Feature Re-Learning with Data Augmentation for Content-based Video Recommendation

Jianfeng Dong[1], Xirong Li[2,3], Chaoxi Xu[2,3], Gang Yang[2,3], Xun Wang[1]
[1]College of Computer and Information Engineering, Zhejiang Gongshang University
[2]Key Lab of Data Engineering and Knowledge Engineering, Renmin University of China
[3]Multimedia Computing Lab, School of Information, Renmin University of China

ABSTRACT

This paper describes our solution for the Hulu Content-based Video Relevance Prediction Challenge. Noting the deficiency of the original features, we propose feature re-learning to improve video relevance prediction. To generate more training instances for supervised learning, we develop two data augmentation strategies, one for frame-level features and the other for video-level features. In addition, late fusion of multiple models is employed to further boost the performance. Evaluation conducted by the organizers shows that our best run outperforms the Hulu baseline, obtaining relative improvements of 26.2% and 30.2% on the TV-shows track and the Movies track, respectively, in terms of recall@100. The results clearly justify the effectiveness of the proposed solution.

KEYWORDS

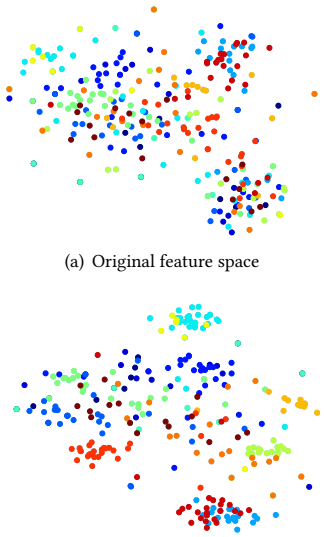
Content based video recommendation, Feature re-learning, Data augmentation

ACM Reference Format:
Jianfeng Dong, Xirong Li, Chaoxi Xu, Gang Yang, Xun Wang. 2018. Feature Re-Learning with Data Augmentation for Content-based Video Recommendation. In *Proceedings of 2018 ACM Multimedia Conference (MM'18), Oct. 22-26, 2018, Seoul, Republic of Korea.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3240508.3266441

1 INTRODUCTION

Video recommendation, by helping users discover videos of interest, is a very useful feature for online video services such as YouTube and Hulu. When new videos are uploaded to the services, user-interaction information such as browsing, commenting and rating is unavailable. This is known as the *cold-start* problem [9]. How to effectively recommend videos in a cold-start scenario is challenging.

This paper attacks the *cold-start* problem in the context of the Hulu Content-based Video Relevance Prediction Challenge [10]. In this challenge, participants are asked to rank a list of pre-specified videos in terms of their relevance with respect to a given video, where the relevance is exclusively computed based on the visual content. Notice that we as participants have no access to original video data. Instead, the organizers provide two visual features, extracted from individual frames and frame sequences by pre-trained Inception-v3 and C3D models, respectively.



(a) Original feature space

(b) Feature space re-learned by this work

Figure 1: Original feature space versus re-learned feature space. We randomly select 15 query videos and their corresponding relevant videos from the validation set of the TV-shows track [10], and use t-SNE [14] to visualize their distribution in (a) the original feature space and (b) the re-learned feature space obtained by our proposed model. Dots with the same color indicate videos relevant to a specific query. The plots reveal that relevant videos stay closer in the re-learned feature space than in the original feature space. Original feature: Inception-v3. Best viewed in color.

Although the deeply learned features are known to be powerful representations of visual content [3], we argue that they are suboptimal for this challenge. For the purpose of video recommendation, the computation of relevance between two videos shall reflect user's implicit feedback, *e.g.,* watch and search history. As Fig. 1(a) shows, relevant videos (denoted by the same colored dots) tend to scatter in the provided feature space. The features have to be re-learned.

It is well recognized that the more data a deep learning model has access to, the more effective it can be. However, collecting many video data for training a content-based video recommendation model is not easy, especially for TV-shows or movies as they have copyright issues. As aforementioned, the challenge does not

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

ACM ISBN 978-1-4503-5665-7/18/10…$15.00
https://doi.org/10.1145/3240508.3266441

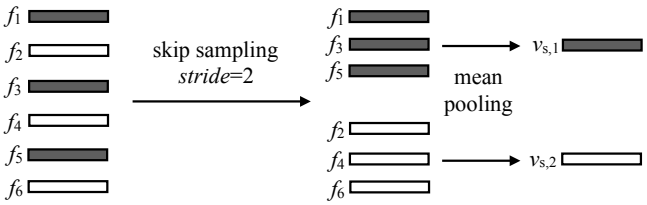


Figure 2: Data augmentation for frame-level features. Skip sampling with a stride of $s = 2$ yields s new sequences of frame-level features, and consequently s new training instances for the subsequent supervised learning.

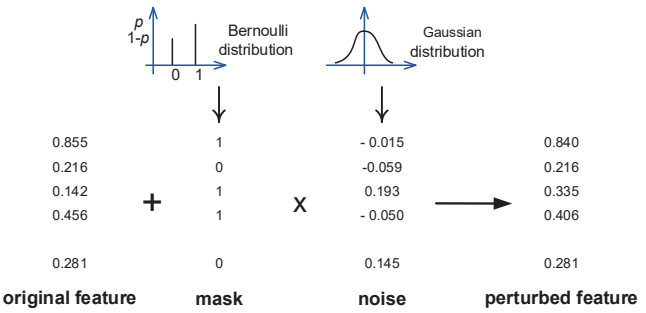


Figure 3: Data augmentation for video-level features, achieved by selectively adding Gaussian noise.

provide original videos. Consequently, existing data augmentation strategies such as flipping, rotating, zooming in/out, are inapplicable. We develop data augmentation strategies for features. For answering the Hulu challenge, this paper makes the following contributions.

- We propose feature re-learning with data augmentation. In particular, we introduce two data augmentation strategies that work for frame-level and video-level features, respectively. Combined with data augmentation strategies, a feature re-learning solution is developed.
- For both TV-shows and Movies tracks, the proposed solution outperforms the Hulu baseline with a large margin. In particular, we obtain recall@100 of 0.178 on the TV-shows track and 0.151 on the Movie track, while the corresponding numbers of the baseline are 0.141 and 0.116, respectively. Code is available at https://github.com/danieljf24/cbvr.

2 PROPOSED SOLUTION

Given a query video, we recommend relevant videos by retrieving its k nearest neighbors from a pre-specified video collection. To simplify our notation, let v indicate a video and a d-dimensional visual feature vector that describes the video content. Given two videos v and v', we measure their similarity in terms of the cosine similarity between the corresponding features, *i.e.,* $cs(v, v')$. We propose feature re-learning, expressed as $\phi(v)$, such that the relevance between two videos is better reflected by $cs(\phi(v), \phi(v'))$.

Next, we introduce our data augmentation strategies in Section 2.1, followed by a description of our feature re-learning model 2.2.

2.1 Data Augmentation for Features

Data augmentation is one of the effective ways to improve the performance of deep learning based models, especially when the training data are inadequate. Due to legal and copyright issue, the organizers do not provide original videos. Instead, they provide two precomputed features, *i.e.,* a 2,048-dim Inception-v3 feature per frame [1] and a 512-dim C3D feature per video [13]. We refer the interested reader to [10] for more details about the features. The unavailability of video data means common data augmentation strategies such as flipping, rotating, zooming in/out, are inapplicable. In what follows, we introduce two data augmentation strategies that work for frame-level and video-level features, respectively, with no need of original videos.

2.1.1 Augmentation for frame-level features. Inspired by the fact that humans could grasp the video topic after watching only several sampled video frames in order, we augment data by skip sampling. Figure 2 demonstrates the augmentation strategy for frame-level features. Given a video of n frames, let f_i be the feature vector of the i-th frame. We perform skip sampling with a stride of s over the frame sequence. In this way, s new sequences of frame-level features are generated. Accordingly, mean pooling is employed to obtain s new features at the video level, that is

$$\begin{aligned} v_{s,1} &= \text{mean-pooling}\{f_1, f_{1+s}, f_{1+2s}, ...\}, \\ v_{s,2} &= \text{mean-pooling}\{f_2, f_{2+s}, f_{2+2s}, ...\}, \\ &... \\ v_{s,s} &= \text{mean-pooling}\{f_s, f_{2s}, f_{3s}, ...\}, \end{aligned} \tag{1}$$

Together with the feature obtained by mean pooling over the full sequence, skip sampling with a stride of s produces $s + 1$ training instances for the subsequent supervised learning.

Notice that the skip sampling strategy is inapplicable for video-level features. To cope with the situation where only video-level features are provided, we devise another data augmentation strategy as follows.

2.1.2 Augmentation for video-level features. Adding tiny perturbations to image pixels are imperceptible to humans. In a similar spirit, we want our video recommendation system to ignore minor perturbations unconsciously introduced during feature extraction. To that end, we introduce perturbation-based data augmentation, as illustrated in Fig. 3. Given a d-dimensional video-level feature $v \in \mathbb{R}^d$, tiny Gaussian noises are randomly generated and selectively injected into the individual elements of the vector. More precisely, the perturbed feature v^* is generated by:

$$\begin{aligned} m &\sim Bernoulli(p), \\ e &\sim N_d(\mu, \sigma^2 I_d), \\ v^* &= v + \epsilon \cdot m \circ e, \end{aligned} \tag{2}$$

where m, as a mask, is a vector of independent Bernoulli random variables each of which has probability $p = 0.5$ of being 1, which controls how many elements in the video-level feature are perturbed. The variable e is a noise vector sampled from a multivariate Gaussian, parameterized by mean μ and covariance matrix $\sigma^2 I_d$, where I_d is a $d \times d$ identity matrix. The mean and the standard deviation are estimated from the dataset. We use $\epsilon = 1$ to control the noise intensity. The symbol $\circ$ indicates element-wise multiplication.

Notice that we perform the first data augmentation strategy on the Inception-v3 feature and the second strategy on the C3D feature. Both are conducted only in the training phase.

2.2 Feature Re-Learning

We devise a re-learning model to learn a new feature vector per video. Video recommendation for a given video is performed in the new feature space, where all candidate videos are sorted in descending order in terms of their cosine similarity to the given video.

2.2.1 Model structure. Before feeding videos to the re-learning model, we choose to first represent each video as a video-level feature vector. As the number of frame features varies over video, we employ mean pooling, which is simple yet consistently found to be effective in multiple content-based tasks [2, 3, 10–12]. We utilize a fully connected layer to map the original features into a new space. More formally, the new feature vector is represented as:

$$\phi(v) = Wv + b, \tag{3}$$

where W is affine matrix and b indicates a bias term. We empirically find that the model performance is insensitive to the dimensionality of the re-learned feature space. This dimensionality is set to be 1,024 in our experiments.

2.2.2 Model training. As we wish to make the relevant video pairs near and irrelevant video pairs far away in the re-learned feature space, we consider to utilize the common triplet ranking loss [5, 7] to train the feature re-learning model. Concretely, we first construct a set of triplets $\mathcal{T} = \{(v, v^+, v^-)\}$ from training set, where v^+ and v^- indicate videos relevant and irrelevant with respect to video v. The triplet ranking loss for a triplet of (v, v^+, v^-) is defined as:

$$\mathcal{L}(v, v^+, v^-; W, b) = max(0, \alpha - cs_\phi(v, v^+) + cs_\phi(v, v^-)), \tag{4}$$

where $cs_\phi(v, v')$ denotes the cosine similarity between $\phi(v)$ and $\phi(v')$, and α represents the margin, empirically set to be 0.2. Notice that other losses that exploit the negative samples exist. In our preliminary experiments, we have tried two other such losses, namely Contrastive Loss [6] and an improved triplet ranking loss [4]. We found them perform worse than the standard triplet ranking loss. Finally, we train the re-learning model to minimize the overall triplet ranking loss on a triplet set $\mathcal{T}$, and the overall objective function of the model is as:

$$\underset{W,b}{\text{argmin}} \sum_{(v,v^+,v^-)\in\mathcal{T}} \mathcal{L}(v, v^+, v^-; W, b). \tag{5}$$

We solve Eq. 5 using stochastic gradient descent with Adam [8], and empirically set the initial learning rate to be 0.001 and batch size to be 32. We adopt a learning schedule as described in [3]. Once the validation loss does not decrease in three consecutive epochs, we divide the learning rate by 2. The early stop occurs if the validation performance does not improve in ten consecutive epochs. The maximal number of epochs is 50.

3 EVALULATION

3.1 Experimental Setup

The challenge provides two video collections, corresponding to TV-shows and movies, respectively. Each collection has been divided into three disjoint subsets for training, validation and test. Detailed data split is as follows: training / validation / test of 3,000 / 864 / 3,000 videos for the TV-shows track and 4,500 / 1,188 / 4,500 videos for the Movies track. For each video in the training and the validation set, it is associated with a list of relevant videos as ground truth.

Following the evaluation protocol of the challenge, we report two rank-based performance metrics, *i.e.,* hit@k (k = 5, 10, 20, 30) and recall@k (k = 50, 100, 200, 300). Since recall@100 is the official metric, it is the default metric in the following experiments unless stated otherwise.

Table 1: Performance of feature re-learning with different loss. No data augmentation. Triplet ranking loss performs the best. Feature: Inception-v3.

Loss	TV-shows	Movies
Triplet ranking loss	**0.199**	**0.163**
Improved Triplet ranking loss [4]	0.181	0.125
Contrastive loss [6]	0.194	0.160

Table 2: Effectiveness of data augmentation. The model with data augmentation gives better performance.

Feature	Data augmentation	TV-shows	Movies
Inception-v3	✗	0.199	0.163
	✓	**0.244**	**0.191**
C3D	✗	0.185	0.155
	✓	**0.196**	**0.163**

Table 3: Effectiveness of feature re-learning.

Feature	Re-Learning	TV-shows	Movies
Inception-v3	✗	0.124	0.099
	✓	**0.244**	**0.191**
C3D	✗	0.145	0.112
	✓	**0.196**	**0.163**

3.2 Ablation Study

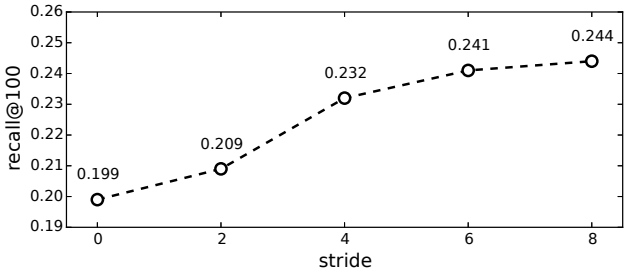
Each participant is allowed to submit five runs at maximum, making an ablation study on the test set unpractical. So we conduct the ablation study on the validation set as follows.

3.2.1 Choice of loss functions. Table 1 shows performance of our re-learning model with different losses. Triplet ranking loss consistently outperforms the other two loss functions on both two datasets.

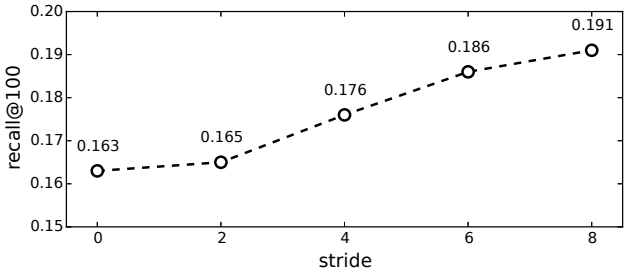
3.2.2 Effectiveness of data augmentation. Figure 4(a) shows the performance curves of feature re-learning with data augmentation as the stride increases. The rising curves justify the effectiveness of data augmentation for the frame-level feature. Best performance is reached with $stride = 8$. So we use this parameter in the rest of the ablation study.

Table 4: Performance comparison on the test set. Our runs outperform the Hulu's best baseline with a large margin. The number in the bracket indicates the relative improvement over the Hulu baseline.

	Track 1: TV-shows								Track 2: Movies							
	hit@k				recall@k				hit@k				recall@k			
	k=5	k=10	k=20	k=30	k=50	k=100	k=200	k=300	k=5	k=10	k=20	k=30	k=50	k=100	k=200	k=300
Hulu	0.249	0.356	0.461	0.525	0.085	0.141	0.219	0.269	0.190	0.242	0.320	0.373	0.081	0.116	0.168	0.206
run 1	0.274	0.365	0.488	0.542	0.099	0.160 (↑ 13.5%)	0.248	0.302	0.210	0.272	0.355	0.412	0.092	0.133 (↑ 14.7%)	0.192	0.237
run 2	0.287	0.381	0.492	0.550	0.104	0.167 (↑ 18.4%)	0.257	0.314	0.211	0.278	0.368	0.427	0.095	0.139 (↑ 19.8%)	0.201	0.248
run 3	0.288	0.391	0.484	0.539	0.099	0.162 (↑ 14.9%)	0.249	0.305	0.215	0.278	0.359	0.422	0.096	0.138 (↑ 19.0%)	0.198	0.244
run 4	**0.309**	**0.411**	0.506	0.567	0.109	0.173 (↑ 22.7%)	0.266	0.323	**0.234**	0.298	**0.390**	**0.448**	0.104	0.148 (↑ 27.6%)	0.210	0.258
run 5	0.308	0.408	**0.522**	**0.589**	**0.112**	**0.178** (↑ 26.2%)	**0.273**	**0.331**	0.232	**0.302**	0.389	0.441	**0.105**	**0.151** (↑ 30.2%)	**0.215**	**0.263**



(a) TV-shows



(b) Movies

Figure 4: Performance curves of feature re-learning with data augmentation performed on the Inception-V3 feature. The starting point, $s = 0$, means no data augmentation.

As for data augmentation on the video-level feature, *i.e.,* C3D, this strategy also works. As Table 2 shows, our model obtains recall@100 of 0.196 and 0.163 on the TV-shows and Movies tracks, while the scores of its counterpart without data augmentation are 0.185 and 0.155, respectively.

3.2.3 Effectiveness of feature re-learning. Table 3 shows performance of video recommendation with and without feature re-learning. For both Inception-v3 and C3D features, re-learning brings in substantial performance gain. These results show the importance of feature re-learning for content-based video recommendation.

3.3 Challenge Results

We submitted the following five runs.

- **run 1**: Feature re-learning with data augmentation of stride $s = 8$ performed on the Inception-v3 feature.
- **run 2**: Late fusion (with uniform weights) of the following eight models. Four models are separately trained with data augmentation of stride $s = 6$, $s = 8$, $s = \{2, 4, 6\}$ and $s = \{2, 4, 6, 8\}$ on the Inception-v3 feature. The setting $s = \{2, 4, 6\}$ means new instances generated by skip sampling with the varied value of s are exploited together. We train another four models in a similar manner as the aboves models, but add the tanh activation function after the fully connected layer.
- **run 3**: Similar to run 2, but using the concatenation of Inception-V3 and C3D as the input feature.
- **run 4**: Late fusion of the sixteen models from run 2 and run 3.
- **run 5**: Based on run 4, we additionally include four models trained using the Contrastive Loss and with their validation performance exceeding an empirical threshold of 0.22 on the TV-shows track.

The performance of our submitted runs, evaluated by the challenge organizers, is summarized in Table 4. All our runs are noticeably better than the Hulu's best baseline, justifying the effectiveness of our solution. Among them, runs using late fusion consistently outperform the single-model run. This result suggests that late fusion is quite helpful for boosting the performance.

4 CONCLUSIONS

We answer the Hulu challenge by proposing feature re-learning with data augmentation. The proposed solution is superior to the Hulu baseline: 0.178 versus 0.141 on the test set of TV-shows and 0.151 versus 0.116 on the test set of Movies in terms of recall@100. We attribute the good performance to the following three factors, *i.e.,* 1) data augmentation on features to generate more training instances, 2) feature re-learning with the marginal ranking loss, and 3) late fusion of multiple models.

Acknowledgments. This work was supported by the National Natural Science Foundation of China (No. U1609215, No. 61672460, No. 61672523, No. 61773385), the Fundamental Research Funds for the Central Universities and the Research Funds of Renmin University of China (No. 18XNLG19). Corresponding author: Xirong Li (xirong@ruc.edu.cn).

REFERENCES

[1] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan. 2016. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675* (2016).

[2] J. Dong, X. Li, W. Lan, Y. Huo, and C. G. M. Snoek. 2016. Early Embedding and Late Reranking for Video Captioning. In *MM*.

[3] J. Dong, X. Li, and C. G.M. Snoek. 2018. Predicting visual features from text for image and video caption retrieval. *T-MM* (2018). https://doi.org/10.1109/TMM.2018.2832602

[4] F. Faghri, D. J Fleet, J. R. Kiros, and S. Fidler. 2017. VSE++: improved visual-semantic embeddings. *arXiv preprint arXiv:1707.05612* (2017).

[5] A. Frome, G. S Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al. 2013. Devise: A deep visual-semantic embedding model. In *NIPS*.

[6] R. Hadsell, S. Chopra, and Y. LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *CVPR*.

[7] A. Karpathy and L. Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *CVPR*. 3128–3137.

[8] D. P Kingma and J. Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[9] B. Lika, K. Kolomvatsos, and S. Hadjiefthymiades. 2014. Facing the cold start problem in recommender systems. *Expert Systems with Applications* 41, 4 (2014), 2065–2073.

[10] M. Liu, X. Xie, and H. Zhou. 2018. Content-based Video Relevance Prediction Challenge: Data, Protocol, and Baseline. *arXiv preprint arXiv:1806.00737* (2018).

[11] M. Mazloom, X. Li, and C. G.M. Snoek. 2016. TagBook: A Semantic Video Representation Without Supervision for Event Detection. *T-MM* 18, 7 (2016), 1378–1388.

[12] Y. Pan, T. Mei, T. Yao, H. Li, and Y. Rui. 2016. Jointly Modeling Embedding and Translation to Bridge Video and Language. In *CVPR*.

[13] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. 2015. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*.

[14] L. van de Maaten and G. Hinton. 2008. Visualizing Data using T-SNE. *JMLR* 9 (2008), 2579–2605.