Automatic Feature Decomposition for Single View Co-training

Minmin Chen, Kilian Q. Weinberger, Yixin Chen MC15, KILIAN, CHEN@CSE.WUSTL.EDU
Washington University in Saint Louis, 1 Brookings Dr., Saint Louis, MO 63130 USA

Abstract

One of the most successful semi-supervised learning approaches is co-training for multi-view data. In co-training, one trains two classifiers, one for each view, and uses the most confident predictions of the unlabeled data for the two classifiers to "teach each other". In this paper, we extend co-training to learning scenarios without an explicit multi-view representation. Inspired by a theoretical analysis of Balcan et al. (2004), we introduce a novel algorithm that splits the feature space during learning, explicitly to encourage co-training to be successful. We demonstrate the efficacy of our proposed method in a weakly-supervised setting on the challenging Caltech-256 object recognition task, where we improve significantly over previous results by (Bergamo & Torresani, 2010) in almost all training-set size settings.

1. Introduction

Co-training (Blum & Mitchell, 1998) is an approach to semi-supervised learning (Zhu, 2006) which assumes that the available data is represented with two views. In its original formulation, these two views must satisfy two conditions: 1. each one is sufficient to train a low-error classifier and 2. both are class-conditionally independent. A classifier is trained for each representation and applied to the unlabeled data. Co-training then utilizes unlabeled data by adding the most confident predictions of each classifier to the training set of the other classifier – effectively letting the classifiers "teach each other". Blum and Mitchell show drastic improvements on data sets where the multi-view assumptions naturally hold. Co-training and its variants have been applied to many applications across computer science and beyond (Collins & Singer, 1999; Ghani, 2001; Nigam & Ghani, 2000; Levin et al., 2003; Brefeld & Scheffer, 2004; Chan et al., 2004).

In many learning scenarios, the available data might not originate from two explicitly different sources. Instead, one might be faced with assorted features that were obtained through various means. For example, in the medical domain features might correspond to different examinations which might or might not be class-conditionally independent and sufficiently informative about the patient's condition.

In this paper we extend co-training to this more common single-view setting. We utilize recent advances in learning theory that have significantly weakened the strong assumptions of co-training. Most notably, Balcan et al. (2004) prove that the class-conditional independence assumption is unnecessarily strong and that a weaker *expanding* property on the underlying distribution of the multi-view data is sufficient for iterative co-training to succeed. We propose a novel feature decomposition algorithm, which automatically divides the features of a single-view data set into two mutually exclusive subsets – thereby creating a pseudo-multi-view representation for co-training. This feature division is learned explicitly to satisfy the necessary conditions to enable successful co-training. In this paper we derive a single optimization problem, which divides the feature space, trains both classifiers and enforces an approximation of Balcan's ϵ-expanding property through hard constraints. We refer to our algorithm as *Pseudo Multi-view Co-training* (PMC).

Our broadening of the scope of co-training is particularly useful for weakly supervised learning scenarios. Through the success of web-search, it is now possible to obtain large quantities of data for almost any topic or class description (*e.g.* through automated image search or wikipedia lookups). Often, however, only a small fraction of the retrieved search results are truly relevant to someone's learning task. As co-training explicitly cherry-picks data instances with similar characteristics as the labeled training data, it is naturally suited for learning with such noisy (weak) labels. We demonstrate this capability by effectively utiliz-

Appearing in *Proceedings of the 28th International Conference on Machine Learning*, Bellevue, WA, USA, 2011.

ing weakly labeled image-search results to improve the classification accuracy on the Caltech 256 object recognition data set – surpassing previously published results on the same task by Bergamo & Torresani (2010).

2. Notation and Setting

Let $\mathcal{X} \subseteq \mathcal{R}^d$ be the instance space with dimension d and $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\} \subset \mathcal{X}$. Assume w.l.o.g. that the first $n \ll m$ instances are accompanied by corresponding labels $\{y_1, \ldots, y_n\} \in \mathcal{Y}$, where labels and instances are drawn from some joint distribution $\mathcal{D}$. The labels of the remaining instances are unknown. For convenience, we denote the set of labeled instances by L and the unlabeled ones by U. For now, until section 4, we focus on binary problems and set $\mathcal{Y} = \{+1, -1\}$.

2.1. Co-Training

Co-training assumes that the data set X consists of two views $\mathcal{X} = \mathcal{X}^1 \times \mathcal{X}^2$ with their respective feature partitions X^1, X^2. The two views must satisfy two conditions: 1. Both have to be *sufficient* within the given hypothesis class $\mathcal{H}$ – *i.e.* there exist two hypothesis $h^1, h^2 \in \mathcal{H}$ having low error on X^1, X^2 respectively. 2. They need to be *class-conditionally independent*, i.e. for a given $\mathbf{x} = (\mathbf{x}^1, \mathbf{x}^2) \in \mathcal{X}^1 \times \mathcal{X}^2$ with label $y \in \mathcal{Y}$, $p(\mathbf{x}^1|y)p(\mathbf{x}^2|y) = p(\mathbf{x}^1, \mathbf{x}^2|y)$.

The fundamental idea behind co-training is what Blum and Mitchell describe as "rote-learning" (Blum & Mitchell, 1998) on the unlabeled data set. Two classifiers h^1, h^2 are trained on the labeled set L, both on their respective views. The two classifiers are then evaluated on the unlabeled set U. For each classifier, the examples on which it is most confident are removed from U and added to L for the next iteration. Both classifiers are now re-trained on the expanded labeled data set and the procedure is repeated until some stopping criteria is met. By carrying out this "rote learning" algorithm, co-training can bootstrap from a small labeled "seed" set and iteratively improve its performance with the help of unlabeled data.

2.2. ϵ-Expandability

The assumption that the two views are class conditionally independent is very strong and, as Nigam & Ghani (2000) show, can easily be violated in practice. Recent work by Balcan et al. (2004) weakens this requirement significantly. Intuitively, for the two classifiers to be able to teach each other, they must make confident predictions on different subsets of the unlabeled data. Balcan et al. (2004) formalize this condition as a concept of *ϵ-expandability*.

Let h^1, h^2 be the two classifiers, trained on the two views. Let us denote the subsets of X^1, X^2 on which these two classifiers are confident as C^1, C^2 respectively. For $S \subseteq X$, let $\mathbf{S}^i$ denote the event that an input $\mathbf{x} = (\mathbf{x}^1, \mathbf{x}^2) \in S$ satisfies $\mathbf{x}^i \in C^i$. We express the probability of an instance in S to be classified confidently by both classifiers as $\Pr(\mathbf{S}^1 \wedge \mathbf{S}^2)$, by *exactly one* of the two classifiers as $\Pr(\mathbf{S}^1 \oplus \mathbf{S}^2)$ and by none as $\Pr(\overline{\mathbf{S}^1} \wedge \overline{\mathbf{S}^2})$.

Definition 1. *$\mathcal{D}$ is ϵ-**expanding** with respect to the hypothesis class $\mathcal{H}$ if for any $S \subseteq X$ and any two classifiers $h^1, h^2 \in \mathcal{H}$, the following statement holds*

$$\Pr(\mathbf{S^1} \oplus \mathbf{S^2}) \geq \epsilon \min[\Pr(\mathbf{S^1} \wedge \mathbf{S^2}), \Pr(\overline{\mathbf{S^1}} \wedge \overline{\mathbf{S^2}})].$$

Intuitively, the condition ensures that with high probability there are data instances in the unlabeled set for which *exactly one* of the two classifiers is confident. These instances can then be added to the labeled set to teach the classifier which wasn't so sure about them. Balcan et al. (2004) show that if the distribution $\mathcal{D}$ is **ϵ-expanding**, and the two classifiers are never "confident but wrong", co-training will succeed.

3. Method

In this section, we extend co-training to the scenario where the two views $\mathcal{X}^1, \mathcal{X}^2$ are not known. We describe how to learn two classifiers on a single view $\mathcal{X}$ that satisfy three conditions: (1) both of them perform well on the labeled data; (2) both are trained on strictly different features; (3) together they are likely to satisfy Balcan's condition of ϵ-expandability. We tackle all three conditions in this order.

3.1. Loss function

In this paper we only consider linear classifiers, $h_{\mathbf{u}}(\mathbf{x}) = \text{sign}(\mathbf{u}^\top \mathbf{x} + b)$ with weight vector $\mathbf{u}$. To simplify notation we drop the bias b and assume that a constant 1 is attached as an additional dimension to each input $\mathbf{x}_i \in X$, which is *not split* between the two classifiers. A classifier $h_{\mathbf{u}}$ is trained by minimizing the log-loss over the data set L:

$$\ell(\mathbf{u}; L) \quad = \quad \sum_{(\mathbf{x},y)\in L} \log\left(1 + e^{-\mathbf{u}^T \mathbf{x} y}\right). \qquad (1)$$

Our framework is agnostic to the specific choice of loss-function, however we choose logistic regression (Ng & Jordan, 2002) as it explicitly models the probability of labels y conditioned on the input $\mathbf{x}$, which provides a natural measure of classifier-confidence.

For co-training we require two classifiers, whose weight vectors we denote by $\mathbf{u}$ and $\mathbf{v}$. We train these two

jointly, and to make sure that *both* suffer low loss we minimize the maximum of the two,

$$\min_{\mathbf{u},\mathbf{v}} \quad \max\left[\ell(\mathbf{u};L),\ell(\mathbf{v};L)\right]. \tag{2}$$

As eq. (2) is non-differentiable we introduce a slight relaxation and replace the max term with a more manageable softmax. The optimization then becomes

$$\min_{\mathbf{u},\mathbf{v}} \quad \log\left(e^{\ell(\mathbf{u};L)}+e^{\ell(\mathbf{v};L)}\right). \tag{3}$$

3.2. Feature Decomposition

A crucial aspect of co-training is that the two classifiers are trained on different views of the data set. Unconstrained, the minimization problem in (3) would result in two identical weight vectors $\mathbf{u}=\mathbf{v}$. Instead, we want the two classifiers to divide up the feature space so that each feature can only be used by one of the two. More precisely, for each feature i, at least one of the two classifiers must have a zero weight in the i^{th} dimension. We can write this constraint as

$$\forall i, 1\le i\le d, \quad \mathbf{u}_i\mathbf{v}_i=0. \tag{4}$$

Although correct, this formulation is unnecessarily hard to optimize and can result in numerical instabilities. Instead, we square both sides and sum over all features to obtain the following constraint:

$$\sum_{i=1}^{d}\mathbf{u}_i^2\mathbf{v}_i^2=0. \tag{5}$$

It is important to point out that any solution to (5) strictly implies (4).

3.3. ϵ-Expandability

The final condition is that the two classifiers must make confident predictions on different subsets of the unlabeled data. We follow the intuition behind the ϵ-expandability of Balcan et al. (2004), as described in section 2.2. For the classifier $h_{\mathbf{u}}$, let $\hat{y}=\text{sign}(\mathbf{u}^\top\mathbf{x})\in\{\pm1\}$ denote the class prediction and $p(\hat{y}|\mathbf{x};\mathbf{u})=(1+e^{-\mathbf{u}^\top\mathbf{x}\hat{y}})^{-1}$ its confidence. We define a binary confidence indicator function[1] as

$$c_{\mathbf{u}}(\mathbf{x})=\begin{cases}1 & \text{if } p(\hat{y}|\mathbf{x};\mathbf{u})>\tau\\ 0 & \text{otherwise.}\end{cases} \tag{6}$$

Let us define the subsets of the inputs on which one classifier is confident as $C_{\mathbf{u}}=\{\mathbf{x}\in X \mid c_{\mathbf{u}}(\mathbf{x})=1\}$, and $C_{\mathbf{v}}$ respectively. For any $S\subseteq X$, let $\mathbf{S_u}$ denote the event that an input $\mathbf{x}$ in S belongs to the confident set of $h_{\mathbf{u}}$, that is, $\mathbf{x}\in S\cap C_{\mathbf{u}}$. $\mathbf{S_v}$ is defined analogously.

The ϵ-expanding condition from section 2.2 becomes

$$\Pr(\mathbf{S_u}\oplus\mathbf{S_v})\ge\epsilon\min[\Pr(\mathbf{S_u}\wedge\mathbf{S_v}),\Pr(\overline{\mathbf{S_u}}\wedge\overline{\mathbf{S_v}})]. \tag{7}$$

As pointed out by Balcan et al. (2004), the definition of ϵ-expanding might still be unnecessarily strict in practice. For our optimization, we relax it and only require that the expanding condition holds on average for the solution $h_{\mathbf{u}},h_{\mathbf{v}}\in\mathcal{H}$. More explicitly we add the following hard constraint to our optimization:

$$\sum_{\mathbf{x}\in U}\left[c_{\mathbf{u}}(\mathbf{x})\bar{c}_{\mathbf{v}}(\mathbf{x})+\bar{c}_{\mathbf{u}}(\mathbf{x})c_{\mathbf{v}}(\mathbf{x})\right]$$
$$\ge\epsilon\min\left[\sum_{\mathbf{x}\in U}c_{\mathbf{u}}(\mathbf{x})c_{\mathbf{v}}(\mathbf{x}),\sum_{\mathbf{x}\in U}\bar{c}_{\mathbf{u}}(\mathbf{x})\bar{c}_{\mathbf{v}}(\mathbf{x})\right] \tag{8}$$

Here, $\bar{c}_{\mathbf{u}}(\mathbf{x})=1-c_{\mathbf{u}}(\mathbf{x})$ indicates that classifier $h_{\mathbf{u}}$ is *not* confident about input $\mathbf{x}$. Intuitively, the constraint in eq. (8) ensures that the total number of inputs in U that can be used for rote-learning because *exactly one* classifier is confident (LHS), is larger than the set of inputs which can *not* because *both* classifiers are already confident or *both are not* confident (RHS).

3.4. Optimization Problem

In summary, we want to learn two logistic regression classifiers, both with small loss on the labeled data set, while satisfying two constraints to ensure feature decomposition and ϵ-expandability. We combine eqs (3-8) as the following optimization problem, which we will later refer to as *Pseudo Multi-view Decomposition* (PMD) :

$$\min_{\mathbf{u},\mathbf{v}} \ \log\left(e^{\ell(\mathbf{u};L)}+e^{\ell(\mathbf{v};L)}\right)$$
subject to:
$$\textbf{(1)}\ \sum_{i=1}^{d}\mathbf{u}_i^2\mathbf{v}_i^2=0$$
$$\textbf{(2)}\ \sum_{\mathbf{x}\in U}\left[c_{\mathbf{u}}(\mathbf{x})\bar{c}_{\mathbf{v}}(\mathbf{x})+\bar{c}_{\mathbf{u}}(\mathbf{x})c_{\mathbf{v}}(\mathbf{x})\right] \ge \epsilon\min\left[\sum_{\mathbf{x}\in U}c_{\mathbf{u}}(\mathbf{x})c_{\mathbf{v}}(\mathbf{x}),\sum_{\mathbf{x}\in U}\bar{c}_{\mathbf{u}}(\mathbf{x})\bar{c}_{\mathbf{v}}(\mathbf{x})\right]$$

We optimize this constrained optimization problem with an augmented Lagrangian method (Bertsekas et al., 1999).

3.5. Pseudo Multi-View Co-Training

Finally, we use our feature decomposition method to apply iterative co-training on single-view data. We refer to the resulting algorithm as *Pseudo Multi-view*

[1] In our implementation, the 0-1 indicator was replaced by a very steep differentiable sigmoid function, and τ was set to 0.8 across different experiments.

Co-training (PMC). A detailed pseudo-code implementation is presented in Algorithm 1. Please note that there is an interesting difference between PMC and traditional co-training. Not only is there no predefined split of the features but the automatically found split can vary between iterations.

Algorithm 1 PMC in pseudo-code.

```
Inputs: L and U.
Initialize u, v and l.
repeat
    Find u*, v* by optimizing PMD on L and U.
    Apply h_{u*} and h_{v*} on all elements of U.
    Move up-to l confident inputs from U to L.
until No more predictions are confident
Train final classifier h on L with all features X.
Return h
```

4. Extension to Multiclass Settings

One way to extend PMC to multiclass settings is by training multiple binary classifiers, one for each class, using the one-versus-the-rest scheme. However, such an approach cannot capture the correlations between different classes. A more natural and efficient way is to construct a hypothesis, considering all the classes at once.

Let us denote the label space as $\mathcal{Y} = \{1, 2, \cdots, K\}$. Let $\mathbf{U} = [\mathbf{u}^1, \ldots, \mathbf{u}^K] \in \mathcal{R}^{d\times K}$ and $\mathbf{V} = [\mathbf{v}^1, \ldots, \mathbf{v}^K] \in \mathcal{R}^{d\times K}$ denote the parameters of the two classifiers. Then the log-loss of a classifier $h_{\mathbf{U}}$ over the data set L is defined as:

$$\ell(\mathbf{U}; L) = -\sum_{(\mathbf{x},y)\in L} \log \frac{e^{\mathbf{x}^\top \mathbf{u}^y}}{\sum_k e^{\mathbf{x}^\top \mathbf{u}^k}}. \tag{9}$$

The confidence indictor function $c_{\mathbf{U}}(\mathbf{x}), c_{\mathbf{V}}(\mathbf{x})$ are defined as in (6) with the class prediction computed as

$$\hat{y} = \max_k (\mathbf{x}^\top \mathbf{u}^k). \tag{10}$$

We can also decompose the instance space by constraining that eq. (5) holds for all classes, *i.e.*

$$\sum_{k=1}^{K}\sum_{i=1}^{d} (\mathbf{u}_i^k)^2 (\mathbf{v}_i^k)^2 = 0. \tag{11}$$

However, without additional regularization, eq. (11) would result in K different decompositions, one for each class. For the classifiers to be compatible, we need to ensure a consistent partition of the instance space $\mathcal{X} = \mathcal{X}^1 \times \mathcal{X}^2$ across different classes. More precisely, for each feature i, at least one of the two classifiers must have zero weights *across all classes*.

A similar problem arises in the context of feature selection in multi-task settings (Argyriou et al.; Obozinski et al., 2006), when similar parameter sparsity patterns across different tasks needs to be imposed. An effective regularization is the group lasso, defined over a matrix $\mathbf{U}$ as

$$\|\mathbf{U}\|_{2,1} = \sum_{i=1}^{d} \sqrt{\sum_{k=1}^{K} (\mathbf{U}_{ik})^2}. \tag{12}$$

Intuitively, eq. (12) enforces the l_1 norm on the l_2-norms of all rows in $\mathbf{U}$, enforcing sparsity on a per-row level – effectively forcing $h_{\mathbf{U}}$ to pick a feature for all classes or none. We encourage readers to refer to (Obozinski et al., 2006) for an intuitive geometric interpretation of the regularization in (12).

We then combine the other two constraints with the regularized objective into the following optimization problem:

$$\min_{\mathbf{U},\mathbf{V}} \ \log\left(e^{\ell(\mathbf{U};L)} + e^{\ell(\mathbf{V};L)}\right) + \lambda(\|\mathbf{U}\|_{2,1} + \|\mathbf{V}\|_{2,1})$$

subject to:

$$\textbf{(1)} \ \sum_{k=1}^{K}\sum_{i=1}^{d} (\mathbf{u}_i^k)^2 (\mathbf{v}_i^k)^2 = 0$$

$$\textbf{(2)} \ \sum_{\mathbf{x}\in U} [c_{\mathbf{U}}(\mathbf{x})\bar{c}_{\mathbf{V}}(\mathbf{x}) + \bar{c}_{\mathbf{U}}(\mathbf{x})c_{\mathbf{V}}(\mathbf{x})] \geq \epsilon \min\left[\sum_{\mathbf{x}\in U} c_{\mathbf{U}}(\mathbf{x})c_{\mathbf{V}}(\mathbf{x}), \sum_{\mathbf{x}\in U} \bar{c}_{\mathbf{U}}(\mathbf{x})\bar{c}_{\mathbf{V}}(\mathbf{x})\right]$$

5. Results

In this section we evaluate PMC empirically on artificial and real-world data sets.

5.1. Paired Handwritten Digits

As a first test, we construct a data set with binary class labels for which a class-conditional feature split exists, but is unknown to the algorithm. Each instance in the set is a pair of digits sampled from the USPS handwritten digits set. If the class label is +1, the left image is uniformly picked from the set of *ones* and *twos* and the right image is picked from the set of *fives* or *sixes*. For a label −1 the left digit is a *three* or *four* and the right image a *seven* or *eight*. Given the class-label, the identities of the two digits in the image are conditionally independent. We construct $m = 6000$ such instances. By design, a natural decomposition is to split the feature space into two views such that one covers the left digit and the other the right digit.

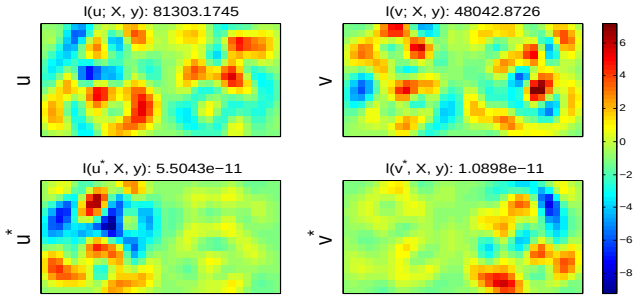

Figure 1. Upper row: The heatmap of two randomly initialized weight vectors $\mathbf{u}, \mathbf{v}$ on the input space; Lower row: The heatmap of $\mathbf{u}^*, \mathbf{v}^*$ learned with PMD.

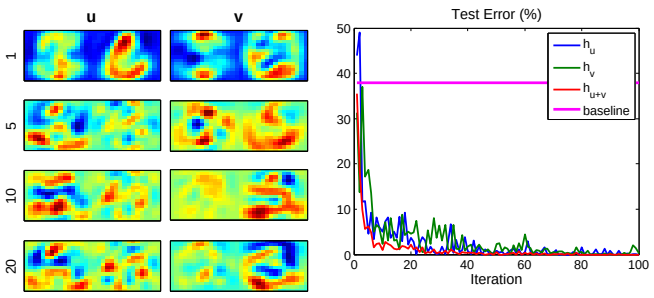

Figure 2. Left: The heatmap of $\mathbf{u}, \mathbf{v}$ in 20 PMC iterations (1^{st} iteration on top). *Right:* The progress made by the two classifiers $h_{\mathbf{u}}, h_{\mathbf{v}}$ during co-training.

Feature Decomposition. First we set $|L| = 2000$ and solve the PMD for $\mathbf{u}$ and $\mathbf{v}$, starting with a random initialization. Figure 1 shows the heat maps of $\mathbf{u}$ and $\mathbf{v}$ before and after training. We also report the log-loss on L. The bottom two images in the figure show that once $\mathbf{u}, \mathbf{v}$ are trained to minimize the loss function, constraining on (5) and (8), their non-zero weights are divided almost exactly into the two class-conditionally independent feature sets. In particular, classifier $h_{\mathbf{u}}$ takes the pixels of the left digit as its features, while classifier $h_{\mathbf{v}}$ uses only the right digit.

Co-Training. As a second evaluation, we set $|L| = 2$ and run 12 sets of identical experiments with different random initializations and different labeled images (always one per class). In this setup we use the transductive setting, i.e. the test set coincides with the unlabeled set.

Table 1. Comparison of co-training with automatic feature split (PMC) to (1) baseline model with only labeled instances; (2) co-training with random feature split (RFS); (3) co-training with ICA and then random feature splitting (ICA-RFS), on the paired handwritten digits set.

Test Err(%)	Baseline	RFS	ICA-RFS	PMC
Mean	18.64	13.78	**12.22**	**3.99**
STD	8.86	14.24	**13.59**	**3.24**

Table 1 summarizes the mean classification error and standard deviation. We compare against three alternative methods: i) the *baseline*, which trains logistic regression exclusively with the labeled instances; ii) co-training on two views obtained by random feature splitting *(RFS)*; iii) co-training with random feature splitting where the features are pre-processed with Independent Component Analysis[2] (Hyvárinen et al.) (*ICA-RFS*). For RFS and ICA-RFS, 10 different random feature decompositions are considered for each run, and the average performance and the standard deviation across 120 runs are reported. As shown in Table 1, PMC achieves by far the lowest error with very small standard deviation.

The left plot in Figure 2 shows the heat maps of the two weight vectors $\mathbf{u}$ and $\mathbf{v}$ in different PMC iterations. Confident predictions are moved from the unlabeled set U to the labeled set L in each PMC iteration, causing the loss function (3) and the constraint (8) to change. As a result, the automatically discovered feature splits vary between iterations. As more confident predictions were added to L, PMC gradually approximates the class-conditional feature split from Figure 1.

The right plot depicts the progress of the two classifiers $h_{\mathbf{u}}, h_{\mathbf{v}}$ in one run of the experiments. The magenta line indicates the test error of the baseline. The blue and green curves plot the errors of the two classifiers between iterations. During the "rote-learning" procedure, the two classifiers "learn" from each other, and finally converge to a almost perfect predictor. Similar trends were observed in the other 11 runs.

5.2. Caltech-256 with weakly labeled web images

As a more challenging real-world data classification task, we evaluate PMC on the Caltech-256 object categorization data (Griffin et al., 2007). The data set consists of images of objects from a set of 256 object categories. The task is to classify a new image into its object category. A great amount of human effort is required to label such data. To address this problem, several researchers suggested a *weakly-supervised* learning setting (Fergus et al., 2005; Vijayanarasimhan & Grauman, 2008; Bergamo & Torresani, 2010), in which additional images are retrieved from image search engines such as Google™ or Bing™ image search using the category names as

[2] We used the open-source implementation from `http://cs.nyu.edu/~roweis/kica.html`.

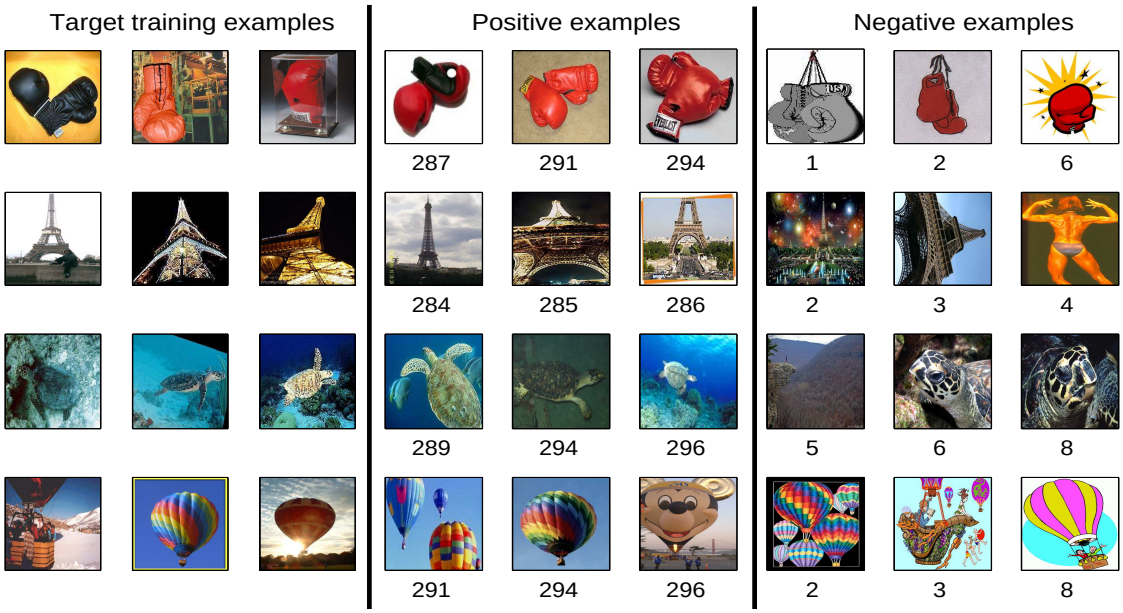

Figure 3. Refining image search ranking with Multi-class PMC. The left three columns show the original training images from Caltech-256; the middle three columns show the images having lowest rank in Bing™search, but were picked by PMC as confident examples; the right three columns show the images with highest ranks, but found to be irrelevant by PMC. The numbers below images are the rankings of the corresponding image search result with Bing™image search. The experiment was run with 5 training images from Caltech-256, and 300 weakly-labeled web images for each class.

search queries, and used to aid learning. These additional images are referred to as *weakly-labeled*, because the quality of the retrieved images is far from the original training data. Usually, a large fraction of the retrieved images do not contain the correct object. With this method, Bergamo & Torresani (2010) report an improvement of 65% (27.1% compared to 16.7%) over the previously best published result on the set with 5 labeled training examples per class.

Though retrieving images from web search engines requires very little human intervention, only a small fraction of the retrieved images actually correspond to the queried category. Further, even the relevant images are of varying quality compared with typical images from the training set. Bergamo & Torresani (2010) overcome this problem by carefully down-weighing the web images and employing adequate regularization to suppress the noises introduced by irrelevant and low-quality images. As features, they use classemes (Lorenzo et al., 2010), where each image is represented by a 2625 dimensional vector of predictions from various visual concept classifiers – including predictions on topics as diverse as "wetlands", "ballistic missile" or "zoo"[3].

[3]A detailed list of the categories is available at http://www.cs.dartmouth.edu/~lorenzo/projects/classemes/classeme_keywords.txt.

In this experiment, we apply PMC to the same dataset from Bergamo & Torresani (2010), using images from Caltech-256 as labeled data, and images retrieved from Bing as "unlabeled" data. Different from classical semi-supervised learning settings, in this case, we are not fully blind about the labels of the unlabeled data. Instead, for each class only the images obtained with the matching search query are used as the "unlabeled" set.

We argue that PMC is particularly well suited for this task for two reasons: i) The "rote-learning" procedure of co-training adds confident instances iteratively. As a result, images that possess similar characteristics as the original training images will be picked as the confident instances, naturally ruling out irrelevant and low-quality images in the unlabeled set. ii) Classemes features are a natural fit for PMC as they consist of the predictions of many (2625) different visual concepts. It is highly likely that there exists two mutually exclusive subsets of visual concepts that satisfy the conditions for co-training.

Figure 3 shows example images of the Caltech-256 training set (left column), positive examples that PMC picks out from the "unlabeled" set to use as additional labeled images (middle) and negative examples which PMC chooses to ignore (right column). The number below the images indicates its rank of the Bing™search engine (out of 300). For this figure, we

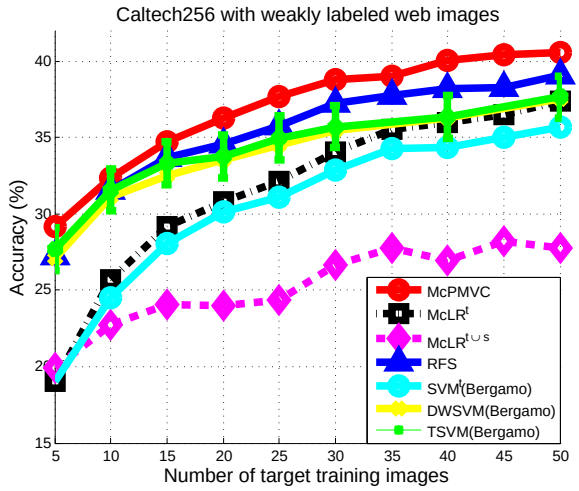

Figure 4. Recognition accuracy with 300 web images and a varying number of Caltech256 training images m.

selected the highest ranked negative and lowest ranked positive images. The figure showcases how PMC effectively identifies relevant images that are similar in style to the training set for rote-learning. Also, it showcases that PMC can potentially be used for image re-ranking of search engines, which is particularly visible in the middle row where it ignores completely irrelevant images to the category "Eiffel Tower", which are ranked second to fourth on Bing™.

Baselines. Figure 4 provides a quantitive analysis of the performance of PMC. The graph shows the accuracy achieved by different algorithms under a varying number of training examples m and 300 weakly-labeled Bing™image-search results. The meta-parameters of all algorithms were set by 5-fold cross-validation on the small labeled set (except for the group-lasso trade-off for PMC, which was set to $\lambda = .1$).

We train our algorithm with the multi-class loss and compare it against three baselines and three previously published results in the literature. The three baselines are: i) multi-class logistic regression trained only on the original labeled training examples from Caltech-256 (LR^t); ii) the same model trained with both the original training images and web images ($LR^{t\cup s}$); iii) co-training with random feature splits on the labeled and weakly-labeled data (RFS).

The three previously published algorithms are: i) linear support vector machines trained on the labeled Caltech-256 images (SVM^t) only; ii) the algorithm proposed by Bergamo & Torresani (2010), which weighs the loss over the weakly labeled data less than over the original data ($DWSVM$); iii) transductive-SVM as introduced by Joachims (1999) ($TSVM$). All previously published results are taken from (Bergamo & Torresani, 2010). All algorithms, including PMC, are linear and make no particular assumptions on the data.

General Trends. As a first observation, $LR^{t\cup s}$ performs drastically worse than the baseline trained on the Caltech-256 data LR^t only. This indicates that the weakly-labeled images are noisy enough to be harmful when they are not filtered or down-weighted. However, if the weakly labeled images are incorporated with specialized algorithms, the performance improves as can be seen by the clear gap between the purely supervised (SVM^t and LR^t) and the adaptive semi-supervised algorithms. The result of co-training with random splitting (RSF) is surprisingly good, which could potentially be attributed to the highly diverse classemes features. Finally PMC outperforms all other algorithms by a visible margin across all training set sizes. PMC achieved an accuracy of 29.2% when only 5 training images per class from Caltech-256 are used, comparing to 27.1% as reported in (Bergamo & Torresani, 2010). In terms of computational time, for a total of around 80,000 labeled and unlabeled images, PMC took around 12 hours to finish the entire training phase (Testing time is in the order of milliseconds).

6. Related Work

Applicability of co-training has been largely depending on the existence of two class-conditionally independent views of the data (Blum & Mitchell, 1998). Nigram and Ghani (Nigam & Ghani, 2000) perform extensive empirical study on co-training and show that the class-conditionally independence assumption can be easily violated in real-world data sets. For datasets without natural feature split, they create artificial split by randomly breaking the feature set into two subsets. Chan et al. (2004) also investigate the feasibility of random feature splitting and apply co-training to email-spam classification. However, during our study we found that random feature splitting results in very fluctuant performance. Brefeld & Scheffer (2004) effectively extend the multi-view co-training framework to support vector machines.

Abney (2002) relaxes the class conditionally independent assumption to weak rule dependence and proposed a greedy agreement algorithm that iteratively adds unit rules that agree on unlabeled data to build two views for co-training. In contrast, PMC is not greedy but incorporates an optimization problem over all possible feature splits. Zhang & Zheng (2009) propose to decompose the feature space by first applying PCA and then greedily dividing the orthogonal components to minimize the energy diversity of the two

feature sets. In contrast, our method is supervised and non-greedy.

7. Conclusion

In this paper, we introduced PMC, a framework for co-training on single-view data. PMC automatically decomposes the feature space and creates a pseudo-multi-view representation explicitly designed for co-training to succeed. It involves a single optimization problem, which jointly divides the feature space, trains two classifiers, and enforces an approximation of Balcan's $\epsilon-$expanding property. We further extended PMC to multi-class settings and demonstrated PMC's efficacy on the Caltech256 object recognition task using weakly labeled web images.

The ability of PMC to effectively select high quality instances from large collections of weakly labeled search results opens the door to future work on diverse sets of web-specific applications across very diverse domains – including web-spam classification, sentiment analysis or information retrieval.

Acknowledgments

The authors thank Yahoo Research for their generous support that enabled this research.

References

Abney, S. Bootstrapping. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pp. 360–367, 2002.

Argyriou, A., Evgeniou, T., and Pontil, M. Multi-task feature learning. *Advances in neural information processing systems*, 19:41.

Balcan, M.F., Blum, A., and Yang, K. Co-training and expansion: Towards bridging theory and practice. *Advances in neural information processing systems*, 17:89–96, 2004.

Bergamo, A. and Torresani, L. Exploiting weakly-labeled web images to improve object classification: a domain adaptation approach. In *Neural Information Processing Systems (NIPS)*, 2010.

Bertsekas, D.P., Hager, W.W., and Mangasarian, O.L. *Nonlinear programming*. Athena Scientific Belmont, MA, 1999.

Blum, A. and Mitchell, T. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pp. 100. ACM, 1998.

Brefeld, U. and Scheffer, T. Co-EM support vector learning. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 16. ACM, 2004.

Chan, J., Koprinska, I., and Poon, J. Co-training with a single natural feature set applied to email classification. In *Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence*, pp. 586–589, 2004.

Collins, M. and Singer, Y. Unsupervised models for named entity classification. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pp. 189–196, 1999.

Fergus, R., Fei-Fei, L., Perona, P., and Zisserman, A. Learning Object Categories from Google's Image Search. *Computer Vision, Tenth IEEE International Conference on*, 2, 2005.

Ghani, R. Combining labeled and unlabeled data for text classification with a large number of categories. In *Proceedings of the IEEE International Conference on Data Mining*, volume 2, 2001.

Griffin, G., Holub, A., and Perona, P. Caltech-256 object category dataset. 2007.

Hyvárinen, A., Hurri, J., and Hoyer, P.O. Independent component analysis. *Natural Image Statistics*, pp. 151–175.

Joachims, T. Transductive inference for text classification using support vector machines. In *Machine Learning International Workshop*, pp. 200–209. Citeseer, 1999.

Levin, A., Viola, P., and Freund, Y. Unsupervised improvement of visual detectors using co-training. In *Proc. ICCV*, volume 2, pp. 626–633. Citeseer, 2003.

Lorenzo, T., Szummer, M., and Fitzgibbon, A. Efficient object category recognition using classemes. In *European Conference on Computer Vision (ECCV)*, pp. 776–789, September 2010.

Ng, A.Y. and Jordan, M.I. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in neural information processing systems*, 2:841–848, 2002.

Nigam, K. and Ghani, R. Analyzing the effectiveness and applicability of co-training. In *Proceedings of the ninth international conference on Information and knowledge management*, pp. 86–93. ACM, 2000.

Obozinski, G., Taskar, B., and Jordan, M. Multi-task feature selection. In *the workshop of structural Knowledge Transfer for Machine Learning in the 23rd International Conference on Machine Learning (ICML 2006)*. Citeseer, 2006.

Vijayanarasimhan, S. and Grauman, K. Keywords to visual categories: Multiple-instance learning forweakly supervised object categorization. 2008.

Zhang, W. and Zheng, Q. TSFS: A Novel Algorithm for Single View Co-training. In *Computational Sciences and Optimization, 2009. CSO 2009. International Joint Conference on*, volume 1, pp. 492–496, 2009.

Zhu, X. Semi-supervised learning literature survey. *Computer Science, University of Wisconsin-Madison*, 2006.