ABSTRACT
We consider the problem of large-scale video classification. Our attention is focused on online video services since they can provide rich cross-video signals derived from user behavior. These signals help us to extract correlated information across videos which are co-browsed, co-uploaded, co-commented, co-queried, etc. Majority of the video classification methods omit this rich information and focus solely on a single test instance. In this paper, we propose a video classification system that exploits various cross-video signals offered by large-scale video databases. In our experiments, we show up to 4.5% absolute equal error rate (17% relative) improvement over the baseline on four video classification problems.

Index Terms— Video classification, Video databases

1. INTRODUCTION
More and more videos are uploaded to online video services every day. Automatic classification of these videos is crucial for monitoring, indexing and retrieval purposes. Motivated from this fact, recently there has been a growing interest on developing web-scale algorithms for video classification. In [1], authors use motion and color features with Hidden Markov Models to classify sports videos into four categories: ice hockey, basketball, football, and soccer. Audio features are used with Multi Layer Perceptron in [2] for real-time video classification. In [3], authors use motion features with Support Vector Machines to identify the quality of a video which is categorized into four classes: blurred, shaky, inconsistent and stable. In [4], authors use cinematic, color and motion features with decision trees to identify video genre such as cartoons, commercials, music, news, and sports. Average shot length, color variance, motion and lighting features are used in [5] to categorize movies into multiple genres: comedies, action, dramas and horror.

Previous work on video classification has two major limitations to be used on large-scale video databases. First, training and testing are generally performed on a controlled dataset. In a recent study, Zanetti et al. showed that most existing video categorization algorithms do not perform well on general web videos [6]. Furthermore, the size of the data-sets are relatively small when compared to the scale of online video services. Second, the algorithms treat each test video independently. We believe that online video services carry important cross-video signals that could be exploited to boost video classification performance. For instance, two videos that are uploaded by the same person, might share common information. Therefore, one should investigate whether the correlated information between multiple videos could be used for better video classification. In the literature, relatively little work address this problem. In [7], authors start with a small manually labeled training set and expand it using related YouTube videos. In [8], authors propagate the labels of photos with high confidence to its neighbors for photo annotation. In [9], human provided labels are “smoothly” propagated to the neighboring samples along a manifold for video face recognition.

In the label propagation algorithms, defining and handling multiple sets of neighbor samples, and finding their relative importance weights still remain as open problems. In this work, we propose a generic video classification system that exploits various cross-video signals offered by large-scale video databases. Specifically, the correlated information between co-browsed, co-uploaded, co-commented and co-queried videos is modeled. For each classification problem, the correlated information across related videos is represented by the video classification scores obtained by a classifier. Then the optimum fusion strategy for the original video classification score and the other cross-video signals is learned for each classification problem separately. In fact, the weights of the cross-video signals, which are obtained by the learning algorithm, are found to vary across different classification problems.

The paper is organized as follows. In Section 2, video features and the standard video classifier are described. Furthermore, the method that fuses the cross-video signals extracted from co-browsed, co-uploaded, co-commented and co-queried videos is introduced. In Section 3, experimental
results on four classification problems are presented. Finally, concluding remarks are explained in Section 4.

2. METHOD

In this section, the components of the proposed video classification algorithm is described in detail. We begin by providing a brief system overview which is presented in Figure 1. Given a classification problem, a classifier is trained based on the features extracted on training videos. This classifier is denoted by $C$. Then given a test video, several related sets of videos are obtained based on user behavior. The classification scores extracted from these related video sets are referred to as cross-video signals. We consider four related video sets and corresponding cross-video signals: co-browsed ($p_b$), co-commented ($p_c$), co-queried ($p_q$), co-uploaded ($p_u$). A test video is classified with the score $p_f$ by fusing its cross-video signals ($p_b$, $p_c$, $p_q$ and $p_u$) with the original test video score $p_o$. The fusion is defined by the weighted summation where the weight vector is represented with $w$. The optimum weight vector is also found by minimizing the final classification error on the training set.

2.1. Video Features and Classifier

The following features are extracted from the videos:

- **Color**: 4×4-bin hue and saturation histogram as well as the mean and variance of the color intensities for each channel are computed on each video frame. Furthermore, the difference between the mean color intensities inside and outside of the central rectangle are computed.

- **Edge**: On each frame, edges are detected using Canny edge detection algorithm. On the edge response image, fraction of edge pixels, edge direction histogram, and the mean edge energy for pixels are computed.

- **Texture**: Textron histogram is computed using the method of [10]. Hierarchical K-Means [11] is used to build the textron vocabulary where the vocabulary size is 1K.

- **Face**: Faces are detected using an extension of AdaBoost classifier [12]. The number of faces and the ratio of largest face area to the image area are computed on each video frame.

- **Motion**: The Cosine distance between the color histograms of two consecutive frames are computed. Furthermore, a binary value is extracted for each frame indicating the shot boundaries. The shot boundaries are detected using an algorithm similar to [13].

- **Audio**: Audio volume and the coefficients of the 32-bin audio spectrogram is computed around the video frames.

- **Visual Words**: Interest points are detected on each video frame using Laplacian-of-Gaussian filters. Similar to SIFT [14], 118 dimensional Gabor wavelet texture features are computed on each interest point. The histograms of these features are accumulated across video frames. Hierarchical K-Means [11] is used to build the Gabor wavelet texture feature vocabulary where the vocabulary size is 20K.

Except the histogram of visual words, each feature sequence is treated as a time series and 1D Haar wavelet transform is applied at 8 scales. Maximum, minimum, mean and variance of the wavelet coefficients are computed. The statistics of feature sequences and the histogram of visual words are concatenated to obtain a 31408 dimensional global feature representation of a video. This type of feature representation was also employed in [7, 15].

Given a classification problem, the above features are extracted from the videos in the training set. Then, an AdaBoost classifier [16] is learned using the training feature instances. The classifier is denoted by $C$. When applied on a test instance, $C$ produces a classification score, denoted by $p_o$, assuming a binary classification problem.

2.2. Fusing the Cross-Video Signals

Cross-video signals are extracted from a group of videos which are related to the test video. Two videos can be related via several ways. In this study, four cross-video relation sources are considered: Two videos are assumed to be

- **Co-Browsed** if more than a certain number of users browsed both of them within a session.

- **Co-Uploaded** if the same user uploaded both videos within a certain time window.

- **Co-Commented** if more than a certain number of users commented on both of them.

- **Co-Queried** if more than a certain number of users clicked on both videos in response to the same queries.

Given a test video and the sets of related videos, the classifier $C$ is applied on all of these videos independently to obtain the classification scores. The classification score of the test video is denoted by $p_o$. Then, the median classification score is computed for each set of related videos. The median scores for co-browsed, co-uploaded, co-commented and co-queried videos are denoted by $p_b$, $p_u$, $p_c$, $p_q$, respectively. The final classification score, $p_f$ is obtained by taking the weighted sum of these scores. For ease of notation, the score vector
is represented with $p = [p_o, p_b, p_c, p_q]^T$. Similarly the weight vector is represented with $w = [w_o, w_b, w_c, w_q]^T$.

$$p_f = w^T p$$ \hspace{1cm} (1)

Given the training instances (with labels $\{y_i\}$) and corresponding classification score vectors $\{p_i\}$, the task is to learn the weight vector which minimizes the classification error:

$$\min_w \sum_i \log(1 + \exp(-y_i w^T p_i)) + \frac{1}{2} ||w||^2. \hspace{1cm} (2)$$

The minimization is performed using trust region Newton method [17]. Note that while performing the optimization, the bias term $w_{bias}$ and 1 is appended to $w$ and $p$, respectively.

### 3. EXPERIMENTAL RESULTS

The proposed algorithm is tested on the following video classification problems:

- **Quality**: In this problem, the task is to determine whether a video is high or low quality.

- **Porn**: In this problem, the task to determine whether a video has pornographic content or not.

- **Racy**: In this problem, the task to determine whether a video has racy content\(^1\) or not.

- **Speech**: In this problem, the task to determine whether a video contains speech or not.

The number of positive and negative samples on training and testing sets are provided in Table 1. For each classification problem, the number of stumps used in the AdaBoost classifier are presented in Table 2.

Table 1. The number of positive and negative instances on training and testing sets of various classification problems.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Train</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Positive</td>
<td>Negative</td>
</tr>
<tr>
<td>Quality</td>
<td>32566</td>
<td>32434</td>
</tr>
<tr>
<td>Porn</td>
<td>32748</td>
<td>32215</td>
</tr>
<tr>
<td>Racy</td>
<td>32500</td>
<td>32215</td>
</tr>
<tr>
<td>Speech</td>
<td>1500</td>
<td>752</td>
</tr>
</tbody>
</table>

Table 2. Equal Error Rates (%) on various classification problems using baseline ($p_o$) and proposed method ($p_f$).

<table>
<thead>
<tr>
<th>Problem</th>
<th># of stumps</th>
<th>$p_o$</th>
<th>$p_f$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quality</td>
<td>585</td>
<td>19.6</td>
<td>17.0</td>
</tr>
<tr>
<td>Porn</td>
<td>3000</td>
<td>18.5</td>
<td>14.3</td>
</tr>
<tr>
<td>Racy</td>
<td>3000</td>
<td>26.6</td>
<td>22.1</td>
</tr>
<tr>
<td>Speech</td>
<td>500</td>
<td>4.5</td>
<td>3.7</td>
</tr>
</tbody>
</table>

- **Porn**: In this problem, the task to determine whether a video has pornographic content or not.

- **Racy**: In this problem, the task to determine whether a video has racy content\(^1\) or not.

- **Speech**: In this problem, the task to determine whether a video contains speech or not.

The Equal Error Rate (EER) performance based on the original video classification scores ($p_o$) and the output of proposed system ($p_f$) are presented in Table 2. The proposed system consistently yields lower EER on four classification problems. The largest absolute gain of 4.5% EER (17% relative) is obtained on the Racy video classification. This problem is a quite difficult to solve by considering only the test video content. However, using the cross-video signals bring

\(^1\)http://www.youtube.com/t/community_guidelines
Table 3. Weights learned using (2) on various classification problems.

<table>
<thead>
<tr>
<th>Problem</th>
<th>( w_a )</th>
<th>( w_b )</th>
<th>( w_c )</th>
<th>( w_d )</th>
<th>( w_{bias} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quality</td>
<td>0.54</td>
<td>0.09</td>
<td>0.14</td>
<td>0.32</td>
<td>0.28</td>
</tr>
<tr>
<td>Porn</td>
<td>0.10</td>
<td>0.28</td>
<td>0.47</td>
<td>0.44</td>
<td>0.29</td>
</tr>
<tr>
<td>Racy</td>
<td>0.37</td>
<td>0.43</td>
<td>0.16</td>
<td>0.39</td>
<td>0.23</td>
</tr>
<tr>
<td>Speech</td>
<td>0.72</td>
<td>0.21</td>
<td>0.14</td>
<td>0.21</td>
<td>0.06</td>
</tr>
</tbody>
</table>

The weights which are learned using (2) are presented in Table 3 for various classification problems. These weights indicate the relative importance of different cross-video sources. For Quality and Speech problems, the original score has the highest weight. However, for the Porn and Racy problems, median of the co-commented and co-browsed video scores has the highest weights, respectively. This shows that the median scores of the related videos could provide important (possibly more important than the original video score) source of information for classification.

4. CONCLUSIONS AND FUTURE WORK

We presented a system for generic video classification which uses cross-video signals to improve classification performance. The cross-video signals are defined between related videos. In this work, four related video sources are considered: co-browsed, co-uploaded, co-commented, co-queried. In fact, the proposed system allows to integrate other related video sources as well. The system is tested on four large-scale classification problems and it improved the classification performance up to 4.5% absolute EER (17% relative). The future work will include extending the system for multi-class classification.

5. REFERENCES


