MULTI-MODAL VIDEO CONCEPT EXTRACTION USING CO-TRAINING

Rong Yan

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA, 15213

Milind Naphade[*]

IBM TJ Watson Research Center
19 Skyline Drive
Hawthorne, NY, 10532

Abstract

For large scale automatic semantic video characterization, it is necessary to learn and model a large number of semantic concepts. A major obstacle to this is the insufficiency of labeled training samples. Semi-supervised learning algorithms such as co-training may help by incorporating a large amount of unlabeled data, which allows the redundant information across views to improve the learning performance. Although co-training has been successfully applied in several domains, it has not been used to detect video concepts before. In this paper, we extend co-training to the domain of video concept detection and investigate different strategies of co-training as well as their effects to the detection accuracy. We demonstrate performance based on the guideline of the TRECVID'03 semantic concept extraction task.

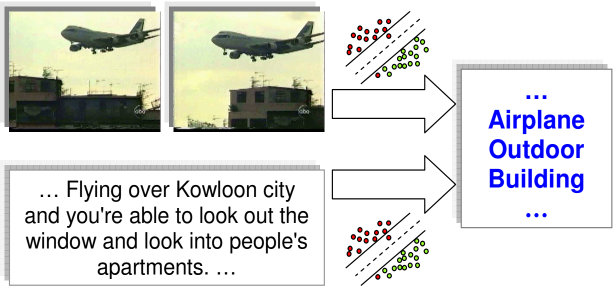

… Flying over Kowloon city and you're able to look out the window and look into people's apartments. …
…
Airplane
Outdoor
Building
…

Fig. 1. Illustration of detecting semantic concepts from video sequences. Each video shot is associated with multi-modal information including both text/speech transcript and visual frames. The semantic concepts can be detected by combining the outputs of multiple unimodal classifiers.

1. INTRODUCTION

Increasingly, the detection of a large number of semantic concepts is being seen as an intermediate step in enabling semantic video search and retrieval[1, 2]. These semantic concepts cover a wide range of topics that can be roughly categorized as objects, sites, events, and specific personalities and named entities. The main idea of semantic concept detection is to treat it as a statistical learning problem. For each video shot, the associated concepts can be detected using multiple unimodal classifiers or multimodal classifiers [3] based on visual, audio and text/speech features. Using a large annotated corpus, these concepts can be learnt if sufficient number of training samples exist. Unfortunately, annotation is a labor-intensive process and the number of labeled video samples is usually not enough for most semantic concepts. Typically annotating 1 hour of video divided into shots, with a lexicon of 100 semantic concepts can take anywhere between 8 to 15 hours. The problem is further worsened for a large number of concepts which appear infrequently.

One way to deal with insufficient labeled data is to apply the semi-supervised learning algorithms which attempt to leverage a large amount of unlabeled data set to boost the classification accuracy along with a small amount of labeled data. The multiple modalities of the video stream further suggest considering the multi-view setting which explicitly split the feature space into multiple subsets, or views. Combining semi-supervised learning and multi-view setting offers a more powerful way to leverage unlabeled data. Co-training[4], proposed by Blum and Mitchell, is one of the most well-known multi-view semi-supervised learning algorithms. The co-training algorithm starts with two initial classifiers learned from each view separately. Both classifiers are then incrementally updated in every iteration using an augmented labeled set, which includes additional unlabeled samples with the highest classification confidence in each view. The idea of co-training is to incrementally update the classifiers of multiple views which allows the redundant information across views to improve the learning performance. The co-EM algorithm[5] can be viewed as a probabilistic version of co-training. It requires each classifier to provide class probability estimation for all unlabeled data and use them to rebuild the classifiers of the other views. Goldman and Zhou[6] proposed a variant of the co-training algorithm which uses two different supervised learning algorithms to label the unlabeled examples instead of relying on the explicit feature split to two independent views. This class of co-training type algorithms has been successfully applied to a variety of real-world domains, from text classification[5], natural language processing[7], web page classification[4] to visual detection[8].

However, these co-training type algorithms have not been successfully applied in the domain of video concept detection before, although it has been considered a potentially applicable domain by Blum et al[4]. In this paper, we extend the usage of co-training to the task of video concept detection. We also investigate different co-training strategies with respect to combination across the multiple modalities and feedback during update iterations. We examine these strategies in the context of the NIST TRECVID Concept De-

[*] This material is based upon work funded in part by the U.S. Government. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the U.S. Government. We also thank IBM team for providing the low-level features.

0-7803-9332-5/05/$20.00 ©2005 IEEE

tection task using the TRECVID 2003 annotated corpus[1].

2. THE CO-TRAINING ALGORITHM

The co-training algorithm belongs to a class of algorithms that combine semi-supervised learning and multi-view learning into one unified framework. Formally, the goal for co-training is to learn a classifier $f(x)$ using a small amount of labeled data L : $\{(x_1,y_1),...,(x_n,y_n)\}$, and a large amount of unlabeled data U : $\{x'_1,...,x'_m\}$. The feature space can be split into two disjoint views V_1 and V_2, and thus each labeled example (x_i,y_i) can be decomposed into (x_{i1},x_{i2},y_i) where x_{i1} and x_{i2} are the features over the views V_1 and V_2 respectively. The classifier learned from view V_j is denoted as $f_j(x)$.

The approach of co-training is to incrementally update the classifiers of multiple views which allows the redundant information across views to improve the learning performance. For each view V_j, the classifier $f_j(x)$ is first initialized by learning a few labeled examples L_t. At each iteration, the algorithm will select a batch of unlabeled data from the unlabeled set U to incorporate into the pool of labeled data L_t. Typically these additional unlabeled data selected are those with the highest prediction confidence for each view. Each classifier $f_j(x)$ is then updated from the augmented labeled data set. This process is iterated until a few iterations. Finally, weighted linear combination of the output classifiers $f_j(x)$ gives a single-view classifier $f(x)$.

The intuition of co-training is that the two classifiers can provide each other with additional automatically labeled data which might be as informative as some random noisy labeled examples. Based on the analysis of Nigam et al[5], the co-training algorithm can naturally leverage the feature split for the data set, of which the views V_j should be conditionally independent of each other in order to provide useful information. Actually, the assumption of conditional independence is reasonable in the task of video concept detection because the text modality can be viewed as an approximately independent source of the visual modality. Therefore, it is of great interest to investigate the performance when applying co-training algorithm to the video concept detection. In the following, we discuss several co-training strategies with respect to both combination across the multiple modalities in step 3 of Figure 2 and feedback during update iterations in step 2c.

2.1. Combination Strategies

In the original co-training algorithm the combination across the multiple classifiers is still an open problem. We choose to combine the final two classifiers via linear weighted sum, however, the approaches to determine the weights can be different. In this paper, we study three possible combination strategies. The simplest strategy is set the weights to be equal, i.e. $w_1 = w_2$. But setting equal weights is not always the best choice because in our application not all the views are sufficient enough or equally relevant to capture the underlying concept. With help of an additional validation data set, we can apply learning algorithms to adjust the weights accordingly. As the second strategy, we choose the Powell's direction set method[9] to learn the combination weights which maximize the average precision on the validation set. The direction set method is a global optimization method that searches the function's minimum along N mutually conjugate directions without computing the gradient directions directly. This property is critical because in our case computing the gradients for average precision is out of question. As an approximation for the direction set method, the third strategy simply set the weights w_i to be the average precision of view V_i on the validation set. This strategy rewards the better performing classifier.

```
Input Two views $V_1$ and $V_2$, labeled data $L$ including training data $L_t$ and validation data $L_v$, unlabeled data $U$, the number of iterations $T$

Co-Training
  1. Create the classifier $f_1^0$ and $f_2^0$ using $L_t$ on $V_1$ and $V_2$
  2. Let $L_{t1} = L_{t2} = L_t$. For $t = 1,2,....T$
      (a) Remove $n_p$ examples with largest $f_i^{t-1}(x')$ and $n_n$ examples with smallest $f_i^{t-1}(x')$ from the unlabeled set $U$, $i = 1,2$
      (b) Label the selected examples using $f_1^{t-1}$ and $f_2^{t-1}$
      (c) Add the examples to the training sets $L_{t1}$ and $L_{t2}$
      (d) Create the classifier $f_1^t$ using $L_{t1}$ on $V_1$ and $f_2^t$ using $L_{t2}$ on $V_2$
  3. Combine $f^T = w_1 f_1^T + w_2 f_2^T$ (using $L_v$)
```

Fig. 2. The co-training algorithm

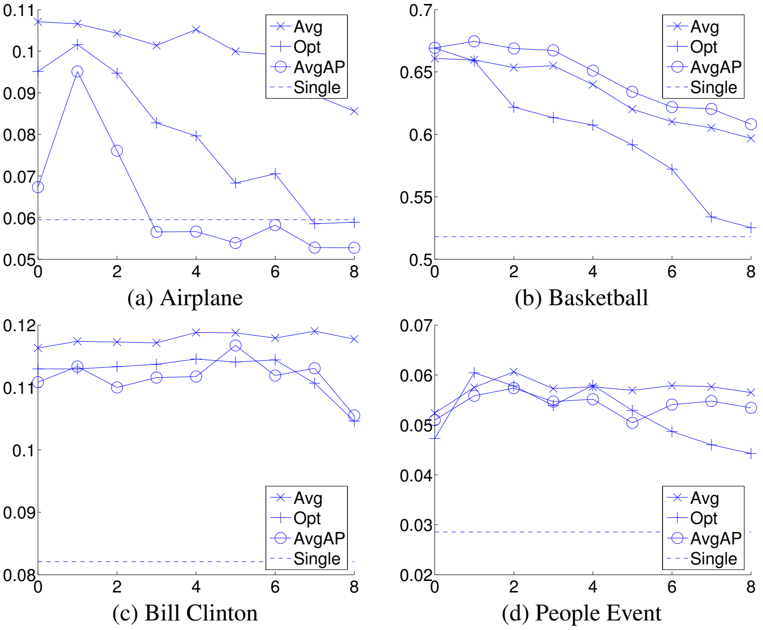
2.2. Feedback Strategies

For each round, co-training has to feed back a certain amount of automatically labeled data to the training set. We explore two strategies to incorporate the additional labeled data. One strategy is called "cross-view feedback" which incrementally incorporates the unlabeled data selected only from the other view. For example, for the view V_1 we only consider adding the unlabeled examples selected based on the view V_2 per round and vice verse. In contrast, the other strategy is called "multi-view feedback" which incrementally incorporates the unlabeled data selected from both views. These two approaches have their pros and cons. Cross-view feedback keeps the most informative labeled data for each view, while multi-view feedback allows more labeled data to be added per round. Original version of co-training use the latter one for combination.

3. EXPERIMENTS

3.1. Experimental Setting

To evaluate the performance of the co-training algorithm for video concept detection, we follow the guideline of TRECVID03 semantic concept extraction task [1] to design our experiments. TRECVID is an annual video retrieval competition organized by National Institute of Standards and Technology(NIST). The goal of the concept extraction task is to detect the presence or absence of a video concept in the reference video shots over a 65 hour news video corpus. In our experiments four concepts are selected from TRECVID '04, i.e. *Airplane, Basketball, Bill Clinton, People Event*. These concepts cover a broad range of interesting topics in news video and they could be detected from the low-level features with reasonable accuracy. Note that we are running 4 separate binary classifications for the presence/absence of each concept.



(a) Airplane

(b) Basketball

(c) Bill Clinton

(d) People Event

Fig. 3. Comparison of three combination strategies against the number of iterations in co-training. Each subgraph corresponds to a specific video concept. See text for more details.

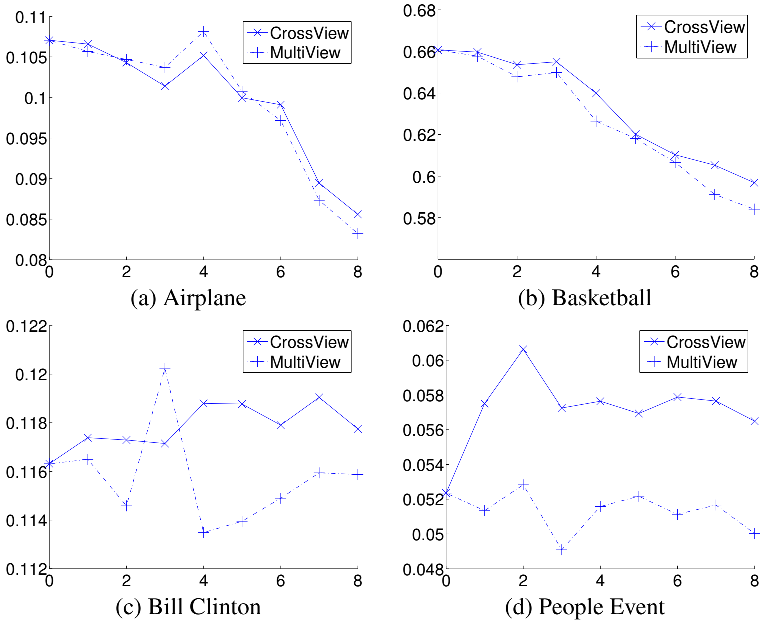
Our data collection is constructed as follows. First, the news video collection is randomly partitioned into four data sets, i.e. 30% of the data as the training set, 50% of the data as the unlabeled set, 5% of the data as the validation set, 15% of the data as the testing set. In the collection, each video shot is associated with the truth annotations over every concept[1, 10]. To collect training data for each concept, we first pick all the positive examples from the training set and downsample all the other negative examples to keep the ratio between positive and negative examples to be 1:5. This ratio is chosen so as to provide a reasonable trade-off between the performance and the running time. Finally, we collected 167 shots for airplane, 436 shots for basketball, 192 shots for Bill Clinton and 756 shots for People Event. In the following experiments, all of the unlabeled data and testing data will be used with their labels discarded. For each video shot, we extract two types of low-level features, i.e. 166 dimensional color correlogram feature vector in HSV color space for each keyframe of the shot and 35,640 binary word presence features of automatic speech transcripts/closed caption.

Since the number of positive samples is usually much smaller than negative data in our task, the classification accuracy is not a preferred performance measure. Alternatively, NIST defines non-interpolated average precision over a set of retrieved documents (shots in our case) as a measure of retrieval effectiveness. Let R be the number of true relevant documents in a set of size S; L the ranked list of documents returned. At any given index j let R_j be the number of relevant documents in the top j documents. Let $I_j = 1$ if the j^{th} document is relevant and 0 otherwise. Assuming $R < S$, the non-interpolated average precision **(AP)** is then defined as $\frac{1}{R}\sum_{j=1}^{S}\frac{R_j}{j} * I_j$.

3.2. Performance Evaluation

To provide a fair comparison, we use the same set of training data to produce the initial classifiers for all of the algorithms. SVM^{Light} is adopted as the underlying classifier where the linear kernel is applied for text features and the RBF kernel for visual features. Cross validation is used to decide the learning parameters and the cost factor that achieve the best average precision on the training data. We choose the cross-view feedback and equal-weight averaging as the default strategies. All the experiments run up to the 8 iterations. In each iteration, we select additional unlabeled data as much as 10% of the training data. Therefore at the end of the learning process, the number of unlabeled data is about the same as the number of training data.

The first series of experiments are designed to compare the performance of co-training under various combination strategies. The results are depicted in Figure 3. Each subgraphs compares three combination strategies, i.e. equal-weight averaging(**Avg**), learning weights using direction set methods(**Opt**) and setting weights to be average precisions on validation set(**AvgAP**) with the best single-view classifiers using only training data. Generally speaking, we observe that co-training has the potential to improve the detection accuracy, especially when the number of additional training examples is small, although it is not statistically significant. For the concepts of Bill Clinton and People Event, the improvement is noticeable across various settings. Unfortunately, the performance will often be degraded after a larger number of unlabeled examples are incorporated. This is because a growing number of the “noisy” labeled data would finally overwhelm the fixed number of clean labeled data and thus corrupt the classification outputs. To determine a good early stopping point for co-training becomes a critical issue in practice, which we leave it for the future work. Next, we compare various combination strategies. Surprisingly, it shows that the equal-weight combination achieves the best performance in 3 of the 4 concepts. This might be caused by the fact that the validation set are too small to be representative and the measure of average precision is too sensitive to learn. But it also suggests that equal-weight combination is a robust combination method without any effect of the validation set’s quality.



(a) Airplane

(b) Basketball

(c) Bill Clinton

(d) People Event

Fig. 4. Comparison of two feedback strategies against the number of iterations in co-training. See text for more details.

Figure 4 depicts the learning curves on all of four concepts, each of which includes the curves using two type of feedback strategies, i.e. cross-view feedback(**CrossView**) and multi-view feedback(**MultiView**). We can observe that for all four concepts, co-training using cross-view feedback strategy is usually superior to co-training using multi-view feedback strategy though the latter strategy can provide more labeled data per iteration. A partial reason is that the cross-view feedback only keeps the most informative unlabeled examples and reduces the risk of deteriorating the classifiers with additional noisy labeled data. In our application, cross-modal feedback strategy turns out to be a better choice.

4. CONCLUSIONS AND FUTURE DIRECTIONS

In this paper, we apply co-training to the task of video concept detection and investigate different co-training strategies for leveraging unlabeled data along with small labeled multimodal data. The conditional independence between text modality and visual modality make it possible for co-training to improve the detection performance. The experimental results show that co-training can achieve higher average precision in some cases, but it also suffers from the noisy label problem when the underlying classifiers are not accurate enough and more than necessary number of unlabeled data are incorporated. Determining a better early stopping point is a useful remedy to address the problem in co-training. Among the choices of the combination and feedback strategies, equal-weight averaging and cross-modal feedback are demonstrated to be superior for most of the settings. In future, we would like to develop a better co-training style algorithm which can achieve robust performance without causing a large performance loss even in a highly noisy environment. We will also investigate interactive labeling strategies beyond the originally labeled samples so as to improve performance without adding a significant annotation overhead.

5. REFERENCES

[1] TRECVID: TREC Video Retrieval Evaluation, “http://www-nlpir.nist.gov/projects/trecvid,” .

[2] A. G. Hauptmann and et al, “Informedia at TRECVID 2003: Analyzing and searching broadcast news video,” in *Proc. of TRECVID 2003*, Gaithersburg, MD, 2003.

[3] M. R. Naphade, I. Kozintsev, and T. S. Huang, “On probabilistic semantic video indexing,” in *Proceedings of Neural Information Processing Systems*, Denver, CO, Nov. 2000, vol. 13, pp. 967–973.

[4] A. Blum and T. Mitchell, “Combining labeled and unlabeled data with co-training,” in *Proc. of the Workshop on Computational Learning Theory*, 1998.

[5] K. Nigam and R. Ghani, “Analyzing the effectiveness and applicability of co-training,” in *Proc. of CIKM*, 2000, pp. 86–93.

[6] S. Goldman and Y. Zhou, “Enhancing supervised learning with unlabeled data,” in *Proc. 17th Intl. Conf. on Machine Learning*, 2000, pp. 327–334.

[7] D. Pierce and C. Cardie, “Limitations of co-training for natural language learning from large datasets,” in *Proc. of EMNLP*, 2001.

[8] A. Levin, P. Viola, and Y. Freund, “Unsupervised improvement of visual detectors using cotraining,” in *Proc. of the Nineth IEEE International Conference on Computer Vision*, 2003.

[9] W. T. Vetterling W. H. Press, S. A. Teukolsky and B. P. Flannery., *Numerical recipes in C - 2nd ed.*, Cambridge University Press, Cambridge, NY, USA, 1994.

[10] C. Lin, B. Tseng, and J. Smith, “VideoAnnEx: IBM MPEG-7 annotation tool for multimedia indexing and concept learning,” in *IEEE International Conference on Multimedia and Expo*, 2003.