Selective Multi-cotraining for Video Concept Detection

Usman Niaz
EURECOM
Sophia Antipolis, France
niaz@eurecom.fr

Bernard Merialdo
EURECOM
Sophia Antipolis, France
merialdo@eurecom.fr

ABSTRACT

Research interest in cotraining is increasing which combines information from usually two classifiers to iteratively increase training resources and strengthen the classifiers. We try to select classifiers for cotraining when more than two representations of the data are available. The classifier based on the selected representation or data descriptor is expected to provide the most complementary information as new labels for the target classifier. These labels are critical for the next learning iteration. We present two criteria to select the complementary classifier where classification results on a validation set are used to calculate statistics for all the available classifiers. These statistics are used not only to pick the best classifier but also ascertain the number of new labels to be added for the target classifier. We demonstrate the effectiveness of classifier selection for semantic indexing task on the TRECIVD 2013 dataset and compare it to the self-training.

Keywords

Cotraining, bootstrapping, feature selection

1. INTRODUCTION

Semi-supervised classification benefits from large number of unlabeled data where the tedious task of human intervention to annotate the data is minimized. These methods start with only a few labeled examples where the unlabeled data can be annotated iteratively to augment training resources. Learning from partially labeled data is gaining importance as unlabeled data is cheaply available and has proven to decrease classification error significantly [1, 6, 2, 13].

Cotraining is a special case of using unlabeled examples which is useful when different feature representations of the data are available [1]. As the name suggests two learners trained on each data representation are used in unison to label training data for each other. This is done iteratively until training error is sufficiently reduced or all the data has been labeled. For final prediction both the classifiers are pooled together. There are however certain conditions that guarantee the successful working of the cotraining algorithm. Specifically it works when the different representations of data are different views and satisfy the cotraining properties. They should be conditionally independent given the class and each of them should be sufficient to learn, i.e. a learner on each view should predict the true class labels for the most part [1].

This comes naturally for some datasets where the features used are in fact complementary to each other and in combination reduce the classification error, however for others these conditions might be hard to satisfy. The text in the web pages and links to those pages are good examples of complementary features [1]. Another good example is multi-modal features where each feature represents a certain modality in learning form multimedia data. We benefit from the situation where more than two views of the data are available and we have the liberty of choosing the best complementary view for each classifier.

The goal is to add useful information to the annotation set of a classifier based on the cotraining principle to augment the training resources. This is achieved by adding positive examples that are expected (or ensure to some degree) to improve the final classification result. We propose two methods to select the most complementary view among a certain available possibilities based on some statistics calculated on the validation set. We take inspiration from [4] to add the most relevant examples that were the most confusing for the classifier. Instead of negatives we focus on adding only positive examples since we already have a large number of negative examples available. Among other possibilities a relevant positive example for a classifier is the one which was previously missclassified by the same classifier. We adapt this setting in the cotraining framework where the most relevant positive examples are added to the annotation set of the target classifier based on the predictions from another classifier. This will tame the target classifier especially for the categories with few positive examples and huge number of unlabeled examples available. The classifier which identifies the largest number of missclassifications of the target learner is considered the most complementary and this information is the basis for our two criteria presented later. One of the criteria proposed also selects automatically the number of new annotations to be added.

We have performed experimentation on the TRECVID 2013 [7] dataset for the video Semantic INdexing task and demonstrate the effectiveness of selecting the complementary classifier in comparison to self-training where the same

ICMR '14, April 01 - 04 2014, Glasgow, United Kingdom
Copyright 2014 ACM 978-1-4503-2782-4/14/04 ...$15.00.
http://dx.doi.org/10.1145/2578726.2578789 .

classifier is used to add new labels. A brief review of some of the recent works on cotraining and feature selection is detailed in the next subsection. The section 3 describes the criteria for selecting the best feature and the resulting cotraining method. Results and experimentation are presented in the section 4 which are followed by conclusions and some future research directions in section 5.

2. COTRAINING AND FEATURE SELECTION

Yan and Naphade [12] achieve improvement in video concept detection performance by using manual human effort to select from the most confident predictions for each cotraining view. In another work [13] they build classifiers separately on the newly labeled data and include them in the final prediction only if it performs well on the validation set. Du et al. [2] identify feature splits for cotraining that satisfy the cotraining independence and sufficiency properties for tire wear classification from images. They identify pairs of features by clustering based on mutual information and use classifier confidence to select unlabeled samples for the other. We follow similar principle but with different selection criteria, and our features are not bounded in pairs. Li et al. [3] perform feature selection for cotraining (FESCOT) by discarding the most irrelevant ones using classification accuracy on validation dataset. Features are iteratively disregarded whereas we select one complementary feature at each iteration.

3. PROPOSED APPROACH

Contrary to cotraining where we start with two views of the data distribution assuming that we have very little data to start the training with, here we have a good number of positive examples to start with. The number of negative examples is manifold of the number of positives. All the descriptors we are using are strong visual classifiers built on powerful visual descriptions. We try to add more annotations to the positive examples of a descriptor using information from another descriptor. This other descriptor is selected out of the available ones which is expected to bring the most information to the target descriptor. The information brought is new positive annotations which are then used to re-train the classifier using the target descriptor.

3.1 Selective Multi Cotraining

A certain number of descriptors are available to start with which can very well be multimodal representation of the data. Each descriptor is represented by a classifier which classifies or labels a video frame. Essentially this is done by assigning a score value to the frame. Our development data is divided into training and validation part. The final predictions are done on the test part which is independent of the development set.

For the target descriptor we try to find the descriptor that brings the most valuable information in terms of annotations. This is judged by the classification performance of the descriptors on the validation set. In the next iteration of the cotraining the training and validation sets are re-labeled for each descriptor and thus we start over the selection process for every descriptor with the newly obtained data. We call this process as Selective Multi Cotraining. It is important to mention here that in this selective multi cotraining features do not work in pairs, rather for each feature the most complementary is selected at each iteration.

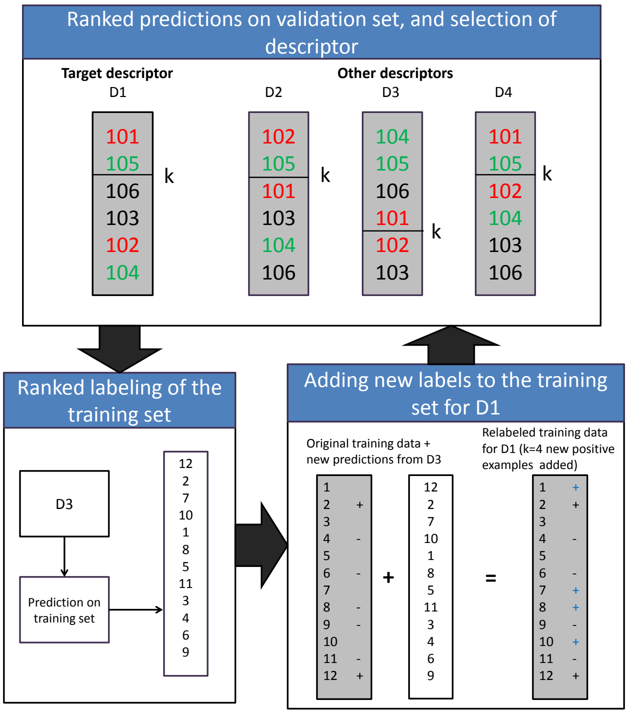
Ranked predictions on validation set, and selection of descriptor
Target descriptor
Other descriptors
D1
D2
D3
D4
101 105 106 103 102 104
102 105 101 103 104 106
104 105 106 101 102 103
101 105 102 104 103 106
k
Ranked labeling of the training set
D3
Prediction on training set
12 2 7 10 1 8 5 11 3 4 6 9
Adding new labels to the training set for D1
Original training data + new predictions from D3
Relabeled training data for D1 (k=4 new positive examples added)
1
2 +
3
4 -
5
6 -
7
8 -
9 -
10
11 -
12 +
+
=
1 +
2 +
3
4 -
5
6 -
7 +
8 +
9 -
10 +
11 -
12 +

Figure 1: Selecting source descriptor for Selective Multi Cotraining methods.

3.2 Selection Methods

We present two methods to select the source descriptor which is used to add annotations to the target descriptor. We add k new positive annotations to the target descriptor's training set using the selected descriptor. For the first method k is fixed while for the other one k is found automatically per descriptor. The two methods are detailed in the next two subsections. In what follows we have used the terms descriptor and classifier interchangeably where the intention is always a learner using (or for) the descriptor.

3.2.1 Positive Disagreement

Let's suppose we want to add k annotations to the examples for descriptor $D1$ and we have an initial ranking of the validation set for all the descriptors. The upper part of figure 1 contains validation ranked lists for all the descriptors, V_D. The numbers represent the ids of the shots and the colors indicate the actual labels where green, red and black mean positive, negative and unlabeled respectively. Each shot S_i in the list has a rank $r^{V_D}(S_i)$which is an integer value. We first define a function f returning 1 when the shot S_i is annotated positive.

$$f(S_i) = \begin{cases} 1 & sign(S_i) = + \\ 0 & otherwise \end{cases} \tag{1}$$

To select the most complementary descriptor we look at the first k shots of the target descriptor and the first k shots of the other descriptors in a pairwise manner. Since we have the labels of the shots on the validation set we can calculate the disagreement between a pair for a fixed value of k, the source descriptor D_s and the target D_t as:

$$DIS_s = \sum_{i:r^{V_{D_s}}(S_i)<k \ \wedge \ r^{V_{D_t}}(S_i)>k} f(S_i) \qquad (2)$$

The disagreement counts for the source descriptor D_s how many labels are positive that did not appear in the first k shots for the target descriptor D_t. This is essentially the information that is missed by the target descriptor for the top k shots. We select the D_s with maximum DIS_s as it is understandable that the predictions done by this descriptor are the most complementary to D_t. For figure 1 if k is fixed as 4, then $DIS_2 = 0$, $DIS_3 = 1$ and $DIS_4 = 1$.

This descriptor selection is done separately for each concept. That is to say that for each concept for D_t the best complementary descriptor is chosen among the pool of available descriptors. The value of k here is fixed and we set it to the percentage of initial positive examples for every concept. We try 3 values of this percentage as $\{10\%, 20\%, 30\%\}$. Once D_s is found we use it to re-label the training set and then add k new positive labels to the training set for D_t as shown in the bottom part of figure 1.

3.2.2 *Precision based Rank*

The positive disagreement approach suffers from two setbacks. First is how to select the optimal value of k. For concepts with low initial positive examples 10% new labels are quite low while on the other hand for concepts with already abundant pool of positive examples 10% new examples may add noise. This noise will increase with k. The other predicament which is related to first one is how sure are we that the new labels added really bring valuable information. Furthermore and as more iterations are performed how useful the newly added labels are before they start to add wrong annotations and distort the data.

To cater these issues we need to find a source D_s for D_t that is expected to add up to a certain percentage of correct positive labels. We use the precision on the classification results of the validation set for each D_s to build this criterion. Figure 1 is again used to explain this method. First for every ranked list we find k for a fixed precision value pr. That is to say using the true labels of the validation set we scroll down the ranked list calculating precision at every step and stop when the precision is lower than pr. The precision for the list V_D for a value of k is

$$P^k_{V_D} = \frac{\sum_{i=1}^{k} f(S_i)}{k} \qquad (3)$$

and to find k for V_D we maximize equation 3 for k as:

$$\begin{aligned} k_s &= \underset{k}{\operatorname{argmax}} \, P^k_{V_{D_s}} \\ s.t. \quad & P^k_{V_{D_s}} >= pr \end{aligned} \qquad (4)$$

So for the example in the figure 1 we find the values of k_s for precision of 50. $k_3 = 4$ and for the other three descriptors the value of k is 2. Note that k_1 is not an important factor here as the k for target descriptor is not used in the criterion presented just after.

Once k is determined for each source descriptor, for D_t we simply select the D_s that maximizes the average rank of the first k_s shots on the validation set V_{D_t}. Maximum rank means that those shots that are ranked as positive by the source classifier are at the bottom of the list V_{D_t}. It means that this source classifier identifies the most serious missclassifications for the target on the validation set. We define this average rank for the source descriptor D_s as:

$$PRR_s = \frac{\sum_{i=1}^{k_s} r^{V_{D_t}}(S_i)}{k_s} \qquad (5)$$

where each source has a different PRR_s for a unique k_s and the D_s which maximizes equation 5 is selected for D_t. So to summarize we use the positive labels to determine the value of k_s but after we only used the average rank of shots ranked by D_t in the first k_s shots of V_{D_s}.

After determining D_s we use it to re-label the training set and add k_s new examples to the training set of D_t. Using k_s which had a precision greater than pr, it is expected that up to $pr\%$ correct labels are added to the training label set for D_t. As shown in the figure 1 D_3 is selected with $k_3 = 4$ and then 4 new labels are added to the training set of D_1 with 3 unlabeled examples being labeled as positive and 1 negative's label flipped. Again as the positive disagreement method this selection is done separately for each concept. For further iterations of the process the modified training and validation lists are used from the previous iteration.

4. RESULTS AND EXPERIMENTS

4.1 Experimental Setup

We have carried out experiments on the TRECVID 2013 [7] dataset where the development set consists of about 800 hours of internet videos of lengths varying divided into training and validation parts. The test part contains 600 hours of slightly longer videos. Training is done on a list of 60 concepts out of which NIST has evaluated 38 for the 2013 Semantic INdexing (SIN) task.

We have mainly extracted 2 kinds of descriptors from the video keyframes the SIFT [5] and the color SIFT [10] which are all densely extracted. These extracted descriptors are then used to build visual dictionaries using k-means clustering. Using the above extractions we build 5 types of descriptors of varying lengths (dictionary sizes). For dense SIFT, dictionaries of 4000 and 10,000 are built and we get two descriptors: dsift4K and dsift10K. For color dense SIFT we have cdsift1K, cdsift4K and cdsift10K from dictionaries of 1000, 4000 and 10,000 visual words.

All the classifiers used are 1 vs. all SVM classifiers using homogeneous kernel maps [11] built on the input features. We have used Pegasos training [8] for speedy optimizations. We calculate the Average Precision (AP) for each concept and present the percentage Mean AP (MAP).

4.2 Results

We have conducted a certain number of experiments using the two proposed descriptor selection methods. For these experiments we have used all the labeled data provided by NIST to train the initial classifiers. All the new labels are added either to the unlabeled examples or in some cases labels of negative examples are flipped. The results are detailed in the next few subsections.

4.2.1 *Cotraining vs. 1 pass and selftraining*

The two selection methods for multi-cotraining are compared with the single pass learning (Baseline) and also with selftraining or bootstrapping. Two kinds of selftrainings were done; adding fixed percentage of positive examples

Descriptor	Baseline	Selftraining				Positive Disagreement				MaxAvg Rank			
		pr	1	2	3	k	1	2	3	**pr**	1	2	3
cdsift1K	8.11	50	7.88	7.84	7.01	10	8.04	8.05	7.33	50	8.23	8.21	7.98
		60	7.58	7.46	6.94	20	8.31	7.60	6.92	60	8.11	8.19	7.75
		70	8.41	7.85	7.74	30	7.67	6.54	5.60	70	7.85	**8.65**	8.32
cdsift4K	8.12	50	8.19	8.15	7.56	10	8.60	**8.72**	8.34	50	8.44	8.51	8.09
		60	8.45	8.13	8.14	20	8.58	8.58	7.75	60	8.33	8.33	8.17
		70	8.42	8.56	8.21	30	8.54	7.79	6.75	70	8.58	**8.61**	8.40
cdsift10K	8.18	50	8.05	7.88	7.25	10	8.54	8.49	**9.19**	50	**8.81**	**8.80**	8.53
		60	8.15	7.92	7.50	20	8.37	8.37	8.28	60	8.59	**9.02**	8.69
		70	8.28	7.88	7.69	30	8.50	8.10	7.13	70	8.60	**8.68**	**8.99**
dsift4K	7.52	50	7.63	7.29	7.09	10	7.68	7.91	8.16	50	7.68	7.39	7.49
		60	7.77	7.72	7.60	20	7.80	7.90	7.24	60	7.82	7.75	7.42
		70	7.87	7.76	7.50	30	7.79	7.59	6.32	70	7.59	7.77	7.40
dsift10K	7.84	50	8.15	7.98	7.44	10	8.31	8.27	**8.40**	50	8.15	8.28	8.03
		60	8.11	7.67	7.68	20	8.19	8.32	8.07	60	8.11	8.02	7.72
		70	7.94	7.86	7.98	30	7.99	7.71	6.67	70	8.02	7.96	7.75

Table 1: Mean Average Precision for various methods for 38 evaluated concepts

and adding positive annotations using the precision method, where we first calculate k_t for expected precision value using equation 3 and then add k_t new positive labels to the training set of D_t. Table 1 shows results for second kind of selftraining as it perform better among the two and compares it to the two proposed selective cotraining methods.

We show results for 3 iterations of relabeling and retraining in table 1 for all the semi-supervised methods. Results that are significantly better with randomization testing [9] are highlighted in bold. Selftraining and positive disagreement (DIS) methods are mostly outperformed by the Maximum rank on precision based selection (PRR). This is true for further iterations as for DIS noise is added with a fixed value of k, and selftraining lacks complementarity of using other descriptors. When k is 10% the performance of DIS is good as few noisy labels are added and it sometimes shows better results than the PRR. For the PRR criteria as the precision increases the value of k_s decreases and in many cases no new labels are added for certain categories for example for $pr = 70\%$. Though we see an improvement for most of the descriptors for DIS and PRR the color SIFT descriptors seem to absorb more noise than others.

4.2.2 Cotraining vs. Linear Fusion

To check the complementarity of descriptors we fused every possible pair of the 5 descriptors and compare the performance with the fusion after first iteration of each semi-supervised learning method. We have used weighted linear fusion to merge the classification scores with the weights optimized on the validation set. Results are compared in the figure 2 where PRR dominates and the best result of the fusion of baseline descriptors (10% MAP) is outperformed by a MAP of 10.43% for PRR.

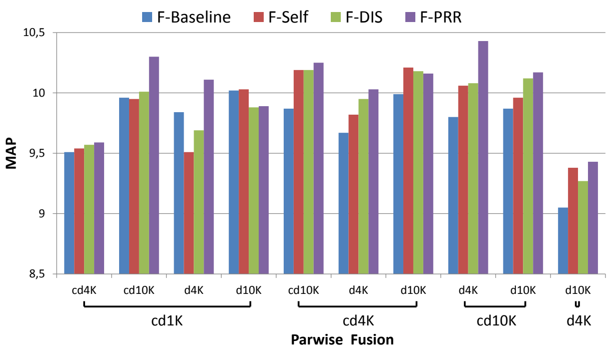

Figure 2: Linear fusion of every pair of descriptor for different methods

5. CONCLUSIONS

We have demonstrated the effectiveness of selecting the appropriate feature to relabel a target learner for cotraining when different options are available. For each descriptor the best complementary descriptor is selected and the number of examples to label is also automatically selected which are expected to be correct up to a certain percentage. For future work it is required to find a successful weighing strategy to find relevance of the newly added labels. The relevance scores for a category can be attached to each example and will highlight its importance in training [4]. Furthermore it is important to (i) refine the k new positive examples added by an automatic or manual [12] selection, and (ii) identify the number of cotraining iterations for each concept. Eventually the selective cotraining can be used to find negatives by reversing the selection criteria.

6. REFERENCES

[1] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT*, 1998.

[2] W. Du, R. Phlypo, and T. Adali. Adaptive feature split selection for co-training: Application to tire irregular wear classification. In *ICASSP*, 2013.

[3] G.-Z. Li, D. Li, W.-C. Lu, J. Y. Yang, and M. Q. Yang. Feature selection for co-training: A qsar study. In *IC-AI*, 2007.

[4] X. Li and C. G. M. Snoek. Classifying tag relevance with relevant positive and negative examples. In *ACM International Conference on Multimedia*, 2013.

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

[6] K. Nigam and R. Ghani. Analyzing the effectiveness and applicability of co-training. In *CIKM*, 2000.

[7] P. Over, G. Awad, M. Michel, J. Fiscus, G. Sanders, W. Kraaij, A. F. Smeaton, and G. Quéenot. Trecvid 2013 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proceedings of TRECVID 2013*. NIST, USA, 2013.

[8] S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal estimated sub-gradient solver for svm. ICML, 2007.

[9] M. D. Smucker, J. Allan, and B. Carterette. A comparison of statistical significance tests for information retrieval evaluation. In *ACM-IKM*, 2007.

[10] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. Empowering visual categorization with the gpu. *IEEE Transactions on Multimedia*, 13(1), 2011.

[11] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms, 2008.

[12] R. Yan and M. R. Naphade. Co-training non-robust classifiers for video semantic concept detection. In *ICIP*. IEEE, 2005.

[13] R. Yan and M. R. Naphade. Semi-supervised cross feature learning for semantic concept detection in videos. In *CVPR*. IEEE, 2005.