Exploiting Feature and Class Relationships in Video Categorization

Hao Fu
School of Data Science
Fudan University
14307130013

Chengming Xu
School of Data Science
Fudan University

Maoran Xu
School of Data Science
Fudan University
14300180099

Abstract

This paper studies the problem of video classification. The dataset FCVID includes over 90k Internet videos and 239 manually annotated categories. Since diverse features carry rich semantic information, we mainly focus on feature processing and fusion. We first employ a restricted Boltzmann machine (RBM) and principal component analysis (PCA) on features for dimension reduction. Then we utilize a feature fusion method based on late fusion to combine the outputs of classifiers for different features. Also we pay attention to class relationships hidden in the data, which helps determine the optimal fusion weights for deriving the final predictions.

1 Introduction

Video posting, sharing and viewing become quite common in people's daily life nowadays. Some see it as a new way of communication between Internet users (e.g., short videos in Wechat *Moments*). Video classification is a technique that deals with the big data composed of explosively generated videos. It is widely applied in content search and organization, recommandation systems on websites like Youtube.

The fact that videos are intrinsically multimodal and of complicated nature makes it a challenging task to categorize precisely. One should not only focus on static appearance information but acoustic channels, motion clues, etc. As for effective classification, prominent features draw heavy attentions. One of the popular methods is that people use Convolutional Neural Network (CNN) based representations as static features. Besides, typical feature descriptors include Mel-Frequency Cepstral Coefficients (MFCC), Space-Time Interest Points (STIP), etc.

Recognizing that single feature is usually not enough for discription, people consider combining multiple features in early works because different features provide complementary information. This is proposed as feature fusion and two simple instances are early fusion and late fusion. We implement a fusion method based on the latter, combining outputs of classifiers. Meanwhile, to tackle the limitations of traditional late fusion, we introduce and study inter-class relationships (e.g., "makingPizza" and "makingSandwich" tend to be correlated), which can boost the classification

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

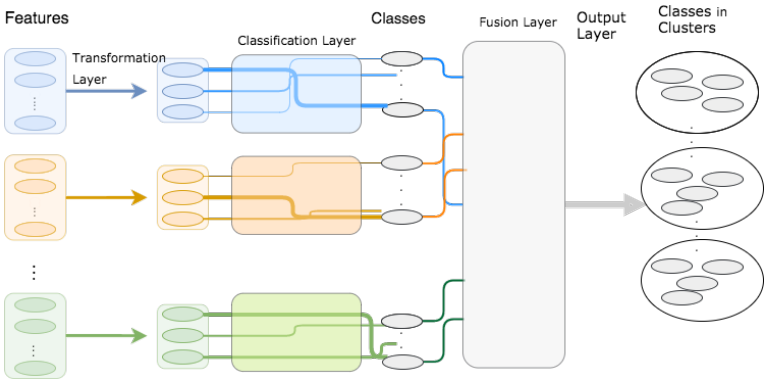

Figure 1: Illustration of our framework for video categorization. With the M features extracted from various points of view on video samples, we first transform them with PCA method to reduce dimensionality. C classifiers are seperately trained on each feature and give M results in predicting labels. That is, some dimensions of one feature are strongly connected with a category, some are weakly connected, as the arrow pointing to a class can be thick or thin. In order to study the feature relationship and class relationship, we put regularizers in the fusion layer and reconstruct a new classifier. This classifier can not only improve the result with shared knowledge in features and classes, but also indicate closely related classes that come into clusters.

performance (1).

Figure 1 delivers the flowchart of our approach. We retrieve the off-the-shelf features from http://bigvid.fudan.edu.cn/data/fcvid/FCVID_Feature/ so that we do not work on the raw video data. We use a restricted Boltzmann machine (RBM) and principal component analysis (PCA) to reduce dimensions. For each feature we deploy, C classifiers are trained, with $C = 239$ being the number of classes. In the fusion layer, an aforemetioned method is developed to learn the optimal fusion weights. Since we have learned latent class relationships beforehand, the final predictions are constrained.

2 Related work

Video features We review recent works related to our approach from the perspective of CNN and hand-crafted representations of video features. CNN is proved successful on image feature extraction. However, its performance in video classification meets the bottleneck caused by complexity and scale of videos. Thus works have been done to extend the CNN to exploit more information. Ji *et al.* (2) added spatial-temporal space to CNN, but only focused on static frames and motion features got by adjacent frames, which is not complete enough. The two-stream CNN approach by Simonyan *et al.* (3) applied the CNN respectively on visual frames and stacked optical flows. Szegedy *et al.* (4) imported a pre-trained CNN on ImageNet 2012 Challenge data to extract a 4,096-dimension feature for each video frame input. Xu *et al.* (5) adopted advanced feature encoding strategies to promote the generalization ability of CNN representations. In this paper, we utilize the CNN representation provided by Jiang *et al.* (1).

Hand-crafted features cover a wider range. One can apply image-based shape features involving histogram of oriented gradient (HOG) and scale-invariant feature transform (SIFT) (6). As for motion features, which extends frame-based local features into 3D space, Wang *et al.* (7) located densely sampled frame patches to generate dense trajectories. Moreover, audio features such as MFCC, which is taken into consideration in this paper, are seen as a complement of visual and motion information.

Feature fusion The aim of fusion is to introduce multiple types of clues in videos for performance improvement. A simple example is using the linear combination of regurlarized features. In advanced cases, Li et al. (8) employed the covariance matrix of features and combine them in the Riemann manifold. Srivastava *et al.* (9) fused features with a deep Boltzmann machine. Kwon *et al.* (10)

decomposed object tracking into multiple components and then combined for robust tracking. In this paper, we concatenate the outputs of classifiers upon different features.

Class relationship Many works studied class relationships to improve classification performance. Jiang *et al.* (11) used a semantic diffusion algorithm to cluster and reveal class relationships. Chen *et al.* (12) used confusion matrix to imply class correlations and then use it in regulizing the loss function in neural network. We cope with class relationships conprehensively according to the theoretical baselines in (1) and (13).

3 The proposed approach

3.1 Dimensionality reduction with RBM

To implement the feature fusion step, we must first extract the import dimension of each feature so that the complex computing is feasible. The mainstream way to reduce dimensionality is Principal component analysis (PCA), which maps the sample data to a lower-dimensional space in order to make the variance max and reconstruct a smaller feature space. However, since it is a linear method, it may not be efficient enough to describe the non-linear properties within the spectra. So Restricted Boltzmann Machine (RBM) is introduced to substitute PCA method.

A restricted Boltzmann machine is a particular type of Markov random field with two-layer structure, i.e. a visible vector $v \in R^{n_v}$ connected to a hidden vector $h \in R^{n_h}$ (n_v,n_h: size of each vector). Succintly put, it can be roughly used as an encoder that projects the target vector on the former layer to the hidden space of desirable dimension.

Suppose that $\theta = \{W, b, a\}$ are the unknown parameters of an RBM, where W is the weight matrix and a, b denote biases of visible and hidden units. The joint distribution is therefore

$$P(v, h, \theta) = \frac{1}{Z(\theta)} \exp(-E(v, h, \theta)),$$

where $Z(\theta) = \sum_v \sum_h \exp\left(-E(v, h, \theta)\right)$ is the partition function and $E(v, h, \theta) = -a^T v - b^T h - h^T W v$ is the energy function. Here with regard to likelihood we have

$$P(h|v, \theta) = \prod_j P(h_j|v)$$
$$P(v|h, \theta) = \prod_i P(v_i|h)$$

satisfying

$$P(h_j = 1|v) = \sigma(\sum_j W_{ij} h_j + a_i)$$
$$P(v_i = 1|h) = \sigma(\sum_i W_{ij} v_i + b_j).$$

Then we demonstrate the RBM training process. The purpose is to learn θ to get the values of units in both layers, so we define $P(v, \theta) = \sum_h P(h, v, \theta)$, which implies the distribution of the data. In this case the optimal θ^* is to maximize $P(v, \theta)$. According to Hinton *et al.* (15), we choose $\delta W = \eta(E_{data}[vh^T] - E_G[vh^T])$ to approximate the gradient. Note that E_{data} is the expectation conputed by data and E_G is the expectation of a distribution P_G in a Gibbs sampling process.

3.2 Exploiting feature fusion

Every single type of feature can be utilized to predict labels of videos. But for the complexity of large-scale videos, each feature may have its advantages and disadvantages. For instance, some classes like sports and action movies may strongly related to visual motion and trajectory feature while some classes including chorus and marching band may closely related to audio features.

Therefore, feature fusion can largely enhance the accuracy of predicted labels.

We can follow and capture video characteristics from various levels as we collect semantic features. Feature fusion incorporates different aspects of videos like audio, motion and frame in order to develop complementary information to make better classifications. Traditional late fusion method simply assigns weights to the prediction scores of multiple features, which is suggested overlooking the intrinsic relations among multiple feature representations (16). When fusing feature with a DNN, the fusion layer can be written as:

$$a_F = \sigma(\sum_{m=1}^{M} W^m a^m + b),$$

where a^m is the transformed neurons in the m^{th} feature.

The objective for the fusion process should capture the relations among the features but not lose their uniqueness and efficiency. Thus, instead of straightly mix all dimensions of features together, we consider to change the optimization function a little, that is to put regularizers concerning both the relationship and every single feature. As Jiang *et al.* presented in (1), a trace norm regularization term is added to the empirical loss function, given by

$$\frac{\lambda_1^2}{2} tr(W_E \Phi^{-1} W_E^T).$$

Feature correlation matrix Φ is symmetric and positive semidefinite, and the term above helps learn the inter-feature relationships. W_E is the weight matrix in the fusion layer and it stores available features after transformation. The Cauchy-Schwarz inequality gurantees the solution of Φ in the extended loss function to be

$$\Phi^* = \frac{(W_E^T W_E)^{\frac{1}{2}}}{tr((W_E^T W_E)^{\frac{1}{2}})}.$$

3.3 Mining class relationships

Since there are 239 classes in this task, that much of labels can share a lot of knowledge to a classification question. For example, making pizza and making cookies can be very similar in pattern recognition, and wedding dance or wedding ceremony may co-occur with the same video. At the same time, there are related classes that share little feature similarity, like cat and dog. Before class fusion step, we have trained several classifier with a single feature. The prediction them give can be a prior message to the fused classifier. Therefore, we choose to obtain knowledge from the pre-trained models with confusion matrix, which straightly gives a class relationship connected to the model. An example confusion matrix is displayed in Fig 2.

To avoid over-fitting problem, regularizers are added to the loss function in the form of l_1 or l_2 norm of the weight matrix to penalize non-zero weight term. Regularization term concerning confusion matrix can constrain how much a class is needed to share its knowledge. The optimal weight of a class could be learned in the optimization function without over-fitting problem. AVideo classes share similarity due to latent inter-class relationships. Automatic grouping of classes helps improve the result. Likewise, we consider a regularization term

$$\frac{\lambda_2^2}{2} tr(W_{L-1} \Omega^{-1} W_{L-1}^T)$$

and have

$$\Omega^* = \frac{(W_{L-1}^T W_{L-1})^{\frac{1}{2}}}{tr((W_{L-1}^T W_{L-1})^{\frac{1}{2}})}$$

where W_{L-1} is the weight matrix on the output layer.

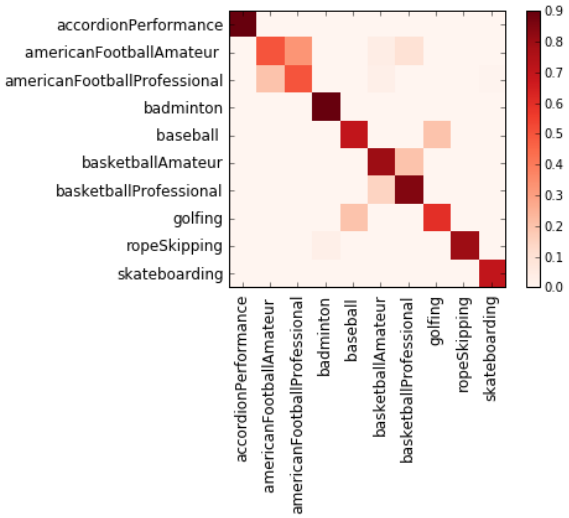

Figure 2: An example of Confusion Matrix in the video classification case, on FCVID.

On the other hand, we measure the confusion matrix V^m of a particular feature m on the validation set and use it as prior. V_{ij} refers to the correlation between class C_i and C_j, computed by

$$V_{ij}^m = \frac{1}{|C_i|} \sum_{n \in C_i} 1_{\arg\max_c(s_n^m)==C_j}.$$

Here s_n^m is the prediction scores of training instance n using feature m. In (13), Wu *et al.* considered another regularization term $\lambda||W - V||_F^2$. V consists of all individual V^m. The class relationship matrix V constrains which classes to be considered, avoiding overfitting in some way.

3.4 Algorithm

According to all the analysis above, we finally get to the optimization problems:

$$\min_W \; L + \frac{\lambda_1}{2}||W - V||_F^2 + \lambda_2||W||_1 + \frac{\lambda_3}{2} tr(W_E \Phi^{-1} W_E^T) + \frac{\lambda_4}{2} tr(W_{L-1} \Omega^{-1} W_{L-1}^T) \quad (1)$$

The first term is the empirical loss between the prediction and the true label. The second term is the regularization term to prevent over-fitting. The third term is the regularization from the confusion matrix, which is unsmooth and optional. The forth and fifth terms are regularizations for feature relationship and class relationship. The concrete algorithm is listed as Algorithm 1. For convenience, we replace the C classifiers with one multi-label classifier. A problem with this multi-label classifier is that the number of classes is large so that the label matrix is sparse, which makes the classifiers tend to return a prediction vector with all zeros. To solve this problem, we try to enlarge the weight on the 1 term in the true label, which makes the penalty for the zero vector larger.

4 Experiments

4.1 Experimental setup

4.1.1 Dataset and Evaluation

We implement the experiments over Fudan-Columbia Video Dataset (FCVID), a video dataset containing 91,223 Internet videos along with 239 manually annotated categories. FCVID covers concepts such as scenes, events, etc. we follow the split with the raw dataset along with the off-the-shelf features, which returns a train set and a test set with equal size. Then for validation, we randomly choose 5% of the train set.
We report mean average precision (mAP) as the overall results.

Algorithm 1 Training procedure

Train M DNN classifiers, where M is the number of features.
Randomly initialize W, $\Phi = \frac{1}{M} I_M$ and $\Omega = \frac{1}{C} I_C$, where I_M and I_C are identity matrices.
Feed forward the feature inputs to different DNN classifiers, with the outputs to be the input of the fusion layer. Compute the confusion matrix V for each feature.
for epoch = 1 to K **do**
Back propagate the prediction error from layer L to layer 1 by proximal gradient descendant.
Update the feature relationship matrix Φ according to:

$$\Phi = \frac{(W_E^T W_E)^{\frac{1}{2}}}{tr((W_E^T W_E)^{\frac{1}{2}})}$$

Update the class relationship matrix Ω according to:

$$\Omega = \frac{(W_{L-1}^T W_{L-1})^{\frac{1}{2}}}{tr((W_{L-1}^T W_{L-1})^{\frac{1}{2}})}$$

end for

4.1.2 Features

Although the dataset have already provided us with as many as 8 features produced by different methods and more features mean more information of the raw video data, which may have a satisfying performance, using all of these features with dimension of more than 4000 is impractical and time-consuming. Therefore, we try combinations of the whole feature set, including visual features such as motion boundary histogram(MBH) descriptor, audio features such as MFCCs(Mel-Frequency Cepstral Coefficients) and static CNN features.

4.2 Result and discussion

We first compare performance out of early-fused features and each single feature with different proportion of the train set as data for train. One can observe subtle but significant advantages of DNN-Fusion Regularization (DNN-FR) in Table 1, for many classes in FCVID can be recognized through few specific frames. As we increase the train data, the performance gets better.

Table 1: Performance comparison (mAP) on the different size of train set and fusion methods

train set proportion	Early Fusion	DNN-FR
20%	30.6%	31.2%
60%	45.8%	48.7
100%	71.2%	72.4%

After that we compare the different combination of features using method of DNN-Fusion Regularization and the whole train set. For convenience, we just combine these futures in two or three. The result is shown in Table 2. From the data in Table 2 we can find that since HOF, HOG etc. are the same kind of feature, when combining them together, the performance is not rather satisfying, which is more like completing the feature, not improving the feature. In sharp contrast, when combining CNN feature with one of visual features and one of audio features, the performance is much better.

Table 2: Performance comparison (mAP) on different combinations of features

Features	Performance
CNN+HOG	63.4%
CNN+MFCC	54.8%
HOF+HOG	59.1%
CNN+MFCC+HOF	72.4%

Then we look into class relationships. With a class relationship regularizer, our network outperforms the previous results. By confusion matrices, we obtain many class clusters in which classes share common characteristics, such as {"carAccidents", "carExhibition", "carRacing", "carWashing"}, {"marathon", "marchingBand"}, etc. We can read from the matrices that if two or more classes prone to be wrongly predicted between each other, then these tend to be related classes.

5 Conclusion

Since all these experiment are based on given features, we are unable to compare them to the performance of the most recent method on video classification through deep learning with CNN and LSTM. The most knotty problem is that this model with several DNN classifiers has more parameters and hyper-parameters, which makes it comparatively less efficiency and hard to train. But the fusion method in our experiments is still powerful. Perhaps with more comprehensive features, this kind of feature fusion method can provide better performance.

References

[1] Y. Jiang, Z. Wu, J. Wang, X. Xue, and S. Chang. Exploiting feature and class relationships in video categorization with regularized deep neural networks. *ArXiv preprint*, 2015.

[2] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. In *ICML*, 2010.

[3] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014.

[4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, Going deeper with convolutions. *CoRR*, 2014.

[5] Z. Xu, Y. Yang, and A. G. Hauptmann. A discriminative cnn video representation for event detection. In *CVPR*, 2015.

[6] D. G. Lowe. Distinctive image features from scale-invariant keypoints. In *IJCV*, 2004.

[7] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013.

[8] X. Li, W. Hu, Z. Zhang, and X. Zhang. Robust visual tracking based on an effective appearance model. In *ECCV*, 2008.

[9] N. Srivastava and R. Salakhutdinov. Multimodal learning with deep Boltzmann machines. In *NIPS*, 2012.

[10] J. Kwon and K. M. Lee. Visual tracking decomposition. In *CVPR*, 2010.

[11] Y. Jiang, J. Wang, S. Chang, and C. Ngo. Domain adaptive semantic diffusion for large scale context-based video annotation. In *ICCV*, 2009.

[12] X. Chen and A. Gupta. Webly supervised learning of convolutional networks. In *ICCV*, 2015.

[13] Z. Wu, Y. Jiang, X. Wang, H. Ye, and X. Xue. Multi-stream multi-class fusion of deep networks for video classification. In *ACMM*, 2016.

[14] Y. Bu, G. Zhao, A. Luo, J. Pan, and Y. Chen. Restricted Boltzmann machine: a non-linear substitute for PCA in spectral processing. *AA*, 2015.

[15] G. Hinton and R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 2006.

[16] P. K. Atrey, M. A. Hossain, A. El Saddik, and M. S. Kankanhalli. Multimodal fusion for multimedia analysis: a survey. *Multimedia systems*, 2010.