Cross-Database Transfer Learning via Learnable and Discriminant Error-correcting Output Codes

Feng-Ju Chang[1], Yen-Yu Lin[1], and Ming-Fang Weng[2]

[1]Research Center for Information Technology Innovation, Academia Sinica, Taiwan
[2]Institute of Information Science, Academia Sinica, Taiwan
{fengju, yylin}@citi.sinica.edu.tw, mfueng@iis.sinica.edu.tw

Abstract. We present a transfer learning approach that transfers knowledge across two multi-class, unconstrained domains (source and target), and accomplishes object recognition with few training samples in the target domain. Unlike most of previous work, we make no assumption about the relatedness of these two domains. Namely, data of the two domains can be from different databases and of distinct categories. To overcome the domain variations, we propose to learn a set of commonly-shared and discriminant attributes in form of *error-correcting output codes*. Upon each of attributes, the unrelated, multi-class recognition tasks of the two domains are transformed into correlative, binary-class ones. The extra source knowledge can alleviate the high risk of overfitting caused by the lack of training data in the target domain. Our approach is evaluated on several benchmark datasets, and leads to about 40% relative improvement in accuracy when only one training sample is available.

1 Introduction

Object recognition is one of the most fundamental problems in vision research. Despite its importance, most of existing recognition techniques are still hindered by two main challenges: the large intra-class variation and the large number of categories to be identified. Recent research efforts, e.g., [2, 3], on *multiple kernel learning* [4, 5] overcome these difficulties via employing various powerful descriptors. However, these approaches rely on a large set of training samples to investigate data variations, such that the optimal kernel combination can be determined. In general, the cost of data labeling is quite expensive.

To reduce the labeling effort, *transfer learning* [6] has been demonstrated to be a promising technique for object recognition with few training samples. It delivers useful knowledge in the *source* to improve the *target* model learning. However, most transfer learning algorithms [7–9] handle merely the knowledge delivery *from either multiple classes or a single class in the source to a single class in the target*, and assume high correlation between the source and target domains. These two restrictions might reduce their applicability to object recognition, where diverse classes and multi-class classification are involved.

The proposed approach transfers knowledge *from muliple classes to multiple classes*, given two multi-class recognition tasks (one in the source domain and

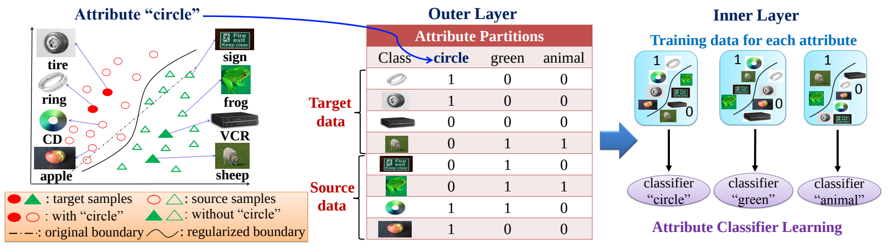


Fig. 1. The proposed two-layer boosting framework for transfer learning. While outer layer discovers attribute partitions, inner layer deals with classifier learning.

the other in the target domain). We leverage the extra source knowledge to learn a more robust multi-class classifier rather than a set of binary classifiers in the target domain. However, no assumptions about the relatedness between the two domains - data of the two domains can be from either the same or different databases, and their respective object categories may partially overlap or don't overlap at all - makes direct knowledge sharing infeasible.

Therefore, we propose to transfer knowledge through a sequence of the automatically learned *attributes*. Figure. 1 illustrates our idea. By associating with attribute `circle`, the *unrelated, multi-class* recognition tasks of the two domains become *correlative, binary-class* (i.e., with or without circle) ones. It follows that the extra source knowledge can be transferred to regularize the target classifier learning, which alleviates *overfitting* caused by the scarceness of target training data, and leads to a better classifier with the low generalization error.

To carry out our idea, we present a *two-layer (the outer and inner layers) boosting framework* that automatically derives a sequence of *discriminant* attributes, *commonly shared* by both the source and target domains, to facilitate knowledge transfer. Specifically, the outer layer implements a multi-task variant of *AdaBoost.OC* [10], which sequentially discovers a set of *attribute partitions* in form of *error-correcting output codes*. Each attribute divides all classes into two subsets - presence (1) or absence (0) of the attribute. With each attribute, the training data for learning the *attribute classifier* are of binary labels.

The inner layer exploits the training data from the outer layer to learn attribute classifiers by the principle of classifier sharing, which has been demonstrated to be effective in [11, 12] for learning related tasks via sharing the weak learners in boosting. Namely, after associating the source and target data with each attribute partition, the learning of attribute classifiers in both two domains are cast into a joint optimizaiton problem.

Based on the two-layer boosting framework, our approach has the following three contributions. First, with the outer layer, the attribute partitions are complementary to each other due to the iterative minimization of the *pseudoloss* [10]. Second, with the inner layer, the robust and discriminant attribute classifiers can be learned in the target domain. Third, our formulation not only carries out knowledge transfer, but also supports multiple kernel learning for feature fusion, since attributes are high-level concepts and each of them might be captured by a distinct combination of low-level features.

The proposed method is evaluated on three benchmark datasets. Both the performances of *within-database transfer* and *cross-database transfer* demonstrate the effectiveness of our approach, when only few training samples are available. Section 2 shows several related work. The AdaBoost.OC is described in Section 3, and the details of our approach is presented in Section 4. Section 5 and 6 respectively show the experiments and conclusions.

2 Related Work

The task of object recognition is typically formulated as a multi-class classification problem. Much research effort has been devoted to the design of more discriminant features, e.g., [13, 14], so as to minimize the intra-class variation while maximize the inter-class discrepancy simultaneously. On the other hand, many variants of multiple kernel learning algorithms, e.g., [1–3], are proposed to integrate various features in a unified fashion, and lead to higher recognition accuracy. However, most of these approaches require training on a sufficient number of data to build stable classifiers.

The early investigation into learning with few examples to recognize objects is pioneered by Fei-Fei et al. [8]. They suggest to extract visual knowledge from a set of previously learned object classes to represent a novel one. Intermediate features [15, 16] such as attributes bridging low-level image features and high-level concepts have been shown potential for establishing a discriminant model with few, even zero new datum. In general, these attributes require tedious and expensive human efforts on labeling. Recently, Qi et al. [17] introduce a cross-category transfer learning algorithm, and explicitly transfer knowledge via label correlation between categories. Although improvement is demonstrated, their method depends on a large number of human-driven multi-label data.

The exploration of auxiliary data drawn from a different distribution for the target data has received a rapidly growing interest for visual object categorization [18, 19]. The methods exploiting knowledge of either labeled or unlabeled data can be generally divided into several categories [6]: transfer by *model parameters* [20], by *data instances* [21], and by *feature representation* [22]. However, most of these approaches handle only transfer learning *from multiple classes to a single class*, in which knowledge from multiple sources is transferred to improve a single object category in the target. Thus it results in sub-optimal effectiveness on solving multi-class recognition problems.

Another notable category of work is to leverage auxiliary data to build a number of classifiers, and new feature representations of target data are yielded by applying these classifiers to themselves, e.g., *classemes* [23]. Along this line of thoughts, knowledge embedded within source domain can be revealed and then transferred to target domain for further use [24, 25]. However, exploring good and proper features among such a rich set to address the target task still relies on a large number of training data in practice. In contrast, our method transfers knowledge by common attribute discovery instead of feature augmentation, and hence allows to account for such an ill-posed situation.

Algorithm 1: *AdaBoost.OC*

```
Input: Dataset $D = \{(\mathbf{x}_n \in \mathcal{X}, y_n \in \mathcal{Y})\}_{n=1}^N$ of $C$ classes, iteration $U$.
Output: Classifier $F(\mathbf{x}) = \arg\max_{c\in\mathcal{Y}} \sum_{t=1}^U \alpha_t [\![ f_t(\mathbf{x}) = B_t(c) ]\!]$.
Initialize: $\tilde{w}_{n,c} = [\![ y_n \neq c ]\!]/(N(C-1))$, for $1 \le n \le N$ and $1 \le c \le C$
for $t \leftarrow 1, 2, \ldots, U$ do
    1. Learn a class partition function $B_t : \mathcal{Y} \to \{-1, 1\}$.
    2. Compute data weights $\{w_n\}_{n=1}^N$ by
        $w_n = \frac{\sum_{c=1}^C \tilde{w}_{n,c} [\![ B_t(y_n) \neq B_t(c) ]\!]}{\sum_{j=1}^N \sum_{c=1}^C \tilde{w}_{j,c} [\![ B_t(y_j) \neq B_t(c) ]\!]}$.
    3. Select weak learner $f_t : \mathcal{X} \to \{-1, 1\}$ by
        $f_t = \arg\min_f \sum_{n=1}^N w_n [\![ f(\mathbf{x}_n) \neq B_t(y_n) ]\!]$.
    4. Compute coefficient $\alpha_t = \frac{1}{2} \ln \frac{1-\epsilon}{\epsilon}$, where
        $\epsilon = \frac{1}{2} \sum_{n=1}^N \sum_{c\in\mathcal{Y}} \tilde{w}_{n,c}([\![ B_t(y_n) \neq f_t(\mathbf{x}_n) ]\!] + [\![ B_t(c) = f_t(\mathbf{x}_n) ]\!])$.
    5. Update and normalize data weights $\{\tilde{w}_{n,c}\}_{n=1,c=1}^{N,C}$ by
        $\tilde{w}_{n,c} = \tilde{w}_{n,c} \cdot \exp(\alpha_t([\![ B_t(y_n) \neq f_t(\mathbf{x}_n) ]\!] + [\![ B_t(c) = f_t(\mathbf{x}_n) ]\!]))/Z$.
```

3 AdaBoost.OC for Multi-class Classification

We introduce AdaBoost.OC [10] in this section, since some of its key elements are used in the establishment of the proposed approach. AdaBoost.OC is a multi-class boosting algorithm, which integrates *error-correcting output codes (ECOC)* [26] into the procedure of boosting, and solves a multi-class classification problem by reducing it to a set of binary ones. With dataset $D = \{(\mathbf{x}_n \in \mathcal{X}, y_n \in \mathcal{Y})\}_{n=1}^N$ of C classes where $\mathcal{Y} = \{1, ..., C\}$, the steps of AdaBoost.OC are given in Algorithm 1.

AdaBoost.OC links the successive boosting iterations by keeping an array of weights $\{\tilde{w}_{n,c}\}_{c=1}^C$ for each $\mathbf{x}_n$. Except for $\{\tilde{w}_{n,y_n}\}_{n=1}^N$, weights $\{\tilde{w}_{n,c}\}_{n=1,c=1}^{N,C}$ are uniformly distributed in initialization. At iteration t, an *output code or partition function* B_t is first built to divide the set of class labels into two subsets, i.e., $B_t : \mathcal{Y} \to \{-1, 1\}$. As suggested in [10], the optimal B_t is obtained by maximizing

$$J(B_t) = \sum_{c,c'\in\mathcal{Y}} [\![B_t(c) \neq B_t(c')]\!] w(c, c'), \text{ where } w(c, c') = \sum_{n=1}^N \tilde{w}_{n,c} [\![c' = y_n]\!] \quad (1)$$

and $[\![\cdot]\!]$ denotes the indicator function. As $w(c, c')$ measures the *difficulty* of separating data of class c and c', maximizing $J(B_t)$ in Eq.(1) tends to assign two classes that currently mix together to opposite subsets. It follows that more emphases on separating these difficult classes will be given at this iteration.

Since partition function B_t converts the multi-class learning problem to the binary one, the data weights, weak hypothesis, and its ensemble coefficient are respectively computed in step 2 $\sim$ 4 of Algorithm 1. Iteration t is then completed by updating the data weights. Refer to [10] for the details of AdaBoost.OC.

4 Multi-class Transfer Learning

We carry out multi-class transfer learning by generalizing AdaBoost.OC to jointly deal with two multi-class learning tasks (one from source and the other from target). At each iteration, we discover a discriminant attribute that is commonly shared by the two domains via redesigning the partition function. By this attribute, the two learning problems become two *related, binary* (with or without this attribute) ones. It follows that knowledge transfer from source to target can be accomplished via the concept of classifier sharing. We summarize the proposed approach in Algorithm 2. In the following, we first give the problem definition. Then the two key components of our approach, including (i) *sharing the partition functions* for domain correlation and (ii) *jointly learning domain-dependent classifiers* for knowledge transfer, are respectively described.

4.1 Problem Definition

Given a *source domain* dataset $D_S = \{(\mathbf{x}_n^S \in \mathcal{X}, y_n^S \in \mathcal{Y}_S = \{1, ..., C_S\})\}_{n=1}^{N_S}$ of C_S classes and a *target domain* dataset $D_T = \{(\mathbf{x}_n^T \in \mathcal{X}, y_n^T \in \mathcal{Y}_T = \{C_S + 1, ..., C_S + C_T\})\}_{n=1}^{N_T}$ of C_T classes, our goal is to learn a classifier that recognize target data with low generalization error. This goal is achieved by considering not only information extracted from D_T but also knowledge transferred from D_S. In the setting, the data space $\mathcal{X}$ in the source and target domains is the same. No assumption about the correlation between D_S and D_T is required.

To deal with complex visual recognition tasks, we consider using multiple descriptors to more precisely characterize the data, i.e., $\mathbf{x}_n = \{\mathbf{x}_{n,m} \in \mathcal{X}_m\}_{m=1}^M$. To handle the unfavorable diversity of representation among $\{\mathcal{X}_m\}$, we represent data under each descriptor by a *kernel matrix*. It allows that these features can be combined in the domain of kernel matrices. For each descriptor m, the corresponding kernel matrix K_m and kernel function k_m can be constructed by

$$K_m(n, n') = k_m(\mathbf{x}_n, \mathbf{x}_{n'}) = \exp\left(-\gamma_m d_m^2(\mathbf{x}_{n,m}, \mathbf{x}_{n',m})\right), \tag{2}$$

where d_m is the associated distance function, and γ_m is a positive constant. We empirically set $1/\gamma_m$ as the average of the pairwise distances. The resulting kernels $\{K_m\}_{m=1}^M$ will serve as the information bottleneck for data access.

4.2 On Sharing Partition Functions for Domain Correlation

At this stage, we aim to correlate two multi-class learning tasks to facilitate knowledge transfer. As mentioned before, this is fulfilled by deriving a class partition function at each iteration of AdaBoost.OC. Two criteria are considered: 1) As in AdaBoost.OC, the partition function should be discriminant for both domains; 2) As in transfer learning, a pair of similar classes (in the view of their data distributions) in the opposite domains should be partitioned into the same side. We propose an effective method to learn the discriminant, commonly-shared partition function, detailed in the following three steps.

Algorithm 2: *Multi-class Transfer Learning*

```
Input: source data $\{(\mathbf{x}_n^S \in \mathcal{X}, y_n^S \in \mathcal{Y}_S)\}_{n=1}^{N_S}$ of $C_S$ classes,
       target data $\{(\mathbf{x}_n^T \in \mathcal{X}, y_n^T \in \mathcal{Y}_T)\}_{n=1}^{N_T}$ of $C_T$ classes, iterations $U$
Output: target domain classifier $F(\mathbf{x}) = \arg\max_{c \in \mathcal{Y}_T} \sum_{t=1}^{U} \alpha_t^T [\![ f_t^T(\mathbf{x}) = B_t(c) ]\!]$
Initialize: $\tilde{w}_{n,c}^S = [\![ y_n^S \neq c ]\!] / (N_S(C_S - 1))$, for $1 \leq n \leq N_S$ and $c \in \mathcal{Y}_S$
            $\tilde{w}_{n,c}^T = [\![ y_n^T \neq c ]\!] / (N_T(C_T - 1))$, for $1 \leq n \leq N_T$ and $c \in \mathcal{Y}_T$.
for $t \leftarrow 1, 2, \ldots, U$ do
    1. Construct the class partition function $B_t$ via (6).
    2. Compute data weights $\{w_n^\ell\}_{n=1,\ell \in \{S,T\}}^{N_\ell}$ by
           $w_n^\ell = \frac{\sum_{c \in \mathcal{Y}_\ell} \tilde{w}_{n,c}^\ell [\![ B_t(y_n^\ell) \neq B_t(c) ]\!]}{\sum_{j=1}^{N_\ell} \sum_{c \in \mathcal{Y}_\ell} \tilde{w}_{j,c}^\ell [\![ B_t(y_j^\ell) \neq B_t(c) ]\!]}$.
    3. Get a pair of weak classifiers $f_t^S$ and $f_t^T$ via Algorithm 3.
    4. Compute coefficient $\alpha_t^\ell = \frac{1}{2} \ln \frac{1 - \epsilon_\ell}{\epsilon_\ell}$, for $\ell \in \{S, T\}$, where
           $\epsilon_\ell = \frac{1}{2} \sum_{n=1}^{N_\ell} \sum_{c \in \mathcal{Y}_\ell} \tilde{w}_{n,c}^\ell ([\![ B_t(y_n^\ell) \neq f_t^\ell(\mathbf{x}_n^\ell) ]\!] + [\![ B_t(c) = f_t^\ell(\mathbf{x}_n^\ell) ]\!])$.
    5. Update and normalize data weights $\{\tilde{w}_{n,c}^\ell\}_{n=1,c \in \mathcal{Y}_\ell,\ell \in \{S,T\}}^{N_\ell}$ by
           $\tilde{w}_{n,c}^\ell = \tilde{w}_{n,c}^\ell \exp(\alpha_t^\ell ([\![ B_t(y_n^\ell) \neq f_t^\ell(\mathbf{x}_n^\ell) ]\!] + [\![ B_t(c) = f_t^\ell(\mathbf{x}_n^\ell) ]\!])) / Z_\ell$.
```

1. Construct *Intra-domain Graph* G: The vertices of G are over data classes of both domains. Its edge weights are defined in $W \in \mathbb{R}^{(C_S+C_T)\times(C_S+C_T)}$ by

$$W(c,c') = \begin{cases} w^S(c,c'), & \text{if } c \in \mathcal{Y}_S \wedge c' \in \mathcal{Y}_S, \\ w^T(c,c'), & \text{if } c \in \mathcal{Y}_T \wedge c' \in \mathcal{Y}_T, \\ 0, & \text{otherwise,} \end{cases} \tag{3}$$

where $w^S(c,c')$ is the same as $w(c,c')$ in Eq.(1) except only source data are considered. Similarly, $w^T(c,c')$ is computed for target data.

2. Construct *Inter-domain Graph* G': Similar to G, the edges of graph G' are recorded in $W' \in \mathbb{R}^{(C_S+C_T)\times(C_S+C_T)}$, defined by

$$W'(c,c') = \begin{cases} s(c,c'), & \text{if } c \in \mathcal{Y}_T \wedge c' \in \mathcal{Y}_S, \\ s(c',c), & \text{if } c' \in \mathcal{Y}_T \wedge c \in \mathcal{Y}_S, \\ 0, & \text{otherwise,} \end{cases} \tag{4}$$

where $s : \mathcal{Y}_T \times \mathcal{Y}_S \rightarrow \mathbb{R}$ measures the similarity between two classes of opposite domains. To account for the imbalance in data numbers, s is defined by

$$s(c,c') = \sum_{\{\mathbf{x}_n^T | y_n^T = c\}} w_n^T \max_{\{\mathbf{x}_{n'}^S | y_{n'}^S = c'\}} k(\mathbf{x}_n^T, \mathbf{x}_{n'}^S), \tag{5}$$

where w_n^T is the weight of $\mathbf{x}_n^T$ in AdaBoost.OC and k is the kernel function. Note that s is *asymmetrically* defined for classes in the source and target domains. By

Eq.(5), it measures how well source class c' *covers* target class c. For multiple kernels, we simply adopt the average of the inter-domain graphs of all kernels.

3. Establish Partition Function B: The shared partition function $B : \mathcal{Y}_S \cup \mathcal{Y}_T \to \{-1, 1\}$ is built as follows

$$B(c) = \begin{cases} +1, & \text{if } \mathbf{b}^*(c) \geq \theta, \\ -1, & \text{otherwise,} \end{cases} \tag{6}$$

where θ is a threshold and $\mathbf{b}^*(c)$ is the cth element of $\mathbf{b}^* \in \mathbb{R}^{C_S + C_T}$, which is the generalized eigenvector corresponding to the largest eigenvalue of

$$L\mathbf{b} = \lambda L'\mathbf{b}. \tag{7}$$

$L = \text{diag}(W \cdot \mathbf{1}) - W$ and $L' = \text{diag}(W' \cdot \mathbf{1}) - W'$ in Eq.(7) are the *graph Laplacian* of G and G' respectively.

To gain insight into the learned partition function B in Eq.(6), we consider the following optimization problem

$$B^* = \arg\max_B J_{dis}(B)/J_{shr}(B), \text{ where} \tag{8}$$

$$J_{dis}(B) = \sum_{c,c' \in \mathcal{Y}_S} [\![B(c) \neq B(c')]\!] w^S(c,c') + \sum_{c,c' \in \mathcal{Y}_T} [\![B(c) \neq B(c')]\!] w^T(c,c'), \tag{9}$$

$$J_{shr}(B) = \sum_{c \in \mathcal{Y}_T} \sum_{c' \in \mathcal{Y}_S} [\![B(c) \neq B(c')]\!] s(c,c'). \tag{10}$$

To optimize Eq.(8), maximizing $J_{dis}(B)$ and minimizing $J_{shr}(B)$ are required simultaneously. While the former makes the learned B *dis*criminant for both domains (cf. Eq.(1) and Eq.(9)), the latter connects the two domains by partitioning similar classes of opposite domains into the same side. Thus the yielded B is commonly *shared* by two domains. Maximizing $J_{dis}(B)$ itself is a NP complete problem [10]. We instead adopt the continuous relaxation to solve Eq.(8). That is, an auxiliary variable $\mathbf{b} \in \mathbb{R}^{(C_S + C_T)}$ is adding, and terms $[\![B(c) \neq B(c')]\!]$ in Eq.(9) and Eq.(10) are replaced with $||\mathbf{b}(c) - \mathbf{b}(c')||^2$. It follows that the resulting optimization problem can be optimally solved via Eq.(7). The proof is omitted here for the sake of space. After $\mathbf{b}^*$ in Eq.(7) is obtained, the value of threshold θ in Eq.(6) can be determined by maximizing the objective function Eq.(8).

4.3 On Jointly Learning Classifiers for Knowledge Transfer

Once the partition function B is solved, datasets D_S and D_T at the current iteration of boosting become *weighted*, *binary*, and *related*, i.e., $\{w_n^S, \mathbf{x}_n^S, B(y_n^S)\}_{n=1}^{N_S}$ and $\{w_n^T, \mathbf{x}_n^T, B(y_n^T)\}_{n=1}^{N_T}$. To learn an accurate target domain classifier, we use the principle of classifier sharing [11, 12]. This way, not only information obtained in the target domain but also knowledge borrowed from the source domain can be considered. Apart from knowledge transfer, two additional requirements arise at this stage: 1) The learning process can deal with weighted data; 2) Multiple kernels should be considered jointly. We fulfill these requirements by introducing a boosting-based approach, whose two main elements are given below.

Algorithm 3: *Dual-domain Boosting*

```
Input: weighted source data $\{(w_n^S, \mathbf{x}_n^S, y_n^S \in \pm 1)\}_{n=1}^{N_S}$,
       weighted target data $\{(w_n^T, \mathbf{x}_n^T, y_n^T \in \pm 1)\}_{n=1}^{N_T}$, iteration $V$.
Output: source classifier $f^S$ and target classifier $f^T$, where
       $f^\ell(\mathbf{x}) = \text{sign} \sum_{t=1}^{V} \beta_t^\ell h_t(\mathbf{x})$ for $\ell \in \{S, T\}$.
for $t \leftarrow 1, 2, \ldots, V$ do
    1. Select the optimal dyadic hypercut $h_t$ by
       $h_t = \arg\min_h \sum_{\ell \in \{S,T\}} \sum_{n=1}^{N_\ell} w_n^\ell [\![ h_t(\mathbf{x}_n^\ell) \neq y_n^\ell ]\!]$.
    2. Compute coefficient $\beta_t^\ell = \max(0, \frac{1}{2} \ln \frac{1-\epsilon_\ell}{\epsilon_\ell})$, for $\ell \in \{S, T\}$ where
       $\epsilon_\ell = \sum_{n=1}^{N_\ell} w_n^\ell [\![ h_t(\mathbf{x}_n^\ell) \neq y_n^\ell ]\!]$.
    3. Update and normalize data weights $\{w_n^\ell\}_{n=1, \ell \in \{S,T\}}^{N_\ell}$ by
       $w_n^\ell = w_n^\ell \exp(-y_n^\ell \beta_t^\ell h_t(\mathbf{x}_n^\ell)) / Z_\ell$.
```

The Design of Weak Learners To learn a boosted classifier with multiple kernels, we adopt the method proposed in [12]. The discriminant power of each kernel is first converted into a set of weak learners, called *dyadic hypercuts* [27]. It turns out that multiple kernel learning is achieved by boosting over the pool of dyadic hypercuts yielded from all the kernels.

A dyadic hypercut h is composed of three elements: a positive sample $\mathbf{x}_p$ (i.e., $y_p = 1$), a negative sample $\mathbf{x}_n$ (i.e., $y_n = -1$), and a kernel function k_m. Note that $\mathbf{x}_p$ and $\mathbf{x}_n$ can be from the source or target domains. The yielded model is

$$h(\mathbf{x}) = \text{sign}(k_m(\mathbf{x}_p, \mathbf{x}) - k_m(\mathbf{x}_n, \mathbf{x}) - b), \tag{11}$$

where b is a threshold, whose value is determined by error minimization in boosting. The size of the weak learner pool is $|\mathcal{H}| = N^+ \times N^- \times M$, where N^+ (N^-) is the number of positive (negative) data, and M is the number of kernels.

Dual-Domain Boosting for Knowledge Transfer To transfer knowledge from the source domain to the target domain, we follow the strategy of classifier sharing in [12] where the relatedness between tasks are modeled by the shared weak learners while the difference between tasks are reflected by their respective ensemble coefficients. That is, the binary classifiers f^S and f^T in the two domains are respectively given by

$$f^\ell(\mathbf{x}) = \sum_{t=1}^{V} \beta_t^\ell h_t(\mathbf{x}), \text{ for } \ell \in \{S, T\}, \tag{12}$$

where $\{h_t\}$ are the shared weak learners and $\{\beta_t^\ell\}$ are the respective coefficients.

Algorithm 3 provides a systematic way of learning f^S and f^T simultaneously. It can be proved that the dyadic hypercut h_t picked in the step 1 will directly minimize the total *exponential loss* of the two domains, while the the ensemble coefficients defined in step 2 are respectively determined to minimize the exponential loss of the corresponding domains. Hence our approach comes with theoretical support of AdaBoost. It turns out that the high risk of overfitting resulting from insufficient samples in the target domain could considerably alleviated, since knowledge from the correlated source domain can smoothly regularize and benefit the process of learning f^T via joint weak learner selection. Further, the intrinsic differences between the two domains can be properly addressed through their respective ensemble coefficients whose values are determined by considering data in individual domains. It hence relieves the unfavorable effect of *negative transfer*, which typically occurs if the underlying differences between the source and target domains are not properly modeled in knowledge transfer.

5 Experimental Results

The proposed method is evaluated in this section. We describe in turn the adopted datasets, features and kernels, baseline approaches, and the within-database and cross-database transfer learning settings along with their results.

5.1 Datasets

To evaluate the performance of the proposed method, we conduct experiments on three publicly available datasets, Caltech256 [28], SUN09 [29], and MSRC [30]. Although each of these datasets has its own emphasis [31], they are popular benchmarks of object recognition due to their broad coverage of object characteristics and divergent appearances of objects within a single category.

5.2 Features and Kernels

Generally speaking, there is no universal feature that can be effectively used to recognize diverse object categories. Hence, we select four representative features to capture various characteristics of images, including:

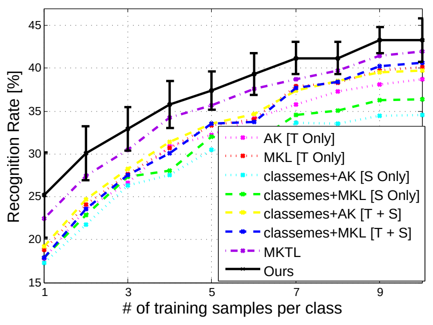
GIST: We apply the *gist* descriptor [14] to the resized images with an 128×128 pixel prior. Each dimension of the resulted vectors is normalized to have a standard normal distribution, i.e., zero-mean and unit-variance.

BoW-SIFT: We uniformly sample interesting points in an image and describe them by the SIFT descriptor [13]. With a dictionary of $1,000$ visual words, each image is represented as a histogram using this dictionary.

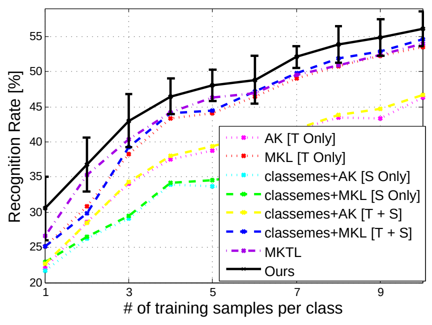
Color Histogram: We use a 166-bin color histogram extracted from the HSV color space to represent an image, where the hue, saturation, and intensity channels are divided into 18, 3, and 3 bins, respectively, and there are four additional scales of intensity for describing gray images.

Texton: We consider 99 filters from three filter banks to generate the vocabularies of texture prototypes [32]. Hence, an image can be represented by a histogram that records its probability distribution over all the generated textons.

For each feature, we construct its RBF kernel with the Euclidean distance.



(a) SUN09



(b) Caltech256

Fig. 2. Within-database transfer learning in (a) SUN09 and (b) Caltech256 datasets. Recognition rates of all the baselines and our approach are plotted.

5.3 Baselines

For comparison, we establish the following baselines:

Target Only: Neither auxiliary data nor prior knowledge are involved in the baselines of this category. Since kernel machines working with multiple kernels are the most powerful methodologies for object recognition, we adopt two multi-kernel extensions of SVMs, including the *average kernel* (AK) suggested in [3] and the ensemble kernel learned by the *multiple kernel learning* (MKL) software [5]. The two baselines are respectively denoted as *AK [T Only]* and *MKL [T Only]*.

Source Only: We implement the *classemes* [23], a set of powerful features designed for transfer learning, to deliver knowledge from source to target. To this end, an SVM-based classifier with probabilistic outputs is learned for each source object class. The classemes are the probabilistic estimates obtained by applying these classifiers to the target data. While the procedure is performed for each of the adopted features, four new kernels based on classemes are constructed. Baselines *classemes+AK [S Only]* and *classemes+MKL [S Only]* are then respectively established by coupling AK and MKL to the four new kernels.

Target + Source: To fuse the information of both the two domains, we jointly consider the kernels yielded by visual features and classemes. Similarly, baselines *classemes+AK [T+S]* and *classemes+MKL [T+S]* are established. Besides, our approach is compared with *multi kernel transfer learning* (MKTL) [25], which is one of the best transfer learning algorithms and supports heterogeneous transfer from different kinds of priors.

5.4 Within-Database Transfer Learning

We first carry out within-database transfer learning on SUN09 and Caltech256 datasets. (MSRC dataset consists of only 22 categories, so it won't be adopted in the experiments of within-database transfer.) For each dataset, 10 target categories and 20 source categories are randomly selected. We extract 1 to 10 samples as well as 50 samples of each target class for training and testing respectively.

Table 1. The relative improvement in accuracy (%) and p value w.r.t AK [T Only].

Dataset	SUN09					Caltech256				
No. of samples per class	1	3	5	10	p	1	3	5	10	p
AK [T Only]	—	—	—	—	—	—	—	—	—	—
MKL [T Only]	5.39	3.91	3.60	3.56	0.97	14.00	7.70	13.73	15.40	0.11
classemes+AK [S Only]	-3.03	-0.90	-5.34	-10.78	0.31	-1.64	-14.39	-12.91	-18.81	0.03
classemes+MKL [S Only]	0.56	2.71	-0.62	-6.24	0.68	4.36	-13.45	-1.08	-14.11	0.11
classemes+AK [T + S]	8.20	6.32	4.10	2.58	0.97	2.91	0.71	1.50	0.69	0.97
classemes+MKL [T + S]	0.79	3.53	4.10	4.80	0.97	14.09	15.45	14.82	17.73	0.03
MKTL [25]	19.78	5.94	7.39	2.73	0.68	15.00	10.00	8.00	11.17	0.03
Ours	**41.80**	**23.83**	**16.09**	**11.76**	**0.11**	**38.82**	**26.38**	**24.01**	**20.88**	**0.006**

The number of samples in each source class is set as 50. All experiments are repeated ten times to reduce the effect of sampling. As for boosting iterations, we set U and V in Algorithm 2 and 3 as 300 and 25 respectively.

In Fig. 2, the average recognition rates of our approach and other baseline methods are plotted. Note that only the standard deviations of our approach are shown for the sake of clearness. The consistent trend of curves indicates that our approach outperforms the other baselines. Although baselines with only source priors, e.g., classemes+AK [S only], don't work well, incorporating the source knowledge indeed helps in constructing more accurate classifiers, e.g., classemes+AK [T+S]. Besides, baselines with MKL work better than baselines with AK in most of the cases. This is because SimpleMKL [5] effectively selects the optimal kernel combination to emphasize discriminant features for the given data. With regard to MKTL, it works better than all other baselines.

The relative improvement of recognition rates with respect to AK [T Only] is reported in Table 1, in which entries for 1, 3, 5, 10 samples per class are computed via $(A–B)/B\times 100\%$ where A is the recognition rate of each method and B is the recognition rate of AK [T Only]. Besides, the widely-used Kolmogorov-Smirnov test (KS test), a type of significance test, is also applied to the recognition rates of 1 to 10 samples per class (quantified by p value); the lower the p value is, the better the relative performance is. We observe that the less the training samples, the more helpful the source information is. The quantitative results validate that our method is indeed effective for object recognition with few training examples.

It is worthy of noting that, the effect of *higher start* [18] in our approach is quite significant in the cases where one training sample per class is available. The relative performance w.r.t AK [T Only] is about 42% in SUN09 or 39% in Caltech256. Moreover, this improvement is consistent even in the cases of 10 training samples per class. It reveals the merit in the aspect of *higher asymptote* [18].

5.5 Cross-Database Transfer Learning

We evaluate if the proposed approach works for cross-database transfer. The experimental setting is the same as the one described in section 5.4, except for data of the source domain and the target domain are from different databases. With Caltech256, MSRC, and SUN09, totally we have six combinations.

The recognition rates of our approach and all baselines are shown in Fig. 3. In the six settings of cross-database of transfer learning, our approach consistently achieves the best results. Besides, the effects of higher-start as well as higher-asymptote still remain in most cases. This validates that the proposed approach can explore commonly shared knowledge across databases, and use it to alleviate the difficulty caused by learning with few training examples.

In Fig. 3, it is observed that the performance gains of our approach with respect to baselines working on just target data, e.g., MKL [T Only], vary from case to case. We consider that it may result from the different degrees of similarity between the source and the target domains. As indicated in [31], each image in Caltech256 is usually pictured indoors, and always contains a single object. The images in MSRC, on the other hand, often contain the entire scene and are captured outdoors. For SUN09, objects may be photographed indoors or outdoors but are more similar to these in MSRC. Above observations explain why transferring knowledge between Caltech256 and MSRC is more difficult than between SUN09 and MSRC. Thus, the performance gains of our approach are relatively limited in cases of Fig. 3(c) and (e), but become significant in Fig. 3(b) and (f). To summarize, the proposed approach can effectively transfer useful knowledge from the source domain, and leads to significant improvements of performances in both the cases of within-database and cross-database transfer. This manifests its robustness and generality.

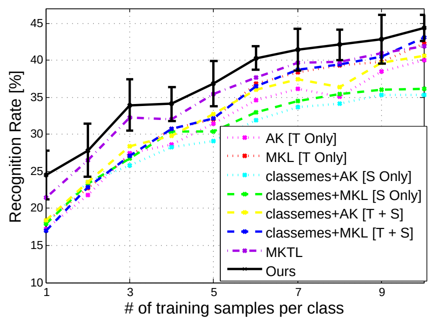
6 Conclusions

The proposed approach transfers knowledge from multiple classes to multiple classes by a two-layer boosting architecture. The outer layer discovers commonly shared attribute partitions, which cast two unrelated, multi-class learning problems into a series of binary, related ones. The inner layer explores useful source knowledge to learn discriminant and robust attribute classifiers in the target domain. Our approach is comprehensively evaluated with three benchmark datasets of object recognition. The significant improvements on recognition rates consolidate the usefulness of our approach. Moreover, the flexibility of our approach in cross-category and cross-database knowledge transfer makes it quite suitable to deal with increasingly complex recognition tasks, such as recognizing new categories or handling new datasets with few manually labeled data.

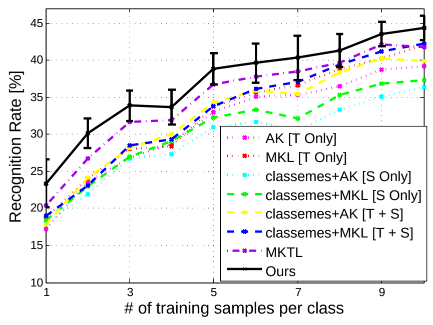
Acknowledgement. This study is conducted under the “III Innovative and Prospective Technologies Project” of the Institute for Information Industry which is subsidized by the Ministry of Economy Affairs of the Republic of China. It is also supported in part by grant NSC 101-2221-E-001-018.

References

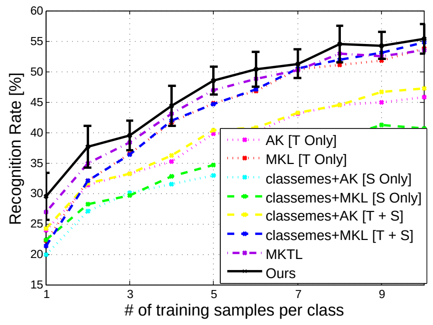
1. Lin, Y.Y., Liu, T.L., Fuh, C.S.: Multiple kernel learning for dimensionality reduction. TPAMI (2011)



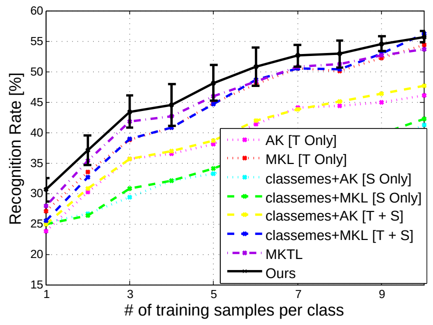
(a) Source: Caltech256, Target: SUN09



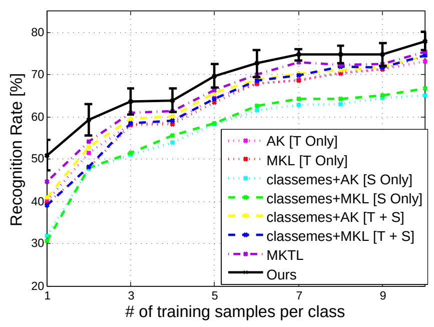
(b) Source: MSRC, Target: SUN09



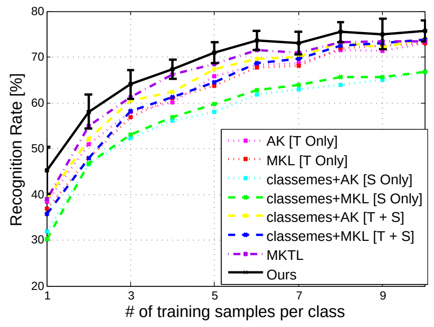
(c) Source: MSRC, Target: Caltech256



(d) Source: SUN09, Target: Caltech256



(e) Source: Caltech256, Target: MSRC



(f) Source: SUN09, Target: MSRC

Fig. 3. Transfer learning across Caltech256, MSRC and SUN09 datasets. Recognition rates are plotted as a function of the number of training samples per class.

2. Varma, M., Ray, D.: Learning the discriminative power-invariance trade-off. In: ICCV. (2007)
3. Gehler, P., Nowozin, S.: On feature combination for multiclass object classification. In: ICCV. (2009)
4. Bach, F.R., Lanckriet, G.R.G., Jordan, M.I.: Multiple kernel learning, conic duality, and the SMO algorithm. In: ICML. (2004)
5. Rakotomamonjy, A., Bach, F.R., Canu, S., Grandvalet, Y.: SimpleMKL. JMLR (2008)

6. Pan, S.J., Yang, Q.: A survey on transfer learning. TKDE (2010)
7. Dai, W., Yang, Q., Xue, G.R., Yu, Y.: Boosting for transfer learning. In: ICML. (2007)
8. Fei-Fei, L., Fergus, R., Perona, P.: One-shot learning of object categories. TPAMI (2006)
9. Yao, Y., Doretto, G.: Boosting for transfer learning with multiple sources. In: CVPR. (2010)
10. Schapire, R.: Using output codes to boost multiclass learning problems. In: ICML. (1997)
11. Torralba, A., Murphy, K., Freeman, W.: Sharing visual features for multiclass and multiview object detection. TPAMI (2007)
12. Lin, Y.Y., Tsai, J.F., Liu, T.L.: Efficient discriminative local learning for object recognition. In: ICCV. (2009)
13. Lowe, D.: Distinctive image features from scale-invariant keypoints. IJCV (2004)
14. Oliva, A., Torralba, A.: Modeling the shape of the scene: A holistic representation of the spatial envelope. IJCV (2001)
15. Farhadi, A., Endres, I., Hoiem, D., Forsyth, D.: Describing objects by their attributes. In: CVPR. (2009)
16. Lampert, C.H., Nickisch, H., Harmeling, S.: Learning to detect unseen object classes by betweenclass attribute transfer. In: CVPR. (2009)
17. Qi, G.J., Aggarwal, C., Rui, Y., Tian, Q., Chang, S., Huang, T.: Towards cross-category knowledge propagation for learning visual concepts. In: CVPR. (2011)
18. Tommasi, T., Orabona, F., Caputo, B.: Safety in numbers: Learning categories from few examples with multi model knowledge transfer. In: CVPR. (2010)
19. Kulis, B., Saenko, K., Darrell, T.: What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In: CVPR. (2011)
20. Duan, L., Tsang, I., Xu, D., Maybank, S.: Domain transfer SVM for video concept detection. In: CVPR. (2009)
21. Bickel, S., Bruckner, M., Scheffer, T.: Discriminative learning for differing training and test distributions. In: ICML. (2007)
22. Bart, E., Ullman, S.: Cross-generalization: Learning novel classes from a single example by feature replacement. In: CVPR. (2005) 672–679
23. Torresani, L., Szummer, M., Fitzgibbon, A.: Efficient object category recognition using classemes. In: ECCV. (2010)
24. Daume III, H.: Frustratingly easy domain adaptation. In: ACL. (2007)
25. Jie, L., Tommasi, T., Caputo, B.: Multiclass transfer learning from unconstrained priors. In: ICCV. (2011)
26. Dietterich, T., Bakiri, G.: Solving multiclass learning problems via error-correcting output codes. JAIR (1995)
27. Moghaddam, B., Shakhnarovich, G.: Boosted dyadic kernel discriminants. In: NIPS. (2002)
28. Griffin, G., Holub, A., Perona, P.: Caltech-256 object category dataset. Technical report, California Institute of Technology (2007)
29. Xiao, J., Hays, J., Ehinger, K., Oliva, A., Torralba, A.: SUN database: Large-scale scene recognition from abbey to zoo. In: CVPR. (2010)
30. Winn, J., Criminisi, A., Minka, T.: Object categorization by learned universal visual dictionary. In: ICCV. (2005)
31. Torralba, A., Efros, A.A.: Unbiased look at dataset bias. In: CVPR. (2011)
32. Varma, M., Zisserman, A.: A statistical approach to texture classification from single images. IJCV (2005)