called VideoTopic, which uses a topic model to represent both visual and textual content of videos, and links user interests and video content by estimating user interests using the topic distributions of user watched videos. Deldjoo *et al.* [2] proposed content-based movie recommender system using stylistic visual features, which consists of lighting, color and motion.

Based on the above analyses, Hulu organizes a content-based video relevance prediction competition to explore efficient ways for video recommender system to deal with the cold-start problem.

*Corresponding author: Dong Liu (dongeliu@ustc.edu.cn).

MM '18, October 22–26, 2018, Seoul, Republic of Korea

ACM ISBN 978-1-4503-5665-7/18/10…$15.00
https://doi.org/10.1145/3240508.3266434

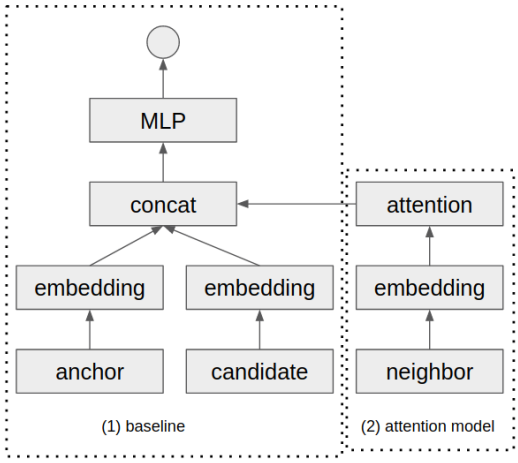


Figure 1: The deep network structure in our proposed method.

In this paper, we propose a deep neural network with second-order relevance and attention modeling to predict video relevance. The main contributions are summarized as follows:

- We explore the usage of second-order relevance to increase positive samples and decrease negative samples during preparing training data;
- We further extend the deep network with an attention module, where the attention mechanism is designed for capturing second-order relevance;
- Extensive experimental results on both track 1 and track 2 demonstrate the superior performance of the proposed methods compared with [6].

2 DATA AND PROTOCOL

There are three sets of items, i.e., training set $\mathcal{R}_{train}$, validation set $\mathcal{R}_{val}$, and testing set $\mathcal{R}_{test}$. For training set and validation set in both tracks, the organizers provide the ground truth (relevance lists) derived from implicit viewer feedbacks. Specifically, for each item r, they provide the ground truth top m most relevant items retrieved from candidate set $\mathcal{C}$. The relevance list of item v is defined as $r(v) = [r_1^v, r_2^v, ..., r_m^v]$, where $r_i^v \in \mathcal{C}$ is the item randed i-th position in $r(v)$.

For each item (i.e., TV-show/Movie) $r \in \mathcal{R}$, the organizers provide two kind of visual feature. Specifically, for frame-level features, they sample 8 frames per second and then feed the decoded frames into the Inception V3 networks [1] trained on ImageNet dataset, and fetch the ReLU activation of the last hidden layer, called *inception-pool3* feature with 2048 dimensions. Then we further perform p-norm pooling on frame features as the final signature. Specifically, the final signature $\bar{x}$ is calculated by:

$$\bar{x} = \left(\frac{1}{F} \sum_{i}^{F} |x_i|^p \right)^{1/p} \tag{1}$$

where F is the number of frames, $x_i \in \mathbb{R}^{2048}$ is the inception feature. Here p is set to 2. For video-level features, they also employ the state-of-the-art architectures - C3D models, resorting to its most popular implementation from Facebook [9] respectively. Here they sample 8 frames per second and feed the frame stream into the model trained on Sports1M dataset [4], and fetch the activations of pool5 layer with 512 dimensions as the final video clip feature, called *c3d-pool5* feauture.

We need to predict top K relevant shows/movies for each item, which can be represented as $\widetilde{r(v)} = [\widetilde{r_1^v}, \widetilde{r_2^v}, ..., \widetilde{r_K^v}]$. The submission results will be evaluated based on recall and hit rate regarding to top K prediction. Based on above formulation, the metric *recall@K* and *hit@K* can be calculated as following:

$$recall@K = \frac{|r(v) \cap \widetilde{r(v)}|}{|r(v)|} \tag{2}$$

$$hit@K = \begin{cases} 1, \text{ if } recall@K > 0 \\ 0, \text{ otherwise} \end{cases} \tag{3}$$

In the end, we report the average of recall and hit on all test cases respectively as the final performance.

3 OUR PROPOSED METHODS

The competition task is to learn a model that can compute the relevance score among TV-Shows/Movies from the video contents and its features. In this section, we introduce our proposed method for content based video relevance prediction.

3.1 Data sampling

3.1.1 Negative sampling. In baseline work [6], the items which do not appear in relevance list of *anchor* are considered as negative samples. This is unreasonable because relevant or not relevant is subjectively determined by the organizer according to the user historical behavior. If we define relevance list of *anchor* as first-order relevance list of *anchor*, we naturally extend to second-order relevance list. Given the first-order relevance list $r(v)$, we define second-order relevance list of item v is $r(r(v)) = [r(r_1^v), r(r_2^v), ..., r(r_m^V)]$ (the relevance list of item v can be seen as first-order relevance list). Obviously, the items appear in second-order relevance list of item v are more similar to item v than others which do not appear in second-order relevance list. Therefore, our negative samples are sampling from the items which do not appear in first-order and second-order relevance list. Our negative sampling method reduces the scope of negative samples.

3.1.2 Expand positive samples. Considering that the items which appear the second-order relevance list have a high probability than items which do not appear (except first-order relevance list), we pick some samples with high confidence which appear in second-order relevance list to expand positive samples. For item v, we can get its seond-order relevance list $r(r(v)) = [r(r_1^v), r(r_2^v), ..., r(r_m^v)]$. If there is a item appearing in its second-order relevance list more than $m/2$ times, we expand the item into positive samples.

3.2 Model overview

We define each query as the *anchor*, all other items as the *candidate*. We use our baseline framework to predict relevance between *anchor*

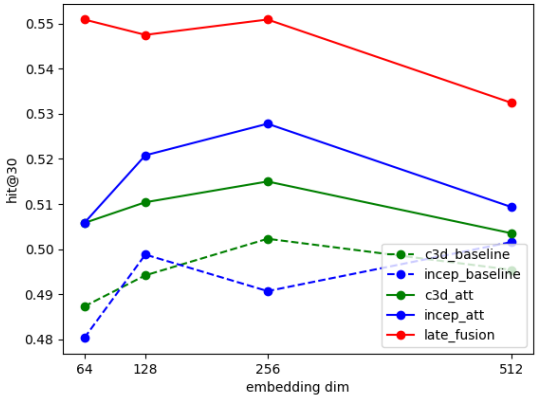
Table 1: Results on the validation set of different data sampling strategies. B stands for baseline (naive sampling), N stands for decreasing negative samples, P stands for increasing positive samples.

(a) Track 1

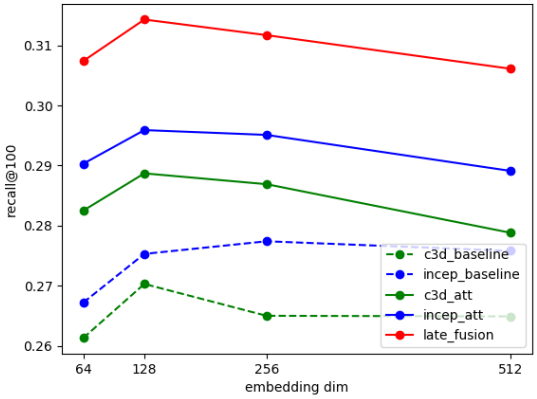
Feature	Sampling	hit@30	recall@100
C3D	B	0.5035	0.2581
	B+N	0.4988	0.2613
	B+P	0.5081	0.2664
	B+N+P	0.5000	0.2666
Inception	B	0.5255	0.2675
	B+N	0.5174	0.2688
	B+P	0.5012	0.2753
	B+N+P	0.5093	0.2757
Late fusion	B	**0.5370**	0.2837
	B+N	0.5289	0.2820
	B+P	0.5312	0.2914
	B+N+P	0.5289	**0.2918**

(b) Track 2

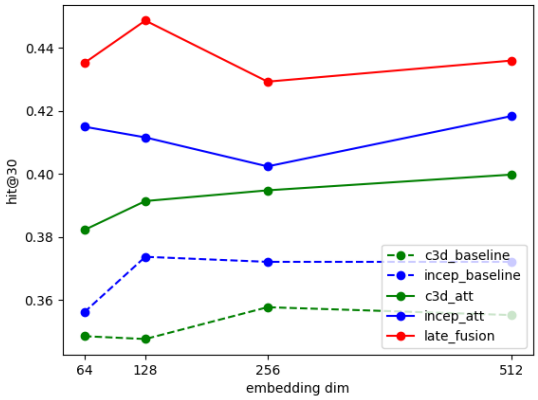
Feature	Sampling	hit@30	recall@100
C3D	B	0.3460	0.1864
	B+N	0.3603	0.1875
	B+P	0.3670	0.1944
	B+N+P	0.3561	0.1936
Inception	B	0.3552	0.1863
	B+N	0.3603	0.1877
	B+P	0.3695	0.1995
	B+N+P	0.3822	0.2031
Late fusion	B	0.3788	0.2089
	B+N	0.3973	0.2101
	B+P	0.4049	0.2180
	B+N+P	**0.4158**	**0.2198**



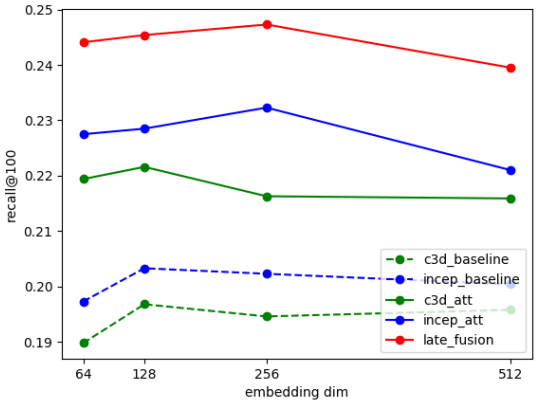
(a) Track 1: hit@30



(b) Track 1: recall@100



(c) Track 2: hit@30



(d) Track 2: recall@100

Figure 2: Results on the validation set of our proposed method with and without attention modeling.

and *candidate*, which is illuminated in Figure 1. We denote the original features of *anchor* and *candidate* for each show/movie trailer as x_a and x_c respectively. We feed x_a and x_c into our baseline model to predict video relevance. To utilize second-order relevance, if available, we also take the relevant items of *candidate* into account to enhance results. More details are shown as follows.

3.2.1 Baseline network. According to the ground truth given, the only supervision is the relationship between two videos, that is, relevant or not relevant. In our baseline methods, we take it as binary classification task for relevance learning. We first construct two embedding function f_a and f_c for *anchor* and *candidate* respectively, which embeds x_a and x_c into d-dimensional space using a single fully-connection layer. Finally, we concatenate them and then feed it into similarity function D. Therefore, the *log* loss function can be represented as:

$$-\sum_{a,c} y_c log\sigma(D(f_a(x_a), f_c(x_c))) + (1-y_c) log(1-\sigma(D(f_a(x_a), f_c(x_c)))) \tag{4}$$

where $y_c \in \{0, 1\}$ is the ground truth that indicates the *candidate* item is relevant to the *anchor* item or not, and $D(\cdot)$ is the similarity function to compute relevance between *anchor* item and *candidate* item, which is implemented by two-layer perceptron. Here $\sigma(\cdot)$ is *sigmoid* function.

3.2.2 Attention modeling. We denote the feature of relevant items to *candidate* as $\vec{x}_n = [x_{n1}, x_{n2}, ..., x_{nm}]$, where m is the number of neighbor nodes (i.e. relevance list). We first adopt embedding function f_n to embeds $\vec{x}_n$ into $\mathbb{R}^{m \times d}$ space. Then we adopt a two-layer perceptron to implement the attention:

$$\vec{\alpha}_{ni}^{(1)} = f(\vec{W}^{(1)} x_{ni} + \vec{b}^{(1)}), \tag{5}$$

$$\alpha_{ni}^{(2)} = W^{(2)} \vec{\alpha}_{ni} + b^{(2)}, \tag{6}$$

where $\vec{W}^{(1)}$ and $\vec{b}^{(1)}$ denote the weight matrix and the bias vector for the first layer, and the $W^{(2)}$ and $b^{(2)}$ denote the weight vector and the bias for the second layer. $f(\cdot)$ is set to the ReLU function.

The final weights are obtained by normalizing the above attentive scores using *softmax* function, specifically, we have $\alpha_{ni} = \frac{\exp(\alpha_{ni}^{(2)})}{\sum_{i=1}^{m} \exp(\alpha_{ni}^{(2)})}$. Then, attention vector $\hat{x}_n \in \mathbb{R}^d$ is computed by $\sum_{j=1}^{m} \alpha_{ni} \odot x_{ni}$, where $\odot$ is element-wise product with broadcasting. The first-order relevance list of *candidate* can be seen as the second-relevance list of *anchor*. We take it into account to improve our performance.

3.2.3 Fusing baseline network and attention modeling network. It is worth noting that the relevant items to *candidate* are not always available to us. In the testing phase, we need to predict the relevance between any two items no matter they are in the training, validation, or test set. If the *candidate* is in the training or validation set, we have its relevant items, but if it is in the test set, we do not have. So we propose a fusing procedure as follows. If the *candidate* is in the training or validation set, we use both the baseline network (without attention modeling) and the enhanced network (with attention modeling) to predict two relevance scores, and combine the two scores with $s = a \times s_{baseline} + b \times s_{attention}$, where $a = 0.3, b = 0.7$ for Track 1 and $a = 0.35, b = 0.65$ for Track 2. If the *candidate* is in the test set, we simply use the baseline network to predict one relevance score, and regard that as the final score. Finally, we rank all *candidates* for each *anchor* and retrieve the top-k items.

Table 2: Results on the test set (courtesy of challenge organizers).

Method	Track 1		Track 2	
	hit@30	recall@100	hit@30	recall@100
[6]	**0.525**	0.141	0.373	0.116
Ours	0.516	**0.187**	**0.400**	**0.147**

4 EXPERIMENTAL RESULTS

We perform experiments using the training set to train and the validation set to test, so as to verify the effectiveness of the proposed method. Results are summarized in Table 1 and Figure 2.

In Table 1a, we first verify the proposed data sampling strategies. In this test, we use the basic deep network, i.e. without attention modeling. The feature embedding dimension is 256. It can be observed that for both tracks and both kinds of features (C3D and Inception), our proposed "decreasing negative samples" and "increasing positive samples" strategies are consistently better than the baseline (naive sampling). For both tracks, the best results on recall@100 are achieved by using the two strategies together and late fusion.

In Figure 2, we further verify the proposed attention modeling. We compare the two networks with and without attention modeling, using both kinds of features (C3D and Inception), and set the feature embedding dimension to 64, 128, 256, 512, respectively. In this test, we use the proposed data sampling strategies to prepare training data. It can be observed that the network with attention modeling performs consistently better than that without. Thus, we perform late fusion of the two networks having attention modeling, and the fused results are even better (shown as red lines).

Based on the above results, our final proposal for the challenge is configured as follows: we use both the training set and the validation set to train, and adopt the proposed data sampling strategies to prepare training data; we train two networks, both with attention modeling and the feature embedding dimension is 256, for the two kinds of features, and perform late fusion. Our achieved results on the test set are shown in Table 2.

5 CONCLUSION

We have presented our proposed method for the content-based video relevance prediction task. On top of a deep network model that is trained to predict the relevance between two video sequences, our method features the mining of second-order relevance both in preparing training data and in extending the deep network with an attention module. Experimental results show the effectiveness of the proposed method.

ACKNOWLEDGMENT

This work was supported by the Natural Science Foundation of China under Grants 61772483, 61331017, 61390512, 61622211, 61472392 and 61620106009, and by the Fundamental Research Funds for the Central Universities under Grant WK2100100030.

REFERENCES

[1] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. 2016. Youtube-8m: A large-scale video classification benchmark. arXiv preprint arXiv:1609.08675. (2016).

[2] Yashar Deldjoo, Mehdi Elahi, Paolo Cremonesi, Franca Garzotto, Pietro Piazzolla, and Massimo Quadrana. 2016. Content-based video recommendation system based on stylistic visual features. *Journal on Data Semantics* 5, 2 (2016), 99–113.

[3] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *IEEE International Conference on Data Mining*. 263–272.

[4] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. 2014. Large-scale video classification with convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*. 1725–1732.

[5] Yan Li, Hanjie Wang, Hailong Liu, and Bo Chen. 2017. A study on content-based video recommendation. In *IEEE International Conference on Image Processing*. IEEE, 4581–4585.

[6] Mengyi Liu, Xiaohui Xie, and Hanning Zhou. 2018. Content-based Video Relevance Prediction Challenge: Data, Protocol, and Baseline. arXiv preprint arXiv:1806.00737. (2018).

[7] Tao Mei, Bo Yang, Xian-Sheng Hua, and Shipeng Li. 2011. Contextual video recommendation by multimodal relevance and user feedback. *ACM Transactions on Information Systems* 29, 2 (2011), 10.

[8] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*. ACM, 285–295.

[9] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2015. Learning spatiotemporal features with 3D convolutional networks. In *IEEE International Conference on Computer Vision*. 4489–4497.

[10] Chong Wang and David M Blei. 2011. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 448–456.

[11] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1235–1244.

[12] Yin Zheng, Bangsheng Tang, Wenkui Ding, and Hanning Zhou. 2016. A neural autoregressive approach to collaborative filtering. In *International Conference on Machine Learning*.

[13] Qiusha Zhu, Mei-Ling Shyu, and Haohong Wang. 2013. Videotopic: Content-based video recommendation using a topic model. In *IEEE International Symposium on Multimedia*. IEEE, 219–222.