Learning to Learn, from Transfer Learning to Domain Adaptation: A Unifying Perspective

Novi Patricia
Idiap Research Institute, 1920 Martigny, Switzerland
EPFL, 1015 Lausanne, Switzerland
novi.patricia@idiap.ch

Barbara Caputo
University of Rome La Sapienza
00185 Rome, Italy
caputo@dis.uniroma1.it

Abstract

The transfer learning and domain adaptation problems originate from a distribution mismatch between the source and target data distribution. The causes of such mismatch are traditionally considered different. Thus, transfer learning and domain adaptation algorithms are designed to address different issues, and cannot be used in both settings unless substantially modified. Still, one might argue that these problems are just different declinations of learning to learn, i.e. the ability to leverage over prior knowledge when attempting to solve a new task.

We propose a learning to learn framework able to leverage over source data regardless of the origin of the distribution mismatch. We consider prior models as experts, and use their output confidence value as features. We use them to build the new target model, combined with the features from the target data through a high-level cue integration scheme. This results in a class of algorithms usable in a plug-and-play fashion over any learning to learn scenario, from binary and multi-class transfer learning to single and multiple source domain adaptation settings. Experiments on several public datasets show that our approach consistently achieves the state of the art.

1. Introduction

The ability of learning to learn, shared by humans and animals, implies that the more categories a biological cognitive system knows, the better it gets at learning a new one. Since entering the Big Data age, the visual recognition community has moved from problems handling hundreds of categories [1], to the challenge of categorizing thousands and more classes [2]. As a consequence, the learning to learn paradigm has gained increasing attention, and several methods have been proposed for leveraging over prior models when attempting to learn a new task [3, 4, 5]. The problem is challenging because a core assumption in machine learning methods is that training and test images are drawn according to the same probability distribution [6]. This is not the case in the learning to learn scenario, where in general one attempts to leverage over existing *source* knowledge to solve a different *target* problem, where source and target present a distribution mismatch [7].

Learning to learn scenarios, and related algorithms, can thus be grouped according to what assumption is made to justify such distribution mismatch. For instance, the underlying assumption in domain adaptation is that the source and target domains are different in terms of marginal data distributions but have identical label sets. For transfer learning instead, the current working hypothesis is that the marginal distributions of data are related, but the source and target tasks have different label sets.

By making two different assumptions for the distribution mismatch experienced in the domain adaptation and transfer learning scenarios, it follows that methods developed to deal with one setting are not usable in the other, and vice versa. There are examples of algorithms designed for the transfer learning scenario and then successfully adapted to domain adaptation (such as [5, 8]), but they require substantial changes to move from one setting to another.

But is this desirable, application-wise? Consider for instance the case of a smartphone, equipped with an App able to recognize N object categories, asked by the user to learn new ones from very few annotated data. Assuming to have some source knowledge available, from the point of view of the system it doesn't really matter what the distribution mismatch is due to – and it might not be even possible to determine it a priori. All that matters is what source knowledge is available, and how to leverage over it so to bootstrap the learning of the new classes. In other words, artificial intelligent systems should be able to learn new target tasks from few annotated samples leveraging over prior source data drawn from a different probability distribution, regardless of the cause of such distribution mismatch.

This is what this paper is about. We propose an algorithm for leveraging over prior knowledge, that works regardless of how the distribution mismatch between source and target has been generated. We consider each source as an expert that judges on the new target samples. Thus, we treat the obtained confidence output as extra features, that we combine with the features from the target samples to build a target classifier. As opposed to [9], where the idea was exploited in a Multi Kernel Learning (MKL) framework for multi class transfer learning, we opt for a high level cue integration framework. This results in a more versatile algorithm, that can be applied on domain adaptation, binary and multi-class transfer learning problems with a plug-and-play approach. At the same time, our choice results in a better performance, compared to the MKL approach, on all these settings. Indeed, extensive experiments performed on several popular benchmark datasets in domain adaptation and transfer learning show that our approach leads consistently to state of the art performance on all of them, often with very consistent increase of performance compared to previous work. To the best of our knowledge, this is the first learning to learn approach usable on domain adaptation and transfer learning problems in a plug-and-play fashion. We call it High Level -Learning2Learn (H-L2L).

The rest of the paper is organized as follows: after a review of related work (section 2), we define our learning to learn framework (section 3) and describe how it can be casted into a high-level cue integration scheme (section 4). Sections 5.1–5.3 describe experiments in the domain adaptation (section 5.1), binary (section 5.2) and multi-class transfer learning scenarios (section 5.3). An overall discussion stating the challenges ahead concludes the paper.

2. Related Work

We briefly discuss previous work on domain adaptation and transfer learning; for a thorough review, see [7].

Domain Adaptation In domain adaptation, the focus is on how to deal with data sampled from different distributions, thus compensating for their mismatch. Although research efforts date back to 2006 [10], it has only recently attracted attention in the visual learning community [3, 11], thanks also to a renewed attention on generalization problems across different databases, and the subsequent dataset bias issue [12]. A popular trend is to focus on how to define procedures for transforming the image features. The goal here is to reduce the dissimilarity between domains, and thus make any classifier applicable. To this end, [3] learns a regularized transformation using information-theoretic metric learning that maps source data to the target domain. Gopalan *et al*. [11] instead projects both the source and target domain samples onto a set of intermediate subspaces, while Boqing Gong *et al*. [13] considers an infinite number of subspaces through a kernel-based approach. Another possible approach is based on classifier adaptation methods, mainly based on max-margin methods associated with strategies to adapt the learning parameters to novel problems [4, 8].

Transfer Learning Depending on the specific application scenario, the transferred knowledge can be in the form of instances, feature representation, or model parameters [14]. Instance transfer approaches assume that there are parts of the source data that can be sampled and considered together with the few available target labeled data [15, 16]. Parameter or model transfer approaches assume that the source and the target tasks share some parameters or prior distributions of the models. Fei-Fei *et al*. [17] proposed to transfer information via a Bayesian prior on object class models, using knowledge from known classes as a generic reference for newly learned models. Tommasi *et al*. [5] proposed a multi-source transfer model with a similar regularizer, where each source classifier was weighted by learned coefficients, obtaining strong results. Feature transfer approaches consist in learning a good representation for the target domain encoding in it some useful knowledge extracted from the source [18]. Several MKL methods were proposed for solving transfer learning problems. Jie *et al*. [9] suggested to use MKL kernel weights as source classifier weights, proposing one of the few truly multiclass transfer learning models in the literature.

Our work can be seen as a generalization of this last approach, on which to some extent we build. As in [9], we consider source knowledge classifiers as experts, and we also consider their output confidence as features to be combined with those coming from the target samples. While Jie *et al*. then pursue the weighting with a MKL approach, we opt for a high level cue integration framework: this gives us the generality needed to tackle the probability distribution mismatch across source and target data regardless of what causes it, while at the same time providing performances equal and often better than the state of the art in domain adaptation, binary and multi class transfer learning.

3. A Learning to Learn Framework

This section introduces formally the notation used in the paper. We indicate matrices and vectors with bold letters, and use $\bar{\boldsymbol{w}}$ to indicate the vector formed by the concatenation of the K vectors $\boldsymbol{w}^j$, hence $\bar{\boldsymbol{w}} = [\boldsymbol{w}^1, \boldsymbol{w}^2, \cdots, \boldsymbol{w}^K]$. We indicate with $X \in \mathcal{X}$ the data and with $Z \in \mathcal{Z}$ the corresponding labels, where $\mathcal{X}$ and $\mathcal{Z}$ specify respectively the feature and the label space. The indices S and T are used to indicate source and target domain.

As mentioned in the previous section, learning to learn has two popular instantiations: domain adaptation and transfer learning. In domain adaptation problem, we are given a set of identical label $\mathcal{Z}_S = \mathcal{Z}_T$ but different marginal distributions of the samples $P_S(X) \neq P_T(X)$. Then, transfer learning solves a problem that consists of dif-

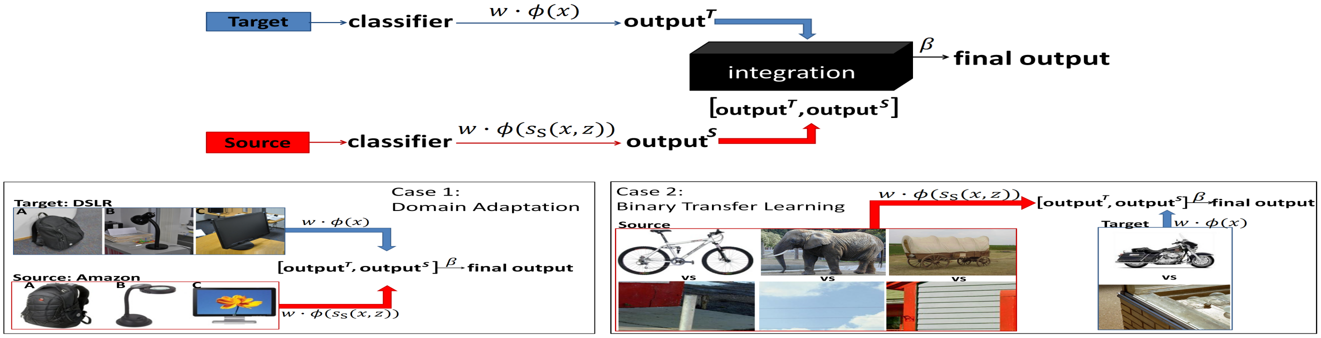

Figure 1. The top row shows the H-L2L framework, where each domain is trained independently and the output of each domain is concatenated into a new feature representation. Then, the system learns the model for the target task final jointly. The box on the bottom left shows an example of the H-L2L framework applied to the domain adaptation problem, where the source (Amazon) and the target (DSLR) have the same class labels (bag, lamp and monitor). The box on the bottom right shows an example of H-L2L applied to binary transfer learning, where the source and target belong to the same domain (Caltech-256 dataset). The source models (bicycle, elephant and covered wagon) are used to learn the new target classifier (motorbike). Note that H-L2L integrates the source confidence knowledge with the target features in the exact same fashion in both cases.

ferent label sets $\mathcal{Z}_S \neq \mathcal{Z}_T$, but the marginal distributions of the data are related $P_S(X) \sim P_T(X)$.

Our goal is to formalize the problem of learning a classifier on a target set for which (a) few labeled training data are available but (b) we have many source sets, in the hypothesis of a distribution mismatch between the target and the sources, and across the sources. As opposed to previous work, we do not want to explicitly model the origin of such distribution mismatch, but we wish to derive a framework general enough to be applicable and effective in several settings, from domain adaptation to transfer learning.

To be as general as possible, we assume to have multiple sources $S(m)$, $m = 1, \ldots, M$ and a single target T, where there might be a domain shift between (some of) the sources and the target, and the label set between source(s) and target might be perfectly overlapping $\mathcal{Z}_{S(m)} = \mathcal{Z}_T$, only partially overlapping $\mathcal{Z}_{S(m)} \cap \mathcal{Z}_T \neq \varnothing$, or completely disjoint $\mathcal{Z}_{S(m)} \cap \mathcal{Z}_T = \varnothing$. The difference in the domains can be caused by both $P_{S(m)}(X) \neq P_T(X)$ and $\mathcal{X}_{S(m)} \neq \mathcal{X}_T$. For sake of simplicity, we will also drop the 'm' index from now on.

Considering that each domain has $z = 1, \ldots, F (F \geq 2)$ categories, we are interested to model the categories of each domain via a function:

$$s(\boldsymbol{x}, z) = \boldsymbol{w} \cdot \phi(\boldsymbol{x}, z) \qquad (1)$$

where $\boldsymbol{w}$ is a hyperplane and $\phi(\cdot, \cdot) \rightarrow X \times Z : H$ is the joint feature mapping function [19]. The score function $s(\boldsymbol{x}, z)$ provides confidence scores for the new sample $\boldsymbol{x}$, to the assigning label z, instead of a decision (hard labels). Then, the predictive function of the new sample can be expressed with $f(\boldsymbol{x}) = \underset{z \in Z}{\operatorname{argmax}}\ s(\boldsymbol{x}, z)$ for multi-class problem and $f(\boldsymbol{x}) = \operatorname{sign}(s(\boldsymbol{x}))$ for binary classification. In the rest of the paper, we will only describe our model for the multi-class situation, as its modification to the binary case is straightforward. We treat the confidence scores as a new feature representation of the data, instead of the original features (*e.g.* a bag-of-visual-words histogram of the image). Hence, we are able to leverage over prior knowledge from all the sources when learning the new target, whether we find ourselves in a semi-supervised domain adaptation scenario, whether we are in a transfer learning setting.

4. A High-level Learning to Learn Framework

The straightforward way to combine different features together is to use a cue integration algorithm. Here we first briefly review the high level cue integration strategy [20], then we show how to cast the learning to learn framework into it, and we describe into details two specific algorithms that we we will then test in various settings. In the following, we will suppose to have a training set $\{\boldsymbol{x}_i, z_i\}_{i=1}^N$ with F features and ϕ^j representing the j-th feature mapping.

High-level integration. A classifier is trained for each feature, then each classifier provides confidence scores for the new sample. Depending on the type of outputs from the feature classifiers, these outputs can be combined to make a final decision.

$$s(\boldsymbol{x}, z) = \sum_{j=1}^{F} \beta_z^j s^j(\boldsymbol{x}) = \sum_{j=1}^{F} \beta_z^j \boldsymbol{w}_z^j \cdot \phi^j(\boldsymbol{x}), \qquad (2)$$

where β_z^j are weights which define how much the integration classifier should trust the j-th classifier. The high-level integration could also be perceived as a two-layers scheme. A classifier is trained for each feature in the first layer, where we could use different types of learning algorithms to obtain the confidence score. In the second layer, the confidence are combined with different flavors. In this level, $\boldsymbol{w}_z^j$ are learnt independently and $\boldsymbol{\beta}$ are learnt jointly [20].

H-L2L framework. We propose to use a high-level integration scheme into the learning to learn framework. We are interested in the task of learning a classifier for F_T target categories, given a training set $\{\boldsymbol{x}_i, z_T^i\}_{i=1}^N$, with N small. As in [9], we propose to incorporate the predictions of prior source models with the training samples as auxiliary features. In addition to the training sample $\boldsymbol{x}_i$, we also gather the scores $s_\mathrm{S}(\boldsymbol{x}_i, z_i)$, predicted by the source models, this is the first stage. At the second stage, we learn the output of the target and source classifiers, through the standard linear model. Therefore, when learning a new category the score function is:

$$\begin{aligned} s(\boldsymbol{x}, z_T) &= \boldsymbol{\beta}\bar{\boldsymbol{w}} \cdot \bar{\boldsymbol{\phi}}(\boldsymbol{x}, z_T) \\ &= \beta^{(0)}\boldsymbol{w}^{(0)} \cdot \phi^{(0)}(\boldsymbol{x}, z_T) \\ &+ \sum_{z=1}^{F_S} \beta^{(z_T,z)}\boldsymbol{w}^{(z_T,z)} \cdot \phi^{(z_T,z)}\left(s_\mathrm{S}\left(\boldsymbol{x}, z\right), z_T\right). \end{aligned} \tag{3}$$

Here, $s_\mathrm{S}\left(\boldsymbol{x}, z\right)$ is the score of $\boldsymbol{x}$ labeled as class z_T predicted by the source models. We use the index 0 to indicate the feature mapping function $\phi^{(0)}(\boldsymbol{x}, z_T)$ for the original input features $\boldsymbol{x}$ and their corresponding model parameters $\boldsymbol{w}^{(0)}$. The indices (z_T, z) correspond to the feature mapping of $s_\mathrm{S}\left(\boldsymbol{x}, z\right)$ to the z_T-th new class, where $z_T = 1, \ldots, F_T$ and $z = 1, \ldots, F_S$. In other words, given the score $s_\mathrm{S}(\boldsymbol{x}, z)$ produced by a source prior, $\boldsymbol{w}^{(z_T,z)}$ represents the contribution of the z-th source model in predicting that $\boldsymbol{x}$ belongs to class z_T. $\boldsymbol{\beta}$ is a weight vector, resulted from second layer classifier, with $\beta^{(0)}$ indicates the weight from target classifier and $\beta^{(z_T,z)}$ corresponds to the z-th source model belongs to the z_T-th class.

Figure 1 illustrates the approach when applied to the domain adaptation and binary transfer learning cases. For instance in this last case, the intuition is that if the source knowledge of a bicycle gives a high score to images of a motorbike, this information may also be useful in the score function of motorbikes, since the two classes share common visual properties. Therefore, we might expect that the model will give to this source knowledge a higher weight. On the contrary, we expect lower weights for classes which are not very relevant, such as covered-wagon. Again, the predicted label is the class achieving the highest score.

We now propose two different algorithms which can be plugged into the high-level learning to learn framework. We consider the methods from [21] and [22], both exploiting the idea of high-level feature selection. In the first stage, we use LS-SVM classifiers on target and source domains to generate the output confidence on each domain. Here, $\bar{\boldsymbol{w}}$ is learnt independently within the classifiers. In this stage, we are free to choose the classifier types, however we need to differentiate the classifiers at each stage, to avoid overfitting. Then, we learn the joint weight $\boldsymbol{\beta}$, through the following methods:

H-L2L(SVM-DAS): In this approach, we simply augment the output confidence from target and source domain into a new feature representation. In case of one target and a single source domain, then the dimension of this new vector is $\mathbb{R}^{(F_T+F_S)}$. The parameters $\beta^{(\cdot)}$ and the support vectors in eq. 3 are inferred from the training data either directly or efficiently during the optimization process [21].

H-L2L(LP-β): This method uses a boosting approach (*e.g. weak learners*) for learning the mixing weights. With the mixing coefficients $\beta^{(\cdot)}$ summing to one, the decision function is a convex linear combination of the real output of SVM. In eq. 2, $s^j(\boldsymbol{x})$ are some real valued functions, not necessarily SVM. In Boosting, the s^j are also known as *weak learners*. From each s^j, we get a set of parameters $\{\boldsymbol{w}_z^j\}$, then subsequently we can optimize $\boldsymbol{\beta}$ using any linear programming solvers. In the case where $\beta^{(\cdot)} = 0$, the feature does not need to be computed for the final decision function. The algorithm proposed in [22] optimizes $\boldsymbol{\beta}$ through a hyperparameter ν, which trades the smoothness of the resulting function with the hinge loss on the points, equivalently to the SVM regularization parameter C.

The idea of leveraging source models/learning weights have been proposed in several works [23, 24, 25]. However, in this first implemetation, we let the methods, *i.e.* SVM-DAS and LP-β , to learn the weight parameters by themselves. We did not develop a procedure for tuning the weights, as we want to demonstrate that leveraging over priors can be applied in H-L2L framework without having to make strong assumptions on the underlying probability distributions of source and target data.

5. Experiments

In this section we describe the experiments made to evaluate the H-L2L framework on domain adaptation (section 5.1), binary (section 5.2) and multi-class transfer learning (section 5.3) problems. For each scenario, we used databases and experimental setup already used before in the literature, benchmarking against several published methods. This should ensure a fair assessment of our approach, while demonstrating its versatility.

5.1. Domain Adaptation Experiments

For the domain adaptation experiments, we used the Office dataset[1], which contains three domains: Amazon (images from the online merchants, downloaded from www.amazon.com), DSLR and Webcam (images captured with a DSLR camera and webcam in realistic environments with various lighting conditions). The dataset consists of 31 categories, in an office environment. This dataset has been proposed first in [3], and it has since become a standard reference benchmark for domain adaptation algorithms. We

[1] http://www1.icsi.berkeley.edu/~saenko/projects.html#data

also follow [13], and use a fourth domain, extracted from Caltech-256 dataset. The so created Office-Caltech dataset[2] takes 10 classes common to all four datasets. On these data, we follow the feature extraction and experimental protocols from [3], in the semi-supervised setting. Each image uses SURF features encoded in a bag of words histogram. Then, it is quantized to 800 histogram with the codebook trained on subset of Amazon images. To learn the source models, 8 samples per class are used for Webcam and DSLR, 20 for Amazon and Caltech. To learn the target model, 3 samples per class are used for training; the remaining images are used as test set. These settings are applied on both 10 and 31 classes datasets.

The H-L2L framework is tested using its two instantiations described in section 4, i.e. H-L2L(SVM-DAS) and H-L2L(LP-β). For both algorithms, we set the regularization parameter C between $\{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$. We use Gaussian kernel with γ equal to the mean of pairwise distances. The training phase is divided into two stage scheme to avoid biased estimates. First we compute model selection (LS-SVM) using 5 fold cross-validation (CV1) to select best C on source and target domain. For each fold, we compute the output confidence on the remaining samples using the best C identified before. The output confidence will be treated as a new feature to the final classifier. For the final hypothesis, we train the classifier using traditional SVM[3], with one-vs-all extension for multi-class. Here, we also choose the best C and computing the kernel as in the first stage. For choosing the best ν in H-L2L(LP-β), we generate another 5 fold cross-validation (CV2) on each existing fold (CV1). Hence, we use minimal 5 number of target samples on the first fold (CV1). We benchmark the two H-L2L variants against the following methods:

No-Transfer(SVM): It corresponds to traditional supervised learning without considering any prior knowledge using SVM method.

No-Transfer(AdaBoost): This standard supervised learning uses AdaBoost technique, which gives a fair comparison for two instantiations of the H-L2L framework.

Prior-Features: The output of all the prior models are considered as features. We concatenate them into a new vector representation and apply a linear SVM classifier.

Metric: The metric learning approach proposed in [3].

SGF: The method represents the source and target domain as points on a Grassmann manifold and all the samples are projected onto the geodesic flow between them. This approach uses the intermediate subspaces to learn domain-invariant features to adapt [11].

GFK: a simplification of SGF: instead of taking a given number of subspaces to sample, it integrates an infinite number of subspaces that characterize changes from the source to the target domain [13].

Method	W → D	A → W	D → W
No-Transfer(SVM)	49.6 ± 0.03	50.7 ± 0.03	49.6 ± 0.03
No-Transfer(AdaBoost)	54.5 ± 0.01	51.3 ± 0.02	51.8 ± 0.01
Prior-Features	43.8 ± 3.00	25.6 ± 2.64	48.8 ± 3.20
Multi-perclass-Adapt	59.5 ± 3.50	52.9 ± 1.60	58.7 ± 1.70
MKAL	48.3 ± 0.01	47.1 ± 0.03	48.3 ± 0.02
Metric [3]	48.1 ± 0.60	34.5 ± 0.70	36.9 ± 0.80
SGF [11]	61.0 ± 0.50	37.4 ± 0.50	55.2 ± 0.60
GFK [13]	66.3 ± 0.40	46.4 ± 0.50	61.3 ± 0.04
H-L2L(SVM-DAS)	55.5 ± 0.03	52.0 ± 0.01	59.5 ± 0.02
H-L2L(LP-β)	**67.8 ± 0.05**	**58.8 ± 0.03**	**66.0 ± 0.03**

Table 2. Accuracy on target domains, *semi-supervised* adaptation, 31 classes (A:Amazon, D:DSLR, W:Webcam).

Method	A+W → D	A+D → W	D+W → A
No-Transfer(SVM)	48.3 ± 0.02	49.3 ± 0.03	21.1 ± 0.01
No-Transfer(AdaBoost)	54.1 ± 0.01	51.1 ± 0.18	20.2 ± 0.08
Prior-Features	43.2 ± 2.80	48.6 ± 2.80	16.1 ± 1.54
Multi-perclass-Adapt	57.3 ± 2.90	50.8 ± 0.03	11.3 ± 0.03
MKAL	48.6 ± 0.04	47.3 ± 0.03	20.1 ± 0.01
SGF [13]	39.0 ± 1.10	52.0 ± 2.50	**28.0 ± 0.80**
H-L2L(SVM-DAS)	53.6 ± 0.03	58.0 ± 0.03	20.7 ± 0.02
H-L2L(LP-β)	**67.9 ± 0.05**	**66.1 ± 0.02**	25.8 ± 0.01

Table 3. Accuracy on target domains, *semi-supervised* adaptation, 31 classes and multi sources (A:Amazon, D:DSLR, W:Webcam).

Multi-perclass-Adapt: it is an extension of [5] to the DA setting, where a weighting matrix learned through a leave-one-out procedure determines how much the source models contribute when learning the target classifier [8].

MKAL: it is a recently proposed extension of [9] to the single and multi-class DA problem, that uses the MKL framework to combine the source confidence output with the target features [26].

Tables 1–3 show the obtained classification accuracies, for all methods, using the settings described above, for single (Table 1-2) and multi-source (Table 3) scenarios. For the single source experiments, we see that H-L2L(LP-β) achieves consistently state of the art results, with the only exception of the single source, 10 classes experiments for the $A \rightarrow C$ case (Table 1). Indeed, in some cases the increase in performance is quite high, with a peak of $+24.6\%$ in accuracy compared to the previous state of the art ($C \rightarrow D$, Table 1). Results obtained by H-L2L(SVM-DAS) are also good, but for this algorithm the advantage over previously published results is less clear. Scaling over the number of classes does not seem to affect these behaviors.

These results are consistent with those obtained in the multiple source setting (Table 3), where again H-L2L(LP-β) achieves the state of the art in two cases out of three. It appears that, for the DA scenario, a boosting-based approach is better suited for learning how to exploit the information contained into the source confidence output, when building the target classifier.

5.2. Binary Transfer Learning Experiments

We did run all experiments on different subsets of the Caltech-256 database, which contains images of 256 classes plus a background category (negative class) that can be used

[2] http://www-scf.usc.edu/~boqinggo/domainadaptation.html

[3] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

Method	C → A	D → A	W → A	A → C	W → C	C → D	A → W	D → W
Prior-Features	40.4 ± 5.06	34.3 ± 3.02	35.2 ± 5.14	28.2 ± 5.60	21.9 ± 4.54	46.6 ± 3.51	37.7 ± 2.20	68.8 ± 4.20
Multi-perclass-Adapt	33.0 ± 0.05	30.4 ± 0.04	28.9 ± 0.03	31.1 ± 0.04	20.0 ± 0.01	35.0 ± 0.04	23.3 ± 0.10	67.2 ± 0.04
MKAL	43.5 ± 0.03	43.5 ± 0.04	43.8 ± 0.03	28.4 ± 0.02	26.6 ± 0.03	58.7 ± 0.06	68.5 ± 0.03	66.6 ± 0.03
Metric [3]	33.7 ± 0.80	30.3 ± 0.80	32.3 ± 0.80	27.3 ± 0.70	21.7 ± 0.50	35.0 ± 1.10	36.0 ± 1.00	55.6 ± 0.70
SGF [11]	40.2 ± 0.70	39.2 ± 0.70	38.2 ± 0.60	37.7 ± 0.50	29.2 ± 0.70	36.6 ± 0.80	37.9 ± 0.70	69.5 ± 0.90
GFK [13]	46.1 ± 0.60	46.2 ± 0.60	46.2 ± 0.70	**39.6 ± 0.40**	32.1 ± 0.70	55.0 ± 0.90	56.9 ± 1.00	80.2 ± 0.40
H-L2L(SVM-DAS)	47.1 ± 0.02	46.0 ± 0.01	44.9 ± 0.02	30.0 ± 0.04	27.5 ± 0.04	72.2 ± 0.12	71.7 ± 0.10	78.1 ± 0.02
H-L2L(LP-β)	**55.3 ± 0.02**	**52.7 ± 0.04**	**51.6 ± 0.03**	38.6 ± 0.02	**34.0 ± 0.02**	**79.6 ± 0.10**	**77.1 ± 0.10**	**81.8 ± 0.03**

Table 1. Accuracy on target domains, *semi-supervised* adaptation, 10 classes (A:Amazon, C:Caltech, D:DSLR, W:Webcam).

in object-vs-background problems. We downloaded[4] the pre-computed features and selected four: PHOG Shape Descriptor, SIFT Apperance Descriptor, Region Covariance and Local Binary Patterns. They were all computed in a spatial pyramid and we use only the first level. We followed the experimental setting used in [5] and analyze the behavior of H-L2L(SVM-DAS) and H-L2L(LP-β) for an increasing number of source models, when only few target samples are available, and when the number of training samples for the target increases. The parameters for the H-L2L algorithms are chosen as described in section 5.1. We benchmarked against the following methods:

No-Transfer(SVM): It uses the standard supervised task without considering any prior knowledge. We train SVM classifiers using a one-vs-all scheme for multi-class problem.

No-Transfer(AdaBoost): This is a standard AdaBoost applied on target domain only, without using any source information.

Prior-Features: We concatenate the output of prior models as feature descriptors and use a linear SVM classifier to test their performance. The method helps us to see the role of the prior models in the performance.

Multi-KT: The method assumes that both the prior models and the new model use the same feature representation and the same classifier. Here, we consider the ℓ_2 norm constraint, where at each iteration the learn parameters are projected onto the ℓ_2-sphere [5].

MKTL: the approach proposes to leverage the source models by using the output of their classification as a features into a MKL framework. It can be seen as a mid-level integration instantiation of our learning to learn framework. MKTL gives the possibility to tune the level of sparsity of the kernels, as it is extended to ℓ_p norm regularization. We set the constraint $p = \frac{2\log K}{2\log K - 1}$, where K is the number of kernels [9].

We made a first set of experiments considering a small number of prior source models, and studied the H-L2L behavior in the case of related, unrelated and mixed priors. To this end, we considered 6 unrelated classes (harp, microwave, fire-truck, cowboy-bat, snake, bonsai), 6 related classes (all vehicles: bulldozer, fire-truck, motorbikes, school-bus, snowmobile, car-side) and 10 mixed classes (motorbikes, dog, cactus, helicopter, fighter, car-side, dolphin, zebra, horse, goose) from the Caltech-256 database, as done in [5]. Result are showed in Figure 2, top row. We see that, for all three settings, H-L2L(SVM-DAS) obtains the strongest performance, especially in the '6 classes, unrelated' experiment (Figure 2, top row, left). In this case, the advantage over Multi-KT, the current state of the art in binary transfer learning, is quite remarkable, while in the '6 classes, related' and '10 classes, mixed' setting the improvement is less marked, although still there. It is interesting to observe that Prior Features also always obtains a strong result. This indicates that the source confidence outputs are indeed very informative[5]. Still, of all the cue integration-based methods used, only H-L2L(SVM-DAS) seems to be able to take advantage from it. We see that No-Transfer(SVM) performs much better than No-Transfer(AdaBoost), the boosting-based algoirithms tend to suffer on transfer learning problems (this has been observed in [27]). H-L2L(LP-β) is always better w.r.t. No-Transfer(AdaBoost), although it does not improve as the number of target samples grows. This might be because, as the training data of source and target are unbalanced, the weights relative to be contribution of the target samples go quickly to 0 (confirmed in our experiments, Figure 2). This behavior has been observed in the literature [27]. Another issue is, we use the idea of feature selection proposed in [22] as a black-box, where we do not have a freedom to control the feature weights. While the boosting algorithms for transfer learning differentiates the weights between target and source domain [15].

We also did run experiments on 20 and 100 randomly extracted classes, and all 256 classes, to see how behaviors change when scaling over the number of prior sources. We extracted a combination of 80 object and 80 background images for each class. In the target domain, we used 20 training and 100 testing samples with half positive and half negative instances. Figure 2, bottom row, shows the obtained results. We see that H-L2L(SVM-DAS) keeps achieving the best performance, but as the number of prior sources grows, its results are identical to those obtained by Prior Features. This is reasonable, as in this setting the weight of the contribution coming from the target data has a lower impact. The performance of MKTL is roughly on par with what achieved by Multi-KT, apart for the very small sam-

[4] http://files.is.tue.mpg.de/pgehler/projects/iccv09/

[5] Although this might seem counterintuitive in the 'unrelated' case, a similar behavior was observed also in [17].

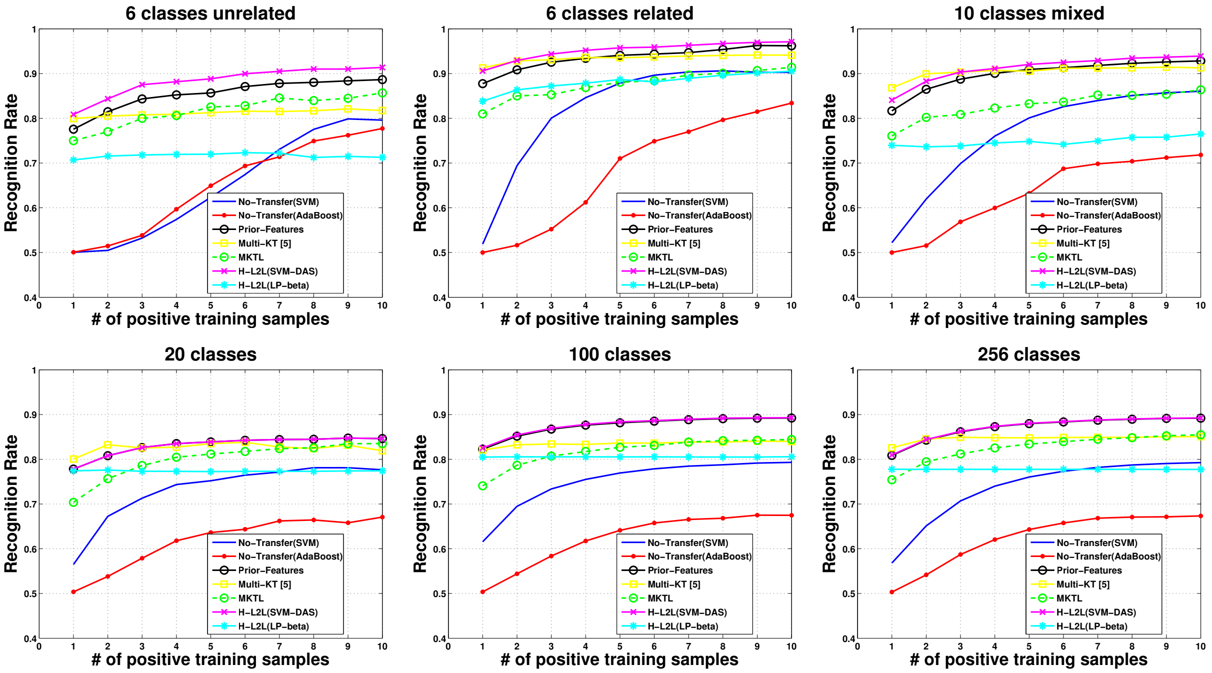

Figure 2. Top row: performance of binary transfer learning with small number of sources and training samples. Bottom row: Performance of binary transfer learning when increasing the number of sources from 20 until all classes of Caltech-256. All results correspond to average accuracy over the categories, over ten different splits considering in turn one of the classes as target and the others as source.

ples regime (as noted also in [9]). The poor performance obtained by H-L2L(LP-β) confirms the challenges that this version of the algorithm has in this setting.

5.3. Multi-class Transfer Learning Experiments

As a final experiment, we tested our algorithm on the multi-class transfer learning problem [9]. We used the Animal with Attributes dataset [28], which consists of 50 animal categories and several pre-extracted features for each image[6]. We followed the same settings from [9], with SURF features and color histogram for describing all the prior classes and PHOG feature for describing the target domain. Then, we built 40 classes as the prior knowledge sources and consider the remaining 10 classes as new classes to learn. We randomly extract a maximum of 100 training samples from each class and 50 test samples. We benchmarked H-L2L(SVM-DAS) and H-L2L(LP-β) against the No Transfer, Prior Features and MKTL algorithms described in section 5.2. Results are given in Figure 3. We see that H-L2L(SVM-DAS) is on par with MKTL (this is confirmed by the p-test, $p < 0.001$), with both methods able to increase their performance over Prior Features. The poor performance of H-L2L(LP-β), clearly experiencing negative transfer in this setting, make us suspect that the procedure used for setting the parameter ν leads to overfitting in transfer learning scenarios.

[6] http://attributes.kyb.tuebingen.mpg.de/

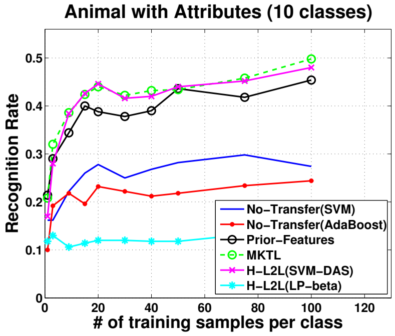

Figure 3. Result obtained on the multi-class transfer learning setting. Each experiment was repeated ten times on different data partitions. Result are an average over all runs.

6. Discussion and Conclusions

From the results reported in sections 5.1–5.3, we can draw two conclusions. The first is that the high level learning to learn framework proposed in this paper is indeed able to solve the distribution mismatch between source and target data, without having to make any assumption on what are the causes of such mismatch. As a consequence, the method is applicable on a much wider range of learning to learn problems. Indeed, of the three scenario considered (domain adaptation, binary and multi-class transfer learning), the H-L2L algorithm, in its two versions, is the only learning method that it has been possible to use on all three. This demonstrates that leveraging over priors can be addressed successfully without having to make strong assumptions on the underlying probability distributions of source and target data. We consider this result the key contribution of this paper.

The second conclusion is that, while every high-level cue integration method can be used in the L2L framework, not all of them are likely to obtain strong performances on every possible scenario. Indeed, our experiments showed that while SVM-DAS obtains competitive and basically stable results on all the considered scenarios, LP-β yields disappointing results on transfer learning problems, while achieving the state of the art on the domain adaptation setting. Our choice of using acritically two existing high-level cue integration algorithms that had show to work well on visual data has been deliberate, and functional to emphasize the versatility of our approach. Still, we see as a necessary and important future development casting our work into the ensemble learning framework. This should permit us to derive principled way to design (or chose) integration methods able to preserve the generality we aim for while achieving competitive performance, possibly also with some theoretical guarantees.

Acknowledgments

This work was partially supported (N. P.) by the SNSF project vision@home (SIVI).

References

[1] G. Griffin, A. Holub, and P. Perona, "Caltech 256 object category dataset," CalTech, Tech. Rep. UCB/CSD-04-1366, 2007. 1

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. CVPR*, 2009. 1

[3] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Proc ECCV*, 2010. 1, 2, 4, 5, 6

[4] A. Khosla, T. Zhou, T. Malisiewicz, A. Efros, and A. Torralba, "Undoing the damage of dataset bias," in *Proc. ECCV*, 2012. 1, 2

[5] T. Tommasi, F. Orabona, and B. Caputo, "Learning categories from few examples with multi model knowledge transfer," *PAMI, to appear*, 2013. 1, 2, 5, 6

[6] L. G. Valiant, "A theory of the learnable," *Communications ACM*, vol. 27, no. 11, pp. 1134–1142, 1984. 1

[7] T. Tommasi, "Learning to learn by exploiting prior knowledge," Ph.D. dissertation, EPFL, 2013. 1, 2

[8] T. Tommasi, F. Orabona, C. Castellini, and B. Caputo, "Improving control of dexterous hand prostheses using adaptive learning," *IEEE Transaction on Robotics*, 2012. 1, 2, 5

[9] J. Luo, T. Tommasi, and B. Caputo, "Multiclass transfer learning from unconstrained priors." in *Proc ICCV*, 2011. 2, 4, 5, 6, 7

[10] J. Blitzer, R. McDonald, and F. Pereira, "Domain adaptation with structural correspondence learning," in *Proc. EMNLP*, 2006. 2

[11] R. Gopalan, R. Li, and R. Chellappa, "Domain adaptation for object recognition: An unsupervised approach," in *Proc. ICCV*, 2011. 2, 5, 6

[12] A. Torralba and A. A. Efros, "Unbiased look at dataset bias," in *Proc. CVPR*, 2011. 2

[13] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proc CVPR*, 2012. 2, 5, 6

[14] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010. 2

[15] W. Dai, Q. Yang, G. Xue, and Y. Yu, "Boosting for transfer learning," in *Proc ICML*, 2007. 2, 6

[16] J. J. Lim, R. Salakhutdinov, and A. Torralba, "Transfer learning by borrowing examples for multiclass object detection," in *Proc. NIPS*, 2011. 2

[17] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *PAMI*, vol. 28, pp. 594–611, 2006. 2, 6

[18] U. Rückert and S. Kramer, "Kernel-based inductive transfer," in *Proc. ECML PKDD*, 2008. 2

[19] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun, "Support vector machine learning for interdependent and structured output spaces," in *Proc. ICML*, 2004. 3

[20] L. Jie, "Open ended learning of visual and multimodal pattern," Ph.D. dissertation, EPFL, 2011. 3

[21] A. Pronobis, O. M. Monoz, and B. Caputo, "Svm-based discriminative accumulation scheme for place recognition," in *Proc. ICRA*, 2008. 4

[22] P. Gehler and S. Nowozin, "On feature combination for multiclass object classification," in *Proc. ICCV*, 2009. 4, 6

[23] J. Yang, R. Yan, and A. G. Hauptmann, "Cross-domain video concept detection using adaptive SVMs," in *International conference on Multimedia (ICM)*, 2007. 4

[24] J. Yang and A. G. Hauptmann, "A framework for classifier adaptation and its applications in concept detection." in *Multimedia Information Retrieval*. ACM, 2008. 4

[25] L. Duan, D. Xu, I. W.-H. Tsang, and J. Luo, "Visual event recognition in videos by learning from web data." in *CVPR*, 2010, pp. 1959–1966. 4

[26] N. Patricia, T. Tommasi, and B. Caputo, "Multi-source adaptive learning for fast control of prothetics hand," in *ICPR, to appear*, 2014. 5

[27] S. Al-Stouhi and C. K. Reddy, "Adaptive boosting for transfer learning using dynamic updates," in *ECML/PKDD (1)*, 2011. 6

[28] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between class attribute transfer," in *Proc CVPR*, 2009. 7