A STUDY ON CONTENT-BASED VIDEO RECOMMENDATION

Yan Li *Hanjie Wang* *Hailong Liu* *Bo Chen*

Tencent WeChat, China
{*rockyanli, hankinwang, hailongliu, jennychen*}@*tencent.com*

ABSTRACT

Video streaming services heavily depend on the video recommender system to help the users discover videos they would enjoy. Most existing recommender systems compute video relevance based on users implicit feedbacks, e.g., watch and search behaviors. For example, one can use Collaborative Filtering based methods to model the user-video preference, and compute the video-video relevance scores. However, when a new coming video is added to the library, the recommender system has to deal with the cold-start problem, i.e., to bootstrap the video relevance score with very few user behavior with respect to the newly added video. To solve this problem, in this paper we propose a content-based video recommendation approach by taking the advantage of deep convolutional neural networks to alleviate the cold-start problem. The proposed approach works well, especially in the case of serious data incompleteness. In addition to the vision feature, we also conduct extensive evaluation on video meta-data, and audio features.

Index Terms— Content-Based Video Recommendation, Recommender System, Deep Convolutional Neural Networks, Synthetic Anchor

1. INTRODUCTION

Video relevance prediction is a promising research direction with increasing demands, especially in the era of Internet multimedia considering huge body of video data (e.g., various types of videos can be found on the video sharing sites like Hulu, such as movies, newscasts, sitcoms, sports, and commercials, etc). Given the relevance of videos and the user watch/search behaviors, recommender system can provide personalized recommendations, which helps users to discover more contents of interests (please kindly refer to Figure 1). Recommender systems typically produce a list of recommendations in one of two ways, i.e., through collaborative filtering [1] [2] [3] or content-based filtering, where collaborative filtering approaches building a model from a user's past behaviour (items previously purchased or selected and/or numerical ratings given to those items) as well as similar decisions made by other users. This model is then used to predict items (or ratings for items) that the user may have an interest in, while content-based filtering approaches utilize a series of discrete characteristics of an item in order to recommend additional items with similar properties.

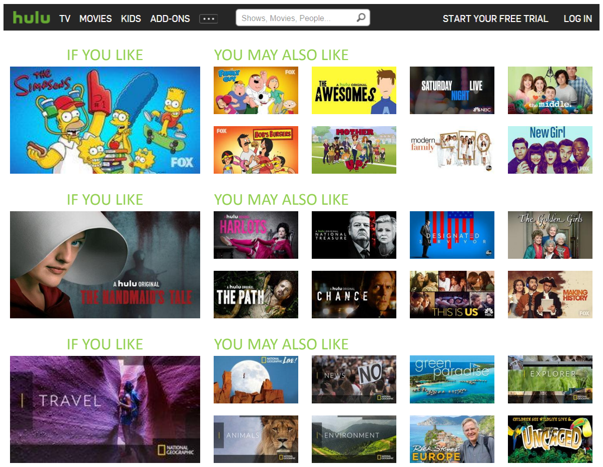


Fig. 1. Three real TV recommendation cases in HULU, i.e., the three host TV shows are *The Simpsons*, *The Handmaid's Tale*, and *National Geographic Travel*, respectively.

For most online services, the computation of video relevance is based on the users behavior. For example, given the watch behaviors of users, collaborative filtering can be used to model the interaction between users and videos, then the features of videos from collaborative filtering can be used to compute the video relevance. However, the problem of *cold-start* limits the development of such collaborative filtering based method, as recommender systems often require a large amount of existing data on a user in order to make accurate recommendations. To solve such problem, computing video relevance from the content directly, instead of as a byproduct from matrix factorization of user behaviors, could be a natural choice. Since the content of videos contains almost all the information about a video (i.e., pixels, audios, subtitles, and meta-data), ideally, we can have enough information to build video relevance table only from contents. By doing this, we do not need the user behaviors for a new coming video.

Based on the above analysis, Hulu organizes a content-based video relevance prediction competition to explore efficient ways for video recommender system to alleviate the *cold-start* problem. In this paper, we propose to represent video data with features from different modalities, i.e., audio, vision, and meta-data, and utilize synthetic anchor points to bridge the gap between training data and test data. Extensive experiments exhibit the effectiveness of the proposed method.

2. PROBLEM FORMULATION

2.1. Problem Definition

The competition task is to learn a model that can compute the relevance score among TV-Shows/Movies from the video contents and corresponding meta-data (e.g., actor/actress, director, description, genre). Specifically, there are two sets of items, i.e., retrieval set $\mathcal{R}$ and candidate set $\mathcal{C}$, where $\mid \mathcal{R} \mid= R$ and $\mid \mathcal{C} \mid= C$. For each item (i.e., TV-Show/Movie) $r \in \mathcal{R}$ in the retrieval set, the organizers provide several trailers and the meta-data associated with it. Besides the trailers and meta-data, a vector $v^r = \{v_1^r, v_2^r, ..., v_C^r\} \in \mathbb{Z}^C$ is also provided for each item in the retrieval set, where $v_c^r = m, m \in \{1, 2, ..., M\}$ indicates candidate item $c \in \mathcal{C}$ is the m^{th} relevant item among all the C candidate items, while $v_c^r = 0$ indicates candidate item c is not among the top M most relevant items of item r.

Here, the organizers only provide the top $M = 30$ relevant items for each item r. All the provided relevance were learned from massive *hulu* user behaviors and could be treated as ground truth, and there is no overlap between $\mathcal{R}$ and $\mathcal{C}$, i.e., $\mathcal{R} \bigcap \mathcal{C} = \emptyset$. Due to the legal issue, neither the videos nor the meta-data of the items in candidate set are provided. The retrieval set $\mathcal{R}$ is divided into training, validation and testing set, and participants are expected to train a model to compute the relevance or similarity score $w_{r^\star,r}$ between item $r^\star$ in the testing set and item r in the training set by utilizing provided videos and meta-data, and then use the relevance score $w_{r^\star,r}$ to compute a result vector $s^{r^\star} = \{s_1^{r^\star}, s_2^{r^\star}, ..., s_C^{r^\star}\} \in \mathbb{R}^C$, where $s_c^{r^\star}$ indicates the relevance between $r^\star$ with the candidate item $c \in \mathcal{C}$.

2.2. Problem Challenge

The competition is challenging, and the reason lies in three aspects, i.e., large vision appearance variance, insufficient training data, and serious data incompleteness.

Large vision appearance variance: the vision appearance and audio usually vary greatly among different trailers, even among trailers coming from the same TV show. In the worst case, intra-class variance could be greater than inter-class variance, thus forming a huge challenge for discriminative feature selection.

Insufficient training data: the provided training data is insufficient, especially in the big-data era. More specifically, the organizers only provide 40 TV shows for model training (each TV show contains 5 trailers in average), thus forming a certain degree of difficulty for model fitting.

Serious data incompleteness: for the current competition, only the training and testing data are available, i.e., participants could have the access to both raw videos and corresponding meta-data. However, for the candidate set $\mathcal{C}$, nothing information was released to participants due to the legal issue. Therefore, how to estimate the representations of candidate set becomes particularly important.

2.3. Measurements

To measure the performance of the proposed method, we use the standard recall@Top$N$ and ranking loss $\mathcal{L}$ [4] [5] as the measurements, which measure the difficulty to derive the desiring ranking by the prediction of the proposed methods. More specifically, suppose the ground-truth top M most relevant items in candidate set for item r in the testing set is $o^r = [o_1^r, o_2^r, ..., o_M^r]$ with order, where $o_i^r \in \mathcal{C}$ is the item in the candidate set that ranked at i^{th} position in o^r. Similarly, $\tilde{o^r} = [\tilde{o_1^r}, \tilde{o_2^r}, ..., \tilde{o_M^r}]$ denotes the predicted result. Moreover, s_c^r is the score indicating the predicted relevance by the proposed method between item r and item o_i^r. Then, the recall@Top$N$ and ranking loss can be defined as Eqn. (1),

$$\text{recall@Top}N = \frac{|o^r \bigcap \tilde{o^r}|}{|o^r|}, \qquad (1)$$

and the ranking loss can be defined as Eqn. (2):

$$\mathcal{L}(s, o) = -\sum_{i=1}^{M} \log \frac{\exp(s_{o_i})}{\sum_{j \in \mathcal{C} \setminus o_{:i-1}} \exp(s_j)}, \qquad (2)$$

where $\mathcal{C} \setminus o_{:i-1}$ is a set containing all the candidate items in $\mathcal{C}$ except the first $i-1$ items in o. In Eqn. (2), we ignore the superscript r for simplicity.

3. PROPOSED METHOD

3.1. Different Modality Representation

In this paper, we propose to represent a complicated TV-show from different modalities, i.e., audio, meta-data, and vision features.

Audio feature: audio information is of significant importance especially for trailer video, and usually the rhythm of background music could tell the difference. The Munich open-Source Media Interpretation by Large feature-space Extraction (openSMILE) toolkit [6] [7] is a modular and flexible feature extractor for signal processing and machine learning applications. Its primary focus is clearly put on audio-signal features. In this work, we utilized several standard audio feature sets as follows, i.e., avec2011, emo_large, emobase, emobase2010, IS09_emotion, IS10_paraling, IS10_paraling_compat, IS11_speaker_state, IS12_speaker_trait_compat, IS13_ComParE.

Meta-data feature: the provided meta-data information includes actor/actress, director, description, and genre. Considering the small amount of training data, in this paper we only take the advantage of show descriptions. For description representation, we first apply the Latent Dirichlet Allocation (LDA) algorithm [8] to generate a topic model from about 400 TV-show descriptions (we build this corpus by crawling data in terms of genre from IMDb [9], which is treated as the world's most popular and authoritative source for movie, TV and celebrity content), and then compute the topic distribution probability for each TV-show.

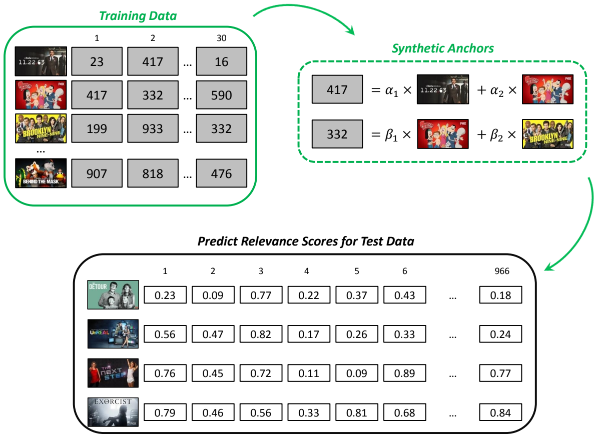
Vision feature: based on recent advances in computer vision more and more deep convolutional neural networks are developed that become increasingly proficient in mimicking perceptual inference abilities of humans. As a side effect of their popularity in technology, the increasing availability and diversity of high-performing neural network models opens a new door for studying the neural mechanisms of perceptual skills such as robust object recognition, and scene understanding. Therefore, in this paper we utilize deep features for video frame representation. More specifically, for each video, we uniformly sample 5% frames to extract their vision feature and further compute their mean representation as the final signature. Technically, we extract frame-level features using a state-of-the-art deep model: the publicly available Inception-v3 network [10] trained on ImageNet [11]. Concretely, we refer to [12] to feed the decoded frames into the Inception network, and fetch the ReLu activation of the last hidden layer, before the classification layer (layer name pool_3/_reshape). The feature vector is 2048-dimensional. While this removes motion information from the videos, recent work shows diminishing returns from motion features as the size and diversity of the video data increases [13] [14]. Besides, we also use another Inception-v3 model based on Open Images (a data set of about 9 million images that have been annotated with labels spanning over 6,000 categories) [15] annotations. The feature length is 6,012.

3.2. Synthetic Anchor

As mentioned before, we have no access to the candidate data, that is to say, matching test data and candidate data directly is infeasible. A very natural solution is estimating the representation of invisible candidate data according to their relevance relationship with training data. The estimated candidate data can be treated as synthetic anchor points which could well bridge the gap between training data and test data. Technically, we conduct the synthesis as Eqn. (3),

$$f_c = \sum_{i \in \mathcal{T}_c} \frac{1}{r_{ci}} \times f_i, c \in \mathcal{C}, \tag{3}$$

where $f_* \in \mathbb{R}^d$ denotes the d-dimensional video representation, r_{ci} denotes the rank position of c^{th} candidate data w.r.t i^{th} training data, and $\mathcal{T}_c$ is the training data subset in which c^{th} candidate data appears within the top M rank list. Please kindly refer to Figure 2 for better understanding. After generating the synthetic anchor points, i.e., the candidate representations, we can predict the relevance score between test data and candidate data.


Training Data
Synthetic Anchors
417 $= \alpha_1 \times$ $+ \alpha_2 \times$
332 $= \beta_1 \times$ $+ \beta_2 \times$
Predict Relevance Scores for Test Data

	1	2	...	30
	23	417	...	16
	417	332	...	590
	199	933	...	332
...				
	907	818	...	476

	1	2	3	4	5	6		966
	0.23	0.09	0.77	0.22	0.37	0.43	...	0.18
	0.56	0.47	0.82	0.17	0.26	0.33	...	0.24
	0.76	0.45	0.72	0.11	0.09	0.89	...	0.77
	0.79	0.46	0.56	0.33	0.81	0.68	...	0.84

Fig. 2. Illustration of the proposed method, where only training and testing data are available, and nothing information for the candidate set is released to participants due to the legal issue.

4. EXPERIMENTS AND ANALYSIS

4.1. Data Organization

The challenge organizers provide 400 trailers (i.e., videos) extracted from 80 TV-shows/Movies as the retrieval set. Along with the video clip, there are also thumbnail and metadata (e.g., description, genre, actor/actress, director) associated with each TV-show/Movie. For evaluation purpose, the whole set is further divided into three subsets, i.e., training, validation, and testing. The distribution is listed as follows: 1) training set, no.shows = 20, no.trailers = 163, no.thumbnails = 163; 2) validation set, no.shows = 20, no.trailers = 59, no.thumbnails = 59; 3) testing set, no.shows = 40, no.trailers = 178, no.thumbnails = 178. In our experiments, we combine the training and validation set for feature and model learning.

4.2. Experimental Analysis

We evaluate the proposed method and report the performance in Table 1 and Table 2. There are three types of features derived from different modalities, i.e., audio, meta-data, and vision. Table 1 demonstrates the recall rate with different valued Top$N$. We can observe that the vision model performs best because it encodes the most crucial and descriptive information in the videos. Besides, some audio features also show

Table 1. Evaluation of different modality representation with recall@Top$N$ measurement, where (a.), (m.), and (v.) respectively denotes audio, meta-data, and vision feature.

Feature	Recall@30	Recall@50	Recall@100
(a.) avec2011	0.0318	0.0443	0.1052
(a.) emo_large	0.0484	0.0836	0.1574
(a.) emobase	0.0790	0.1260	0.2224
(a.) emobase2010	0.0780	0.1184	0.2106
(a.) IS09_emotion	0.0764	0.1239	0.2276
(a.) IS10_paraling	0.1139	0.1565	0.2407
(a.) IS10_paraling_compat	0.1104	0.1364	0.2236
(a.) IS11_speaker_state	0.0480	0.0682	0.1530
(a.) IS12_speaker_trait_compat	0.0488	0.0754	0.1256
(a.) IS13_ComParE	0.0359	0.0509	0.0820
(m.) topic-model-20	0.0163	0.0541	0.1381
(m.) topic-model-30	0.0434	0.0717	0.1639
(m.) topic-model-40	0.0494	0.0782	0.1427
(m.) topic-model-50	0.0425	0.0796	0.1449
(v.) Inception-v3-2048 [12]	0.1668	0.2521	0.3632
(v.) Inception-v3-6012 [15]	0.1026	0.1555	0.2511

good performance, e.g., IS10_paraling and IS09_emotion, that's because the trailer with short time duration usually contains the most representative rhythm for a specific show. In comparison, the topic model is not very desirable. The reason may be that the training corpus for topic model training is insufficient, i.e., we only collected 466 top shows' storyline from the IMDb website [9] to learn the topic model.

Table 2. Evaluation of different modality representation with ranking loss measurement, where (a.), (m.), and (v.) respectively denotes audio, meta-data, and vision feature. This table only exhibits the best feature for each modality.

Feature	Recall@100	Ranking Loss
(a.) IS10_paraling	0.2407	184.6668
(m.) topic-model-30	0.1639	184.6667
(v.) Inception-v3-2048 [12]	0.3632	184.6671

5. CONCLUSION

This paper addressed the cold-start problem in the video recommender system, i.e., when a new coming video is added to the library, the recommender system has to bootstrap the video relevance score with very few user behavior with respect to the newly added video. To solve this problem, we proposed a content-based video recommendation approach by taking the advantage of deep convolutional neural networks. Specifically, we extract frame-level features using state-of-the-art deep models and propose to utilize synthetic anchor points to bridge the gap between training data and test data and cope with the serious data incompleteness. In addition to the vision feature, we also conducted extensive evaluation on video meta-data, and audio features. In summary, with the support from deep convolutional neural networks, frame-level vision feature exhibits its superiority against textual meta-data and audio information.

6. REFERENCES

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

[2] Greg Linden, Brent Smith, and Jeremy York, "Amazon. com recommendations: Item-to-item collaborative filtering," *IEEE Internet computing*, vol. 7, no. 1, pp. 76–80, 2003.

[3] JHJB Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen, "Collaborative filtering recommender systems," *The adaptive web*, pp. 291–324, 2007.

[4] Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li, "Listwise approach to learning to rank: theory and algorithm," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 1192–1199.

[5] Yin Zheng, Bangsheng Tang, Wenkui Ding, and Hanning Zhou, "A neural autoregressive approach to collaborative filtering," in *Proceedings of the 33nd International Conference on Machine Learning*, 2016, pp. 764–773.

[6] Florian Eyben, Martin Wöllmer, and Björn Schuller, "Opensmile: the munich versatile and fast open-source audio feature extractor," in *Proceedings of the 18th ACM international conference on Multimedia*. ACM, 2010, pp. 1459–1462.

[7] Florian Eyben, Felix Weninger, Florian Gross, and Björn Schuller, "Recent developments in opensmile, the munich open-source multimedia feature extractor," in *Proceedings of the 21st ACM international conference on Multimedia*. ACM, 2013, pp. 835–838.

[8] David M Blei, "Probabilistic topic models," *Communications of the ACM*, vol. 55, no. 4, pp. 77–84, 2012.

[9] "Imdb: Internet movie database," `http://www.imdb.com/?ref_=nv_home`.

[10] "Tensorflow: Image recognition," `https://www.tensorflow.org/tutorials/image_recognition`.

[11] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

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

[13] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici, “Beyond short snippets: Deep networks for video classification,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4694–4702.

[14] Du Tran, Lubomir D Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri, “C3d: generic features for video analysis,” *CoRR, abs/1412.0767*, vol. 2, pp. 7, 2014.

[15] I Krasin, T Duerig, N Alldrin, A Veit, S Abu-El-Haija, S Belongie, D Cai, Z Feng, V Ferrari, V Gomes, et al., “Openimages: A public dataset for large-scale multi-label and multiclass image classification,” *Dataset available from https://github. com/openimages*, vol. 2, no. 6, pp. 7, 2016.