Tree-based Regression for Few-shots Video Categorization

Anonymous Author(s)
Affiliation
Address
email

Abstract

Video categorization is becoming an important problem with the increasing amount of Internet video data. While the great success of image classification has been achieved by deep neural network, video categorization problem is getting more attention. However, labeled video data is much more expensive than labeled image data, and this posed a big challenge to the society. In this project, we propose to use tree-based logistic regression to utilize the relationship in class labels as a prior to improve the accuracy of video categorization. The experiments show that our method can achieve a good result with limited training data(at most 10% of the state-of-the-art model) and limited computation resources(only a CPU).

1 Introduction

With the increasing number of Internet users, images and videos become ubiquitous on the Internet. The uprising growth in visual media calls for a robust algorithm to do online media categorization, summarization and research, in which categorization is the basic task. The classification problem of images gets earlier attention than the one of videos, thus having more mature features and algorithm[7]. Video categorization problem is more complicated. Besides the information of the time series of images, video also contains audio information and moving trajectory information. So we face the problem of feature selection and combination.

Since there are so many videos uploaded all the time, the traditional time-consuming method based on training on a large manual-labeled dataset can't reach a timely categorization result. What's more, the pile up information in time series leads to a much larger scale of dataset, which makes problem solving limited by computing resources. Based on the above two points, our task aims at setting up a model trained on a small data to have good performance.

In our work, we use Fudan-Columbia Video Dataset (FCVID)[5], which contains 91,223 YouTube videos and 239 manually annotated categories. It is one of the largest manually annotated datasets of Internet videos. Several features, such as CNN, MFCCs, e.t.c, have been extracted. The hierarchy of categorization is also provided in the data. In our work, we build a tree-based regression model using static CNN and MFCCs features ,and train it respectively on 1,000, 3,000, 5,000 samples. In addition, we use different combinations of the features to figure out their differences and complementarity of each type of feature. We also compare the tree-based regression model with simple logistic regression model and sampling k-NN model to confirm its superiority on a small samples. Our algorithm can achieve a good accuracy of classification with few manually labeled data.

The rest of this paper is organized as follows. Section 2 discusses related works, where we mainly focus on the classification models exploiting class relationships. Section 3 elaborates the proposed tree-based regression approach. We also introduce linear regression model(as baseline method) and sampling k-NN(We add the word 'sampling' to emphasize k-NN as a sampling method). Extensive

Submitted to 30th Conference on Neural Information Processing Systems (NIPS 2016). Do not distribute.

experimental results and comparisons with other methods and recent state of the arts are discussed in Section 4. Finally, Section 5 concludes this paper.

2 Related work

The problem of video classification has become a hot research direction in recent years. Many works focused on feature design, such as CNN feature [7], audio feature[1], trajectory feature[8]. Others mainly focused on design of the model. Neural network has been adopted for video categorization in [6], which used local spatio-temporal information and suggested a multi-resolution. There are other famous models based on CNN, such as AlexNet, VGGNet and GoogleNet[10]. In [4], the authors trained two two-layer LSTM networks for action recognition.

The relationship between class labels has received significant research attention. [2] used Hierarchy and Exclusion graphs to capture the co-occurrence class relationships as well as mutual exclusion and subsumption. [3] used a tree-style classifier. Through maximizing the information gain by applying Dual Accuracy Reward Trade-off Search (DARTS) algorithm, they proposed that the hierarchical classification can ensure both the accuracy and the specificity.

[5] released the FCVID dataset and constructed a DNN model based on Static CNN, Motion Trajectory and Audio features. They added special regularizers to learn the relationships between different classes. They used standard split of 45,611 videos for training and 45,612 videos for testing and reached an mAP of 73% [5]. In recent work of the FCVID dataset, [9] proposed a novel object and scene-based semantic fusion network and representation and reached an mAP of 76.5%. The following table shows the state-of-the-art of the current models on FCVID dataset.

Model	mAP on FCVID
Early Fusion-NN	75.2
Late Fusion-NN	73.3
Early Fusion-SVM	75.5
Late Fusion-SVM	73.4
SVM-MKL	74.9
DNN[5]	73.0
OSF Networt[9]	76.5

Table 1: The table shows the state-of-the-art of the current models on FCVID dataset. Different from our work, these models all used a large training sample.[9]

3 Approach

In this section, we start by introducing feature selection. Then we discuss three classifiers: logistic regression, tree-based regression and sampling k-NN. A simple logistic regression is used as the baseline of tree-based model. In addition, we apply sampling k-NN as another sampling method to verify the good performance of tree-based method.

3.1 Feature selection

Several features have been extracted already in FCVID dataset. So we only make a brief restatement of the features we use.

Static CNN Features As is described in the dataset, a CNN model pre-trained on the ImageNet 2012 is adopted to extract a 4096-d feature representation. And the average of a frame of image CNN features represents the video feature.

MFCCs Features Two types of video features, MFCCs and Spectrogram SIFT, are concluded in the dataset. We only consider MFCCs features in our work. Over each 32ms time-window with 50% overlap is computed. And a 4,000-d bag-of-words representation is quantized into.

A simple observation shows the following two points:
1.Both features have high dimensions, each video with a 4096-d CNN feature and a 4000-d MFCCs

feature. The dataset has 91123 labeled video in all. Due to our limited computing resources, it is necessary to find a good classifier trained on a small sample.
2.In the scale of magnitude,

$$CNN \gg MFCCs \sim 0$$

So the three types of feature probably share different weights. In our work, we experiment different combinations of the two types of features to provide a comparison of different features.

	Average	Median	Maximum	Minimum	Sum
CNN	0.36	0.11	4.92	0.00	1483.79
MFCCs	0.00	0.00	0.39	0.00	1.57

Table 2: A simple statistics of the dataset(e.g. the first set of the data). It provides us with an anticipation of the following work.

3.2 Model Setup

We define our training video dataset with n_{Tr} videos as:

$$Tr = \{(V_i, X_i, l_i)\}_{i=1,\cdots,n_{Tr}}$$

where V_i denotes the i^{th} video, which is given a label of video name l_i. Each video is described by the feature vector $\mathbf{x}_{i,j}$. As one video contains three types of features

$$X_i = \{\mathbf{x}_{i,j}\}_{j=1,2}$$

where $j = 1$ indicates CNN feature, $j = 2$ indicates MFCCs feature, and a single classification label, this is a typical multi-instance learning problem.

In addition, we define a test set with n_{Te} videos:

$$Te = \{(V_i, X_i, l_i)\}_{i=1,\cdots,n_{Te}}$$

where symbols are similarly defined.

Then we consider the following methods to do classification.

3.2.1 Two-layer Neural Network

First, we apply a two-layer neural network on this problem. In the first layer, we train the weight and bias of two features respectively. Here we use $a_{1,j}$ as the output of the first layer and also the input of the second layer for the j-th feature

$$a_{1,j} = \sigma(W_j^{(1)} X_{\cdot,j} + b_j^{(1)})$$

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

where $W_j^{(1)}$ indicates the weight matrix of the first layer, $b_j^{(1)}$ indicates the bias vector of the first layer, $X_{\cdot,j}$ indicates the matrix form of the feature j. Then the cost function which is composed of total loss of training data and an l-2 penalty

$$\sum_{i=1}^{Tr} \ell(\widehat{y_i}, y_i) + \frac{\lambda_1}{2} \sum_{j=1}^{2} \parallel W_j^{(1)} \parallel_F^2$$

is minimized to train the model.

The second layer is a fusion layer. We use a_2 to indicate the output of the second layer.

$$a_2 = \sigma(W_1^{(2)} X_{\cdot,1} + W_2^{(2)} X_{\cdot,2} + b^{(2)})$$

Then we minimize the loss function which is similar to the one in the first layer.

However, in experiment we found that this model may fail in a small training data since the training target vector y_i has too many zero terms and the dimension of the vector is relatively high compared to the number of training samples. The parameters of this model is also to large without some special regularizers. By the impact of L-2 penalty, the weights may converge to a local minimum and lose the ability to do categorization. In practice, this method is useless.

3.2.2 Simple Logistic Regression

Facing the failure of two-layer neural network , we decided to try simpler models. Our simple logistic regression model assigned each model a logistic regression:

$$s^i(x) = \sigma(W_i X_{i,\cdot} + b_i),$$

in which $X_{i,\cdot}$ is the feature of video v_x and $s^i(x)$ is the score of "instance x belongs to category i". In training procedure, if x really belongs to category i, the regression target of $s^i(x)$ is set to 1. Otherwise the regression target of $s^i(x)$ is set to 0. In predicting procedure, the answer is:

$$\tilde{y} = arg\max_i s^i(x)$$

In experiment part, we show the performance of this method. We try to add some other information provided by the data in order to get better performance.

3.2.3 Tree-based logistic regression

Method While the normal regression methods like linear regression and logistic regression method ignore the rich information from the relationship between the labels, we propose a simple tree-based logistic regression method, which is motivated by [3].
As we know, all events and activities could be organized as a hierarchy structure, and FCVID dataset also provides us with the hierarchy of its 239 labels. In [5], they argue that their special regularizers could learn the relationships between different classes. However, a more natural method is to facilitate the tree form of the hierarchy as a prior to constrain the learning process. We build a two-class classifier for each class (including the class labels in original data and the ancestors of those labels) except the root class. So all meaningful class labels have their own classifier to tell whether an instance belongs to it. All the classifier is modeled by a logistic regression, so it could produce a score in [0,1].
The original idea of the predicting procedure is very simple: starting from the root node and choose the child node with highest score among all children node. If we arrived at a node with no child(bottom node), the label of the node is our prediction. Here we do prediction in the improved way by using an additional score to the bottom node(abbreviated to a-score in the following article),

$$ascore(v_x^i) = (\prod_{l \le k(i)} s_l^i)^{1/|L|}$$

where s_l^i indicates the score of the l-th node in the path from root to the i-th bottom node, L indicates the set of l which satisfies $l \le k(i)$, $k(i)$ indicates the number of nodes in the path from root to the bottom node through tree, which only depends on the number of the category i.
Our target is to find the i that maximizes the function

$$\mathscr{L}(i) \triangleq \log(ascore(v_x^i)) = \frac{1}{|L|}\sum_{l \in L} s_l^i$$

Then we make the prediction $x = i$. The advantage of introducing a new score is that the model can avoid misclassifying v_x in l-th hierarchy if $s_l^i \approx s_{l'}^i$, thus a video with confusing parent node score can be predicted to a wrong category. The relationship of classes is used here to improve the performance.

Training The training of this model is not trivial: we may have more than 300 classes in the hierarchy(including those ancestors), but each instance could belong to less than 10 classes. If all instances are used to train all classifiers, most classifier may produce 0 all the time because the positive samples are too few. So we set a negative sampling rate ϵ to reduce the amount of negative samples.
Each time an instance is fed into the model to update the classifiers. If it belongs to label c, the target of the classifiers of label c and its ancestors are set to 1, while the rest of them are set to 0. So the classifiers of only ϵ*100% of the other classes will be put into real training. We simply use the stochastic gradient descent to train the classifiers. More details of the training is described in the experiment part.

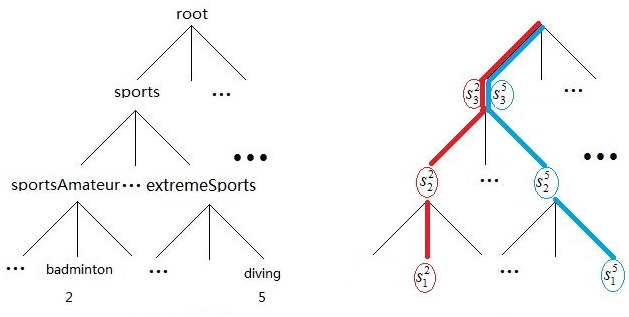

Figure 1: Here we use this example to explain ascore. The category 'Badminton' has a hierarchy structure: root-sports-sportsAmateur-badminton, and the category 'Diving':root-sports-extremeSports-diving, $score(x = 2) = \sqrt[3]{s_1^2 * s_2^2 * s_3^2}$, $score(x = 5) = \sqrt[3]{s_1^5 * s_2^5 * s_3^5}$. If $score(x = 2) > score(x = 5)$, we judge that v_x more likely to be a badminton video.

Algorithm 1 Pseudo-code describing the predicting method of tree-based regression.

Input: The trained tree(w,b of each layer);
The element v_j to be categorized.
Output: v_j's category x

for i=1 to 239 **do**
  while l not reach 'root' **do**
    $s_l^i \leftarrow \sigma(W_l^i X + b_l^i)$;
  end while
end for
Initialize $max\_score = -1$;
for i=1 to 239 **do**
  $ascore(v_j^i) = (\prod_{l \leq k(i)} s_l^i)^{1/|L|}$;
  if $ascore(v_j^i) > max\_score$ **then**
    $x \leftarrow i$;
    $max\_score \leftarrow ascore(v_x^i)$;
  end if
end for

3.3 Verification of tree-based model: Sampling k-NN

In order to verify the good performance of tree-based model, we do an additional experiment of k-nearest neighbours method. First, we pick up randomly k training samples of each category. Pay attention that k can't be set too large(because the computing resource is limited and k-NN is a resource-consuming algorithm) or too small(because the performance can be bad). Then the distance between the test data and the center of each category is computed. And we choose the category that minimizes the distance.

Algorithm 2 Pseudo-code describing the training method of tree-based regression.

Input: The initial tree (w,b of each layer);
The element v_j with ground truth label l_j.
Output: The trained tree (w,b of each layer).

$p \leftarrow ancestors(l_j)$
for l_i in p **do**
  Train classifier of l_i with instance v_j, and the target of $\sigma(w_i v_j + b_i) = 1$.
end for
$np \leftarrow$ sample $\epsilon \times 100\%$ of categories other than p.
for l_i in np **do**
  Train classifier of l_i with instance v_j, and the target is $\sigma(w_i v_j + b_i) = 0$.
end for

Algorithm 3 Pseudo-code describing the sampling k-NN algorithm.

Input: The feature vector of random sampling points $\{p_{l,i}\}_{l=1,2,...,k}$ of i-th category;
The element x to be categorized.
Output: x's category c
for i=1 to 239 **do**
$dist(c_i, x) \leftarrow \frac{1}{k} \sum_{l=1}^{k} dist(p_{l,i}, x)$;
end for
$c \leftarrow argmax(dist(c_i, x))$;

4 Experiments

4.1 Settings

Though FCVID provided us with a large amount of data, we do not have enough computational resources to train our model on the whole dataset. Even the test on the whole dataset costs a lot of time. So we randomly sample 1000 data from test set as our experimental test set and 1000 data from training set as our experimental validation set. The training set is sampled from the original training set. The random seed is set in our code, which ensures that every run produces the same data.

The classifiers ares trained by stochastic gradient descending optimizer with learning rate setting to 0.01. The negative sampling rate is set to 0.03. We use top-1 accuracy rather than mAP to compare the performance of our models, which is stricter than mAP. The results are the average of 5 runs.

4.2 Result

Firstly, we use static CNN features and MFCC features as the input of our model. The result (top-1 accuracy) is shown in Table 3.

Training Set Size	1000	3000	5000
Logistic Regression	30.2%	39.5%	41.9%
Tree-based Logistic Regression	30.9%	40.7%	44.4%

Table 3: The experiment results(top-1 accuracy) of two regression model. The feature is the concatenation of static CNN feature and MFCCs feature.

The result of the experiments shows that our tree-based logistic regression method always improves the accuracy of the classification. This empirical results show that the tree-based score is more robust than normal logistic regression. This proves that the hierarchical structure really helps. We do not report the result of k-NN because we find in the experiment that this method suffers from the curse of dimensionality. With almost the same amount of data we use in training logistic regression methods, k-NN can't provide any meaningful result. The accuracy is less than 1%. In theory, this result can be explained by the reason that the vectors of CNN feature and MFCCs feature are not in the Euclidean space.

Secondly, we simply use static CNN feature to train the models in order to explore the relationship and the value of the combination of two different features. As is reported in [5], the audio feature is not good enough to classify the videos alone. Our experiment confirms this assertion. We find the phenomenon that no matter what kind of regression methods proposed above is applied, the accuracy is lower than 2% with audio feature alone. However, the experiment of the model using static CNN feature alone shows that MFCCs feature really helps to improve the performance, thus proving the complementarity of two types of feature, as it is shown in Table 4.

Training Set Size	1000	3000	5000
Logistic Regression	29.2%	38.8%	42.1%
Tree-based Logistic Regression	29.4%	40.6%	43.9%

Table 4: The experiment results of two regression model(mAP). The feature is static CNN feature.

Finally, we use the matric of mAP to test our tree-based model in order to compare it with the previous work. In fact, it is meaningless in some way to compare because of the huge gap in the scales of training samples(we only used 5000 training samples). The results is shown in Table 5. Though there seems to be a great difference between our method and the state-of-the-art methods, those deep neural network methods need large amounts of manually labeled data and expensive GPUs and long time to trained. The advantage of our model is that we can finish training on a Thinkpad x260 laptop in half an hour.

Model	mAP on FCVID
Tree-based regression(5000 samples)	57.1%
DNN[5]	73.0%
OSF Networt[9]	76.5%

Table 5: Comparing our methods to the state-of-the-art models trained with all training set.

5 Conclusions

In this project, we propose to use tree-based logistic regression to solve video categorization problem with few training instances. Our experiments show that our method is able to utilize the information from the hierarchical structures of the labels to improve the performance of our model. With only a few training samples, our method can achieve a good classification performance. This property of our model does good to the status quo where labeled data is expensive and the computational resources are limited. In the future, we might try to explore our method to the larger scale dataset to get better result.

References

[1] Courtenay V Cotton and Daniel PW Ellis. Subband autocorrelation features for video soundtrack classification. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8663–8666. IEEE, 2013.

[2] Jia Deng, Nan Ding, Yangqing Jia, Andrea Frome, Kevin Murphy, Samy Bengio, Yuan Li, Hartmut Neven, and Hartwig Adam. Large-scale object classification using label relation graphs. In *European Conference on Computer Vision*, pages 48–64. Springer, 2014.

[3] Jia Deng, Jonathan Krause, Alexander C Berg, and Li Fei-Fei. Hedging your bets: Optimizing accuracy-specificity trade-offs in large scale visual recognition. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3450–3457. IEEE, 2012.

[4] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2625–2634, 2015.

[5] Yu-Gang Jiang, Zuxuan Wu, Jun Wang, Xiangyang Xue, and Shih-Fu Chang. Exploiting feature and class relationships in video categorization with regularized deep neural networks. *arXiv preprint arXiv:1502.07209*, 2015.

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

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[8] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3551–3558, 2013.

[9] Zuxuan Wu, Yanwei Fu, Yu-Gang Jiang, and Leonid Sigal. Harnessing object and scene semantics for large-scale video understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3112–3121, 2016.

[10] Zuxuan Wu, Ting Yao, Yanwei Fu, and Yu-Gang Jiang. Deep learning for video classification and captioning. *arXiv preprint arXiv:1609.06782*, 2016.