Learning categories from few examples with multi model knowledge transfer

Learning Categories from Few Examples with Multi Model Knowledge Transfer

Tatiana Tommasi, Francesco Orabona and Barbara Caputo

Abstract—Learning a visual object category from few samples is a compelling and challenging problem. In several real-world applications collecting many annotated data is costly and not always possible. However a small training set does not allow to cover the high intraclass variability typical of visual objects. In this condition, machine learning methods provide very few guarantees. This paper presents a discriminative model adaptation algorithm able to proficiently learn a target object with few examples by relying on other previously learned source categories. The proposed method autonomously chooses from where and how much to transfer information by solving a convex optimization problem which ensures to have the minimal leave-one-out error on the available training set. We analyze several properties of the described approach and perform an extensive experimental comparison with other existing transfer solutions, consistently showing the value of our algorithm.

Index Terms—Knowledge Transfer, image categorization, discriminative learning

1 Introduction

As human beings, our learning ability develops progressively in time. At the age of six, we recognize around 10^4 object categories [1] and we go on learning more while we grow up. All the information acquired through our five senses is encoded and stored in memory, with concepts and categories organized on the basis of their common properties. This means that any new concept is not learned in isolation, but considering connections to what is already known, which makes the skill of building analogies one of the cores of human intelligence [2]. Even focusing only on visual tasks, we can give several examples of this cognitive ability. Have you ever seen a *guava* or an *okapi*? The guava is a fruit that externally looks like a lime, while its inner part is similar to an apple. An okapi is an animal that can be roughly described as a horse, with the legs of a zebra and the head of a giraffe (see Figure 1). Once we have seen a single image for each of the two target objects, we can easily memorize and recognize them by referring to the source objects mentioned in the provided description. In psychology this process is known as *knowledge transfer*: it encompasses phenomena ranging from simple (e.g. generalization of conditioned response between familiar and novel stimuli) to extremely complex (e.g. carrying over a solution from a problem in arithmetic to a novel class of problems) behaviors [3], and it makes learning further concepts extremely efficient. This capacity allows us to mine many kinds of recurrent patterns and to make inductive inferences on a new task even with only a small amount of data.

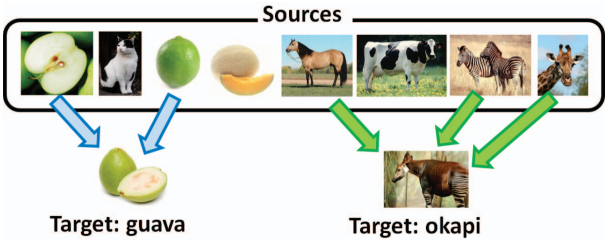

Fig. 1. Two examples of using some source knowledge on fruits and animals while learning the target objects *guava* and *okapi*.

A large part of recent literature on visual object categorization focuses on reaching impressive results on large and difficult datasets [4], [5]. However, these works rarely refer to the effort done in collecting the data. In many real applications gathering fully annotated images can be extremely time consuming and might have a significant impact on the overall cost of the final system. On the other hand, standard learning techniques do not handle well the case of very small training sets. Differently from the described cognition mechanism, all the learning approaches consider each task separately with respect to other possible source of relative information. Reproducing the knowledge transfer process in this scenario might consistently boost the learning performance. The basic intuition is that, if a system has already learned j categories, learning the $(j+1)$-th should be easier even from one or few training samples [6].

A first practical implementation of the knowledge transfer idea was presented in [7] following a Bayesian approach. A generic object model is estimated from some source categories and it is then used as prior to evaluate the target object parameter distribution with a maximum-a-posteriori technique. This work left some open questions discussed in its conclusive section: (i) All the different known source categories are used

together to define a single prior; would a more sophisticated *multi-modal prior* be beneficial in learning? (ii) Is there any other productive point of view beside the *generative Bayesian* one that allows to incorporate prior knowledge? (iii) Is it easier to learn new target categories which are *similar* to some of the source categories? Several other works in the computer vision literature followed this first attempt [8], [9], [10], [11], introducing different methods to increase the categorization performance with respect to learning from scratch in case of few available samples. However, due to the small differences in the chosen settings, the proposed solutions were never compared among each other.

In this work we focus on knowledge transfer across visual object categories and our main contribution is a learning algorithm that directly addresses the open problems in [7]. We consider (i) the availability of *several separate source models* and we introduce (ii) a *discriminative* approach based on Least Square Support Vector Machines (LS-SVM, [12]). Any new target class is learned through adaptation by imposing closeness between the target classifier and a linear combination of the source classifiers already learned on the j object sources. The weight assigned to each source knowledge is defined by solving a convex optimization problem which minimizes an upper bound of the leave-one-out error on the training set. This provides a principled solution for choosing from where to transfer and how much to rely on each known source. In practice, the proposed method (iii) autonomously tunes the transfer process depending on the *similarity* between the sources and the target tasks. We analyze in detail several properties of the described approach and perform an extensive experimental comparison with other existing transfer solutions, consistently showing the value of our algorithm.

The rest of the paper is organized as follows. Section 2 provides a short introduction to the goals, challenges and possible scenarios of knowledge transfer. Section 3 briefly reviews the literature. A detailed description of the notation and of the mathematical framework for our method follows in Section 4. Section 5 contains the formal definition of our knowledge transfer algorithm. Section 6 introduces an extension to the case of heterogeneous sources. Finally in Section 7 we present a thorough experimental evaluation, benchmarking against several other state of the art approaches. Section 8 concludes the paper with an overall discussion and pointing out possible avenues for future research.

2 Knowledge Transfer: Issues and Scenarios

The main assumption in theoretical models of learning, such as the standard PAC (Probably Approximately Correct [13]) model, is that training instances are drawn according to the same probability distribution as the unseen test examples. This hypothesis permits to estimate the generalization error and the uniform convergence theory [14] provides basic guarantees on the correctness of future decisions.

This ideal assumption is not always true in practical problems. It can happen that we have a lot of annotated data on a

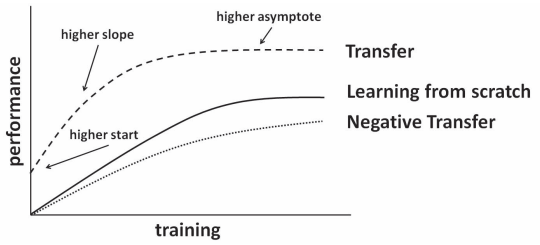

Fig. 2. Three ways in which transfer might improve the learning performance when the number of target training samples increases. Forcing the target learning process to rely on unrelated sources produces the negative transfer effect. (Figure reproduced and adapted from [16]).

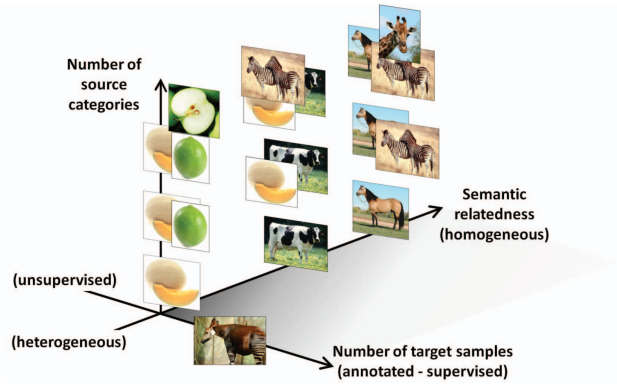

Fig. 3. A scheme of the possible transfer learning conditions in visual object categorization. The number of source sets can increase with different possible levels of relatedness with respect to the target category. The tasks are heterogeneous (homogeneous) if the samples are represented with different (the same) descriptors. The target task can be supervised with an increasing number of training samples or unsupervised when the target samples are not annotated.

source problem and the need to solve a different *target* problem with few labeled samples, where source and target present a distribution mismatch. In this case knowledge transfer (a.k.a transfer learning [15]) may decrease the effort of collecting new samples, while at the same time it may reduce the risk of overfitting by leveraging over the existing source knowledge to solve a target task. It is possible to define three measures by which transfer may improve the effectiveness of learning (see Figure 2):

(1) *Higher start*: the initial performance achievable on the target task is much better compared to that of an ignorant agent [16]. This is true even using only the source transferred knowledge, before any further learning on the target problem.

(2) *Higher slope*: this indicates a shorter amount of time needed to fully learn the target task, given the transferred knowledge, in comparison with learning from scratch [16].

(3) *Higher asymptote*: in the long run, the final performance level achievable over the target task may be higher compared to the final level without transfer [16].

How to get these advantages and up to which extent the transfer process can be useful depends on the specific scenario

at hand (object categorization, detection, reinforcement learning, etc.) and on the relation between source and target tasks. Apart from the different levels of semantic similarity, source and target might be represented with the same or with different descriptors which give rise respectively to a *homogeneous* or a *heterogeneous* transfer process. Moreover, a transfer learning problem can scale with respect to the number of annotated target samples and of possible source sets (see Figure 3). Indeed, to fully define any knowledge transfer method it is necessary to answer to three main questions.

(1) *What to transfer?* It refers to which knowledge can be transferred and to the form in which it is coded. In general terms, some knowledge might be specific for a task while some other knowledge might be common and shareable.

(2) *How to transfer?* This question is about the definition of a learning algorithm that can properly incorporate the source knowledge while building on the target samples.

(3) *When to transfer?* Finally, it is always necessary to evaluate the differences among the source and the target task and question whether the transfer is worthwhile or not.

In the following section we review the knowledge transfer literature, referring to how each of the proposed method addresses these challenging questions.

3 Related Work

The fundamental motivation for knowledge transfer in the field of artificial learning was discussed in a NIPS-95 workshop on *learning to learn* [17] which focused on the need for open ended learning systems that retain and reuse previously acquired knowledge. Since then, research on this topic has attracted more and more attention and several transfer approaches has been proposed in machine learning, natural language processing and computer vision.

3.1 What to Transfer

Depending on the problem to solve, the transferred knowledge can be in the form of instances, feature representation, or model parameters [15].

The main idea at the basis of *instance transfer* approaches is that, although no all the source data are useful, there are certain parts of them that can still be selected and considered together with the few available target labeled samples. In [18] Dai et al. proposed a boosting algorithm that uses both the source and the target data to solve visual object classification problems. Lim et al. [19] have shown that it is possible to borrow and transform examples across different visual object classes, demonstrating a performance improvement in detection problems.

Any *feature transfer* approach consists in learning a good representation for the target domain encoding in it some relevant knowledge extracted from the source. Bart and Ullman [20] proposed to perform feature adaptation using a single example of a novel class and showed a significant gain in classification performance. An alternative solution is to consider directly a metric learning approach [21] or more in general to exploit suitable kernels for the target data in SVM-based methods [22]. Moreover, the feature transfer approach has proven to be extremely useful in the deep learning framework for unsupervised classification tasks [23]. In this setting some recent work proposed also to represent object categories indirectly by their *attributes* [24]. An attribute is a high level semantic information (e.g. striped, furry) that is shared by multiple object categories and can be easily transferred as a descriptor.

Finally, a *parameter or model transfer* approach assumes that the source tasks and the target tasks share some parameters or prior distributions of the models. As already mentioned, Fei-Fei et al. [7] 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. Stark et al. [10] defined a technique to transfer a shape model across object classes. Yang et al. [25] presented a method to transfer the source information originally coded into an SVM model.

3.2 How to Transfer

A large variety of methods have been studied to integrate in different ways the source and target information: boosting approaches [18], [9], KNN [26], Markov logic [27], graphical models [28]. Most of the work has however been done in the generative probabilistic setting. Given the data, the target model makes predictions by combining them with the prior source distribution to produce a posterior distribution. A strong prior significantly affects these results serving as a natural way for Bayesian learning methods to transfer source knowledge. Some discriminative (maximum margin) methods are presented in [21] by learning a distance metric, and in [25] by exploiting a pre-learned SVM model. Also [11], [29] proposed to use a template learned previously for some object categories to regularize the training of a new target category.

3.3 When to Transfer

In real learning scenarios, the information acquired in the past is not always relevant for a new target problem. Rosenstain et al. [29] empirically showed that if two tasks are dissimilar, brute force transfer hurts the performance producing the so called *negative transfer* (see Figure 2). Ideally, a transfer method should be beneficial between appropriately related tasks while avoiding negative transfer when the tasks are not a good match. In practice, these goals are difficult to achieve simultaneously. Approaches that have safeguards against negative transfer often produce a smaller effect from positive transfer due to their caution. Conversely, approaches that transfer aggressively and produce large positive-transfer effects often have no protection against negative transfer.

It is possible to identify two main strategies to decide when to transfer. One consists in rejecting bad information, or at least making sure that its impact is minimized. This means choosing always *how much* to transfer, and disregard completely the source knowledge if not relevant for the target. A different strategy can be applied when there are more than one source task: in this condition the problem becomes choosing the best source. Transfer methods without much protection against negative transfer may still be effective in this scenario, as long as the best source task is at least a

decent match. Taylor et al. [30] proposed a transfer hierarchy, sorting the tasks by difficulty. Given a task ordering, it may be possible to locate the position of the target task in the hierarchy and select the most useful source set. In [31] the authors used conditional Kolmogorov complexity to measure relatedness between tasks and transfer the right amount of information.

Our work fits in this context. We propose a discriminative knowledge transfer method that relies on a set of models learned on the source categories (*what to transfer*) which are then used to regularize the target object model (*how to transfer*). The relatedness among the tasks is automatically evaluated (*when to transfer*) through a principled optimization problem without any need of hand tuned parameters, extra validation samples or a pre-defined ontology.

4 Mathematical Framework

We introduce here the formal notation and the necessary mathematical tools used in the rest of the paper. In the following we denote with small and capital bold letters respectively column vectors and matrices, e.g. $\boldsymbol{a} = [a_1, a_2, \ldots, a_N]^T \in \mathbb{R}^N$ and $\boldsymbol{A} \in \mathbb{R}^{M \times N}$ with A_{ji} corresponding to the (j, i) element. When only one subscripted index is present, it represents the column index, e.g., $\boldsymbol{A}_i$ is the i-th column of the matrix $\boldsymbol{A}$. Moreover we indicate with $\|\boldsymbol{a}\|_p := \left(\sum_{i=1}^N |\boldsymbol{a}_i|^p\right)^{1/p}$ the *p-norm* of a vector $\boldsymbol{a} \in \mathbb{R}^N$.

Let us assume $\boldsymbol{x}_i \in \mathcal{X}$ to be an input vector to a learning system and $y_i \in \mathcal{Y}$ its associated output. Given a set of data $D = \{\boldsymbol{x}_i, y_i\}_{i=1}^N$ drawn from an unknown probability distribution P, we want to find a function $f : \mathcal{X} \to \mathcal{Y}$ such that it determines the best corresponding y for any future sample $\boldsymbol{x}$. We consider $\mathcal{X} \subseteq \mathbb{R}^d$ and $\mathcal{Y} = \{-1, 1\}$.

The described learning process can be formalized as an optimization problem which aims at finding f in the hypothesis space of functions $\mathcal{H}$, which minimizes the structural risk [14]

$$\Omega(f) + C \sum_{i=1}^{N} \ell(f(\boldsymbol{x}_i), y_i) \ . \tag{1}$$

Here $\Omega(f)$ is a regularizer, which encodes some notion of smoothness for f, and guarantees good generalization performance avoiding overfitting. In the second term, ℓ is some convex non-negative loss function which assesses the quality of the function f on the instance and label pair $\{\boldsymbol{x}_i, y_i\}$. In practice it expresses the price we pay by predicting $f(\boldsymbol{x}_i)$ in place of y_i. The predictivity is a trade-off between fitting the training data and keeping the complexity of the solution low, controlled by the parameter $C > 0$.

4.1 Adaptive Regularization

We set $\mathcal{H}$ equal to space of all the linear models of the form

$$f(\boldsymbol{x}) = \boldsymbol{w}^\top \phi(\boldsymbol{x}) + b \ . \tag{2}$$

Here $\phi(x)$ is a feature mapping that maps the samples into a high, possible infinite dimensional space, where the dot product is expressed with a functional form $K(x, x') = \phi(x)^\top \phi(x')$ named kernel [32]. We also set the regularizer to be $\Omega(f) = \frac{1}{2}\|\boldsymbol{w}\|^2$, so that, regardless of the specific form of the loss function, the learning problem (1) becomes

$$\min_{\boldsymbol{w}} \frac{1}{2}\|\boldsymbol{w}\|^2 + C \sum_{i=1}^{N} \ell(\boldsymbol{w}^\top \phi(\boldsymbol{x}_i) + b, y_i) \ . \tag{3}$$

In this classical scheme for inductive learning, the knowledge eventually gained on the data $\hat{D} = \{\hat{\boldsymbol{x}}_i, \hat{y}_i\}_{i=1}^{\hat{N}}$ extracted from a distribution $\hat{P}$, different with respect to the target one P, is not taken into consideration. However, if $\hat{N} \gg N$ with a small number of available samples N (~ 10) and if the two distributions P, $\hat{P}$ are somehow related, the auxiliary knowledge can be helpful in guiding the learning process.

Let us suppose that the optimal $\hat{\boldsymbol{w}}$ has been already found by minimizing (3) for some source problem. When facing a new target task, we can always ask $\boldsymbol{w}$ to be close to the known $\hat{\boldsymbol{w}}$ by simply changing the regularization term [33] such that the learning problem results

$$\min_{\boldsymbol{w}} \frac{1}{2}\|\boldsymbol{w} - \hat{\boldsymbol{w}}\|^2 + C \sum_{i=1}^{N} \ell(\boldsymbol{w}^\top \phi(\boldsymbol{x}_i) + b, y_i) \ . \tag{4}$$

Thus, the optimization problem aims now at obtaining a vector $\boldsymbol{w}$ close to the source model $\hat{\boldsymbol{w}}$ by maximizing the projection of the first on the second. To properly scale the importance of this projection in the optimization problem, it is possible to add a weighting factor β such that the regularizer becomes $\|\boldsymbol{w} - \beta\hat{\boldsymbol{w}}\|^2$.

5 Multi Model Knowledge Transfer

Consider the following situation. We want to learn the target object class *okapi* from few examples, having already a model for the source categories *horse*, *zebra*, *melon* and *apple*. On the basis of the visual similarity, we can guess that the final model for okapi will be close to that of horse and zebra. Thus in the learning process we would like to transfer information from these two categories. We would expect the model obtained in this way to produce better recognition results with respect to (i) just considering horse *or* zebra as reference, and (ii) relying over all the source knowledge in a flat way, as melon and apple might induce negative transfer. This kind of reasoning motivates us to design a knowledge transfer algorithm able to find autonomously the best subset of known models from where to transfer, and to weight properly the relevant information.

Any transfer method, based on the adaptive regularization described in the previous section, answers the question *what to transfer* in terms of model parameters, by passing the known $\hat{\boldsymbol{w}}$ to the new target problem. However, previous work did not pay too much attention on *when* and *how much* to transfer. The discussed weight factor β in the regularizer is usually set equal to 1 with the hypothesis that the known models are useful and related to the target problem [25]. In other cases β is treated as a learning parameter, and is chosen by cross validation assuming the availability of extra target training samples [11]. Both these choices present some issues: the first case does not consider the danger of negative transfer

when only unrelated prior information is available, while in the second, the existence of extra data for cross validation is incoherent with the small sample scenario of transfer learning.

Here we study instead the case of multiple (J) available sources. We propose a learning method which relies over all of them and assigns to each a weight β_j for $j = 1, \ldots, J$. These values are automatically tuned on the basis of the few available target training data. We name our algorithm Multi Model Knowledge Transfer (*Multi-KT*) and we present its basic components in the following subsections.

5.1 Adaptive Least-Square Support Vector Machine

The first step to define our transfer learning algorithm consists in combining linearly the source models to have $\sum_{j=1}^{J} \beta_j \hat{\boldsymbol{w}}_j$ and using this as reference instead of the single source in (4). Moreover, we choose the weighted square loss $\ell(f(\boldsymbol{x}_i), y_i) = \zeta_i (f(\boldsymbol{x}_i) - y_i)^2$ [34], where the parameter ζ_i can be used to balance the contribution of positive and negative samples, taking into account that their proportion in the training set may be not representative of the operational class frequency.

The obtained optimization problem is:

$$\begin{aligned} \min_{\boldsymbol{w},b} \ & \frac{1}{2} \left\| \boldsymbol{w} - \sum_{j=1}^{J} \beta_j \hat{\boldsymbol{w}}_j \right\|^2 + \frac{C}{2} \sum_{i=1}^{N} \zeta_i \xi_i^2 \\ \text{s.t.} \quad & y_i = \boldsymbol{w}^\top \phi(\boldsymbol{x}_i) + b + \xi_i, \ \forall i = 1, \ldots, N, \end{aligned} \tag{5}$$

where we have introduced the slack variables ξ_i which measure the degree of misclassification on the data $\boldsymbol{x}_i$. Thus we obtain a new formulation for Least Square Support Vector Machine (LS-SVM [12]), that uses the adaptive regularizer introduced before. The corresponding Lagrangian $\mathcal{L}$ is

$$\frac{1}{2} \|\boldsymbol{w} - \sum_{j=1}^{J} \beta_j \hat{\boldsymbol{w}}_j\|^2 + \frac{C}{2} \sum_{i=1}^{N} \zeta_i \xi_i^2 - \sum_{i=1}^{N} a_i (\boldsymbol{w}^\top \phi(\boldsymbol{x}_i) + b + \xi_i - y_i) \ .$$

Here $\boldsymbol{a} \in \mathbb{R}^N$ is the vector of Lagrange multipliers and the optimality condition with respect to $\boldsymbol{w}$ is

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}} = \boldsymbol{0} \implies \boldsymbol{w} = \sum_{j=1}^{J} \beta_j \hat{\boldsymbol{w}}_j + \sum_{i=1}^{N} a_i \phi(\boldsymbol{x}_i) \ . \tag{6}$$

Thus, the adapted model is given by the weighted sum of the pre-trained source models $\hat{\boldsymbol{w}}_j$ and a linear combination of the target samples. Note that when all the β_j are 0 we recover the original LS-SVM formulation without any adaptation. Considering also the derivative of $\mathcal{L}$ with respect to ξ_i and a_i, we have respectively $a_i = C \zeta_i \xi_i$ and $\boldsymbol{w}^\top \phi(\boldsymbol{x}_i) + b + \xi_i - y_i = 0$. By combining them with (6) we find

$$\sum_{k=1}^{N} a_k \ \phi(\boldsymbol{x}_k)^\top \phi(\boldsymbol{x}_i) + b + \frac{a_i}{C \zeta_i} = y_i - \sum_{j=1}^{J} \beta_j \hat{\boldsymbol{w}}_j^\top \phi(\boldsymbol{x}_i) \ . \tag{7}$$

By denoting with $\boldsymbol{K}$ the kernel matrix, i.e. $\boldsymbol{K}_{ji} = K(\boldsymbol{x}_j, \boldsymbol{x}_i) = \phi(\boldsymbol{x}_j)^\top \phi(\boldsymbol{x}_i)$, the obtained system of linear equations can be written more concisely in matrix form as

$$\begin{bmatrix} \mathbf{K} + \frac{1}{C} \mathbf{Z} & \mathbf{1} \\ \mathbf{1}^\top & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{a} \\ b \end{bmatrix} = \begin{bmatrix} \boldsymbol{y} - \sum_{j=1}^{J} \beta_j \hat{\boldsymbol{y}}_j \\ 0 \end{bmatrix}, \tag{8}$$

where $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}_j$ are the vectors containing respectively the label of each training sample and the prediction of the previous model j, i.e. $\boldsymbol{y} = [y_1, \ldots, y_N]^\top$, $\hat{\boldsymbol{y}}_j = [\hat{\boldsymbol{w}}_j^\top \phi(\boldsymbol{x}_1), \ldots, \hat{\boldsymbol{w}}_j^\top \phi(\boldsymbol{x}_N)]^\top$. Moreover, $\mathbf{Z} = diag\{\zeta_1^{-1}, \zeta_2^{-1}, \ldots, \zeta_N^{-1}\}$ and to balance the contribution of differently labeled samples to the misfit term we set

$$\zeta_i = \begin{cases} \frac{N}{2N^+} & \text{if } y_i = +1 \\ \frac{N}{2N^-} & \text{if } y_i = -1 \ . \end{cases} \tag{9}$$

Here N^+ and N^- represent the number of positive and negative examples respectively.

Finally, the model parameters can be calculated simply by matrix inversion:

$$\begin{bmatrix} \boldsymbol{a} \\ b \end{bmatrix} = \boldsymbol{P} \begin{bmatrix} \boldsymbol{y} - \sum_{j=1}^{J} \beta_j \hat{\boldsymbol{y}}_j \\ 0 \end{bmatrix}, \tag{10}$$

where $\boldsymbol{P} = \boldsymbol{M}^{-1}$ and $\boldsymbol{M}$ is the first matrix on the left in (8). We underline that the pre-trained models $\hat{\boldsymbol{w}}_j$ can be obtained by any training algorithm, as long as it can be expressed as a weighted sum of kernel functions; the framework is therefore very general.

5.2 When and How Much to Transfer

Finding the optimal value for the elements of the weight vector $\boldsymbol{\beta}$ corresponds to ranking the prior knowledge sources and decide from where and how much to transfer. We propose to choose $\boldsymbol{\beta}$ in order to minimize the leave-one-out error, which is an almost unbiased estimator of the generalization error [34]. While in general computing the leave-one-out error is a very expensive procedure, we show that for (5) it can be obtained with a closed-formula, using quantities that are already computed during the training phase.

Let us denote by $\tilde{y}_i$, $i = 1, \ldots, N$, the prediction on sample i when it is removed from the training set. LS-SVM in its original formulation makes it possible to write these leave-one-out predictions in closed form and with a negligible additional computational cost [34]. We show below that the same property extends to the modified problem in (5).

Proposition 1: Let $[\boldsymbol{a}'^\top, b']^\top = \boldsymbol{P}[\boldsymbol{y}^\top, 0]^\top$ and $[\boldsymbol{a}_j''^\top, b_j'']^\top = \boldsymbol{P}[\hat{\boldsymbol{y}}_j^\top, 0]^\top$ with $\boldsymbol{a} = \boldsymbol{a}' - \sum_{j=1}^{J} \beta_j \boldsymbol{a}_j''$. If we indicate with $\boldsymbol{A}''$ the matrix containing the vector $\boldsymbol{a}_j''^\top$ in the j-th row, the prediction $\tilde{y}_i$, obtained on sample i when it is removed from the training set, is equal to

$$y_i - \frac{a_i'}{P_{ii}} + \frac{\boldsymbol{\beta}^\top \boldsymbol{A}_i''}{P_{ii}} \ , \tag{11}$$

where $\boldsymbol{\beta} \in \mathbb{R}^J$ is a vector containing all the values β_j.

Proof of Proposition 1: We start from

$$\boldsymbol{M} \begin{bmatrix} \boldsymbol{a} \\ b \end{bmatrix} = \begin{bmatrix} \boldsymbol{y} - \sum_{j=1}^{J} \beta_j \hat{\boldsymbol{y}}_j \\ 0 \end{bmatrix}, \tag{12}$$

and we decompose $\boldsymbol{M}$ into block representation isolating the first row and column as follows:

$$\boldsymbol{M} = \begin{bmatrix} \mathbf{K} + \frac{1}{C} \mathbf{Z} & \mathbf{1} \\ \mathbf{1}^\top & 0 \end{bmatrix} = \begin{bmatrix} m_{11} & \mathbf{m}_1^\top \\ \mathbf{m}_1 & \mathbf{M}_{(-1)} \end{bmatrix} \ .$$

Let $\boldsymbol{a}_{(-i)}$ and $b_{(-i)}$ represent the model parameters during the i-th iteration of the leave-one-out cross validation procedure. In the first iteration, where the first training sample is excluded we have

$$\left[\begin{array}{c} \boldsymbol{a}_{(-1)} \\ b_{(-1)} \end{array} \right] = \mathbf{P}_{(-1)}(\boldsymbol{y}_{(-1)} - \sum_{j=1}^{J} \beta_j \hat{\boldsymbol{y}}_{j(-1)}),$$

where $\mathbf{P}_{(-1)} = \mathbf{M}_{(-1)}^{-1}$, $\boldsymbol{y}_{(-1)} = [y_2, \ldots, y_N, 0]^\top$ and $\hat{\boldsymbol{y}}_{j(-1)} = [\hat{\boldsymbol{w}}_j^\top \phi(\boldsymbol{x}_2), \ldots, \hat{\boldsymbol{w}}_j^\top \phi(\boldsymbol{x}_N), 0]^\top$. The leave-one-out prediction for the first training sample is then given by

$$\begin{aligned} \tilde{y}_1 &= \mathbf{m}_1^\top \left[\begin{array}{c} \boldsymbol{a}_{(-1)} \\ b_{(-1)} \end{array} \right] + \sum_{j=1}^{J} \beta_j \hat{\boldsymbol{w}}_j^\top \phi(\boldsymbol{x}_1) \\ &= \mathbf{m}_1^\top \mathbf{P}_{(-1)} \left(\boldsymbol{y}_{(-1)} - \sum_{j=1}^{J} \beta_j \hat{\boldsymbol{y}}_{j(-1)} \right) + \sum_{j=1}^{J} \beta_j \hat{\boldsymbol{w}}_j^\top \phi(\boldsymbol{x}_1) \ . \end{aligned}$$

Considering the last N equations in the system in (12), it is clear that $[\mathbf{m}_1 \ \mathbf{M}_{(-1)}][\boldsymbol{a}^\top, b]^\top = (\boldsymbol{y}_{(-1)} - \sum_{j=1}^{J} \beta_j \hat{\boldsymbol{y}}_{j(-1)})$, and so

$$\begin{aligned} \tilde{y}_1 &= \mathbf{m}_1^\top \mathbf{P}_{(-1)} [\mathbf{m}_1 \mathbf{M}_{(-1)}][a_1, \ldots, a_N, b]^\top + \sum_{j=1}^{J} \beta_j \hat{\boldsymbol{w}}_j^\top \phi(\boldsymbol{x}_1) \\ &= \mathbf{m}_1^\top \mathbf{P}_{(-1)} \mathbf{m}_1 a_1 + \mathbf{m}_1^\top [a_2, \ldots, a_N, b]^\top + \sum_{j=1}^{J} \beta_j \hat{\boldsymbol{w}}_j^\top \phi(\boldsymbol{x}_1) \ . \end{aligned}$$

In (12) the first equation of the system is $y_1 - \sum_{j=1}^{J} \beta_j \hat{\boldsymbol{w}}_j^\top \phi(\boldsymbol{x}_1) = m_{11} a_1 + \mathbf{m}_1^\top [a_2, \ldots, a_N, b]^\top$, and we have $\tilde{y}_1 = y_1 - a_1(m_{11} - \mathbf{m}_1^\top \mathbf{P}_{(-1)} \mathbf{m}_1)$. Finally, by using $\mathbf{P} = \mathbf{M}^{-1}$ and the block matrix inversion lemma we get

$$\mathbf{P} = \left[\begin{array}{cc} \mu^{-1} & -\mu^{-1} \mathbf{m}_1 \mathbf{P}_{(-1)} \\ \mathbf{P}_{(-1)} + \mu^{-1} \mathbf{P}_{(-1)} \mathbf{m}_1^\top \mathbf{m}_1 \mathbf{P}_{(-1)} & -\mu^{-1} \mathbf{P}_{(-1)} \mathbf{m}_1^\top \end{array} \right],$$

where $\mu = m_{11} - \mathbf{m}_1^\top \mathbf{P}_{(-1)} \mathbf{m}_1$. By noting that the system of linear equations (12) is insensitive to permutations of the ordering of the equations and of the unknowns, we have

$$\tilde{y}_i = y_i - \frac{a_i}{P_{ii}} \ .$$

By defining $[\boldsymbol{a}'^\top, b']^\top = \boldsymbol{P}[\boldsymbol{y}^\top, 0]^\top$, $[\boldsymbol{a}_j''^\top, b_j'']^\top = \boldsymbol{P}[\hat{\boldsymbol{y}}_j^\top, 0]^\top$, and $\boldsymbol{a} = \boldsymbol{a}' - \sum_{j=1}^{J} \beta_j \boldsymbol{a}_j''$, we get

$$\tilde{y}_i = y_i - \frac{a_i'}{P_{ii}} + \sum_{j=1}^{J} \beta_j \frac{A_{ji}''}{P_{ii}} = y_i - \frac{a_i'}{P_{ii}} + \frac{\boldsymbol{\beta}^\top \boldsymbol{A}_i''}{P_{ii}} \ ,$$

where $\boldsymbol{\beta} \in \mathbb{R}^J$ is a vector containing all the values β_j and $\boldsymbol{A}''$ is the matrix containing the vector $\boldsymbol{a}_j''^\top$ in the j-th row. ■

Notice that in (11) $\boldsymbol{a}$ depends linearly on $\boldsymbol{\beta}$, thus it is straightforward to obtain the learning model once all the β_j have been chosen. The best values for β_j are those producing positive values for $y_i \tilde{y}_i$, for each i. However, focusing only on the sign of those quantities would result in a non-convex formulation with many local minima. We propose instead the following loss function, similar to the hinge loss

$$\ell(\tilde{y}_i, y_i) = \zeta_i |1 - y_i \tilde{y}_i|_+ = \zeta_i \left| y_i \frac{a_i' - \boldsymbol{\beta}^\top \boldsymbol{A}_i''}{P_{ii}} \right|_+ , \qquad (13)$$

Algorithm 1 Projected Sub-gradient Descent Algorithm

Input: Set $\boldsymbol{a}'$, $\boldsymbol{a}_j''$, and $\boldsymbol{A}''$ according to Proposition 1
Initialize: $\beta \leftarrow 0$ and $t \leftarrow 1$
repeat
  $\tilde{y}_i \leftarrow y_i - \frac{\boldsymbol{a}_i'}{P_{ii}} + \sum_{j=1}^{J} \beta_j \frac{A_{ji}''}{P_{ii}} \quad \forall\ i = 1, \ldots, N$
  $d_i \leftarrow \mathbf{1}\{y_i \tilde{y}_i > 0\} \ , \quad \forall\ i = 1, \ldots, N$
  $\beta_j \leftarrow \beta_j - \frac{1}{\sqrt{t}} \sum_{i=1}^{N} d_i y_i \frac{a_{ji}''}{P_{ii}} \ , \quad \forall\ j = 1, \ldots, J$
  if $\|\boldsymbol{\beta}\|_2 > 1$ **then**
    $\boldsymbol{\beta} \leftarrow \boldsymbol{\beta} / \|\boldsymbol{\beta}\|_2$
  end if
  $\beta_j \leftarrow \max(\beta_j,\ 0), \quad \forall\ j = 1, \ldots, J$
  $t \leftarrow t + 1$
until convergence
Output: $\boldsymbol{\beta}$

where $|x|_+ = \max\{0, x\}$. It is a convex upper bound to the leave-one-out misclassification loss and it favors solutions in which $\tilde{y}_i$ has an absolute value equal or bigger than 1, and the same sign of y_i. The weights ζ_i are set again according to (9). Finally, the objective function is

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{N} \ell(y_i, \tilde{y}_i) \quad \text{subject to} \quad \|\boldsymbol{\beta}\|_p \leq 1 \ , \ \ \beta_j \geq 0 \ , \qquad (14)$$

where we added some constraint on the $\boldsymbol{\beta}$ vector as a form of regularization. They may be helpful to avoid overfitting problems when the number of known models J is large compared to the number of training samples N. Depending on the value of p, how the target learning model leverages over the source models changes:

$\mathbf{p = 2}$, $\mathbf{L_2}$ norm constraint. This is the well known Euclidean norm indicated by $\|\cdot\|_2$ or simply $\|\cdot\|$. A regularization based on it generally induces numerical stability. The optimization process can be implemented by using a projected sub-gradient descent algorithm, where at each iteration $\boldsymbol{\beta}$ is projected onto the L_2-sphere $\|\boldsymbol{\beta}\| \leq 1$, and then on the positive orthant. The pseudo-code is in Algorithm 1.

$\mathbf{p = 1}$, $\mathbf{L_1}$ norm constraint. This is simply the sum of the absolute values of the vector elements. This constraint induces a sparse solution, i.e. only some vector elements remain different from zero. Applied on prior knowledge regularization, the condition $\|\boldsymbol{\beta}\|_1 \leq 1$ can be easily implemented (e.g. by using the algorithm in [35]), and it gives rise to an automatic source selection technique.

$\mathbf{p = \infty}$, $\mathbf{L_\infty}$ norm constraint. This norm is defined as

$$\|\boldsymbol{x}\|_\infty := \max\{|x_1|, \ldots, |x_d|\} \ . \qquad (15)$$

In practice, by using $\|\boldsymbol{\beta}\|_\infty \leq 1$ as constraint, all the vector elements will have an absolute value not bigger than one. In this case the projection consists of a simple truncation.

The second condition in (14) limits the weights of the source knowledge models to be positive. In fact, in the object category detection problem, all the considered source and target sets have the background category as negative class, thus it is reasonable to expect that the angle between $\boldsymbol{w}$ and any $\hat{\boldsymbol{w}}_j$ is always acute.

5.3 Computational Complexity

From a computational point of view the runtime of the Multi-KT algorithm is $\mathcal{O}(N^3+JN^2)$, with N the number of training samples, and J the number of source models. The first term is related to the evaluation of the matrix $\boldsymbol{P}$, which must anyway occur while training, while the second term is the computational complexity of (11), which results negligible, if compared to the complexity of training. Thus, we match the complexity of a plain SVM, which in the worst case is known to be $\mathcal{O}(N^3)$ [36]. The computational complexity of each step of the projected sub-gradient descent to optimize (13) is $\mathcal{O}(JN)$, and it results extremely fast (our MATLAB implementation takes just half a second with $N = 12$ and $J = 3$ on current hardware).

6 Heterogoenous Knowledge Transfer

The proposed Multi-KT transfer method is based on the idea of pushing the target model $\boldsymbol{w}$ close to a linear combination of prior known sources $\sum_{j=1}^{J} \beta_j \hat{\boldsymbol{w}}_j$. However, to impose this closeness, all the vectors should live in a single space. This means that the kernel used in learning over all the sources and on the new target must be the same. This is quite a strict condition because it does not give the freedom to build the source knowledge over heterogeneous feature descriptors, and imposes a unique metric to evaluate the sample similarity. In this section we show how to overcome this limit by enlarging the space in which the learning function lies, by a multi-kernel approach. We call this variant MultiK-KT.

Assume to have $j = 1, \ldots, J$ mappings, each to a different space, where the image of a vector $\boldsymbol{x}$ is $\phi_j(\boldsymbol{x})$. We can always compose all of them orthogonally (see Figure 4) obtaining the mapping to the final space by concatenation: $\phi'(\boldsymbol{x}) = [\phi_1(\boldsymbol{x}), \phi_2(\boldsymbol{x}), \ldots, \phi_J(\boldsymbol{x})]^\top$. The dot product $\phi'(\boldsymbol{x})^\top \phi'(\boldsymbol{z})$ in this new space is equal to the kernel K', defined as

$$K'(\boldsymbol{x}, \boldsymbol{z}) = \sum_{j=1}^{J} \phi_j(\boldsymbol{x})^\top \phi_j(\boldsymbol{z}) = \sum_{j=1}^{J} K_j(\boldsymbol{x}, \boldsymbol{z}), \qquad (16)$$

where $K_j(\boldsymbol{x}, \boldsymbol{z})$ is the kernel function in the j-th space.

Now let us consider the transfer learning problem with $j = 1, \ldots, J$ source object classes and suppose to solve the binary classification object-vs-background for each of them in a specific space, i.e. choosing different feature descriptors, different kernel functions, and/or different kernel parameters. The obtained model vectors are

$$\hat{\boldsymbol{w}}_j = \sum_{i=1}^{\hat{N}_j} \alpha_i^j \phi_j(\boldsymbol{x}_i) \ .$$

These solutions can always be mapped in the composed new space using zero padding. In fact, $\phi_j(\boldsymbol{x}) \rightarrow \phi'_j(\boldsymbol{x}) = [0, \ldots, \phi_j(\boldsymbol{x}), \ldots, 0]^\top$, and we have

$$\begin{aligned} \hat{\boldsymbol{w}}_j \rightarrow \hat{\boldsymbol{w}}'_j &= [0, \ldots, \hat{\boldsymbol{w}}_j, \ldots, 0]^\top \\ &= [0, \ldots, \sum_{i=1}^{\hat{N}_j} \alpha_i^j \phi_j(\boldsymbol{x}_i), \ldots, 0]^\top \ . \end{aligned}$$

Hence, in the new space, a vector obtained as linear combination of all the known models results:

$$\begin{aligned} \sum_{j=1}^{J} \beta_j \hat{\boldsymbol{w}}'_j &= [\beta_1 \hat{\boldsymbol{w}}_1, \ldots, \beta_J \hat{\boldsymbol{w}}_J]^\top \\ &= [\beta_1 \sum_{i=1}^{\hat{N}_1} \alpha_i^1 \phi_1(\boldsymbol{x}_i), \ldots, \beta_J \sum_{i=1}^{\hat{N}_J} \alpha_i^J \phi_J(\boldsymbol{x}_i)]^\top \ . \end{aligned}$$

By supposing that the target problem lives in the new composed space, we can apply our Multi-KT algorithm there. Hence the original optimization problem in (5) becomes

$$\min_{\boldsymbol{w}', b} \ \frac{1}{2} \left\| \boldsymbol{w}' - \sum_{j=1}^{J} \beta_j \hat{\boldsymbol{w}}'_j \right\|^2 + \frac{C}{2} \sum_{i=1}^{N} \zeta_i (y_i - \boldsymbol{w}'^\top \phi'(\boldsymbol{x}_i) - b)^2 \ .$$

The solving procedure is the same described in Section 5.1 and the optimal solution is:

$$\boldsymbol{w}' = \sum_{j=1}^{J} \beta_j \hat{\boldsymbol{w}}'_j + \sum_{i=1}^{N} a_i \phi'(\boldsymbol{x}_i) \ .$$

When we use it for classification we get

$$\begin{aligned} \boldsymbol{w}'^\top \phi'(\boldsymbol{z}) &= \sum_{j=1}^{J} \beta_j \hat{\boldsymbol{w}}_j'^\top \phi'(\boldsymbol{z}) + \sum_{i=1}^{N} a_i \phi'(\boldsymbol{x}_i)^\top \phi'(\boldsymbol{z}) \\ &= \sum_{j=1}^{J} \beta_j \hat{\boldsymbol{w}}_j^\top \phi_j(\boldsymbol{z}) + \sum_{i=1}^{N} a_i \left(\sum_{j=1}^{J} \phi_j(\boldsymbol{x}_i)^\top \phi_j(\boldsymbol{z}) \right) \ , \end{aligned}$$

that is exactly the same that would be obtained from (6) using $K'(\boldsymbol{x}, \boldsymbol{z})$ as kernel. Even the original procedure to choose the best $\boldsymbol{\beta}$ can be easily enlarged to the case of linearly combined orthogonal spaces. The vector $\hat{\boldsymbol{y}}'_j$ containing the predictions of the $j-$th known model is:

$$\begin{aligned} \hat{\boldsymbol{y}}'_j &= [\hat{\boldsymbol{w}}_j'^\top \phi'(\boldsymbol{x}_1), \ldots, \hat{\boldsymbol{w}}_j'^\top \phi'(\boldsymbol{x}_N))] \\ &= [\hat{\boldsymbol{w}}_j^\top \phi_j(\boldsymbol{x}_1), \ldots, \hat{\boldsymbol{w}}_j^\top \phi_j(\boldsymbol{x}_N))] = \hat{\boldsymbol{y}}_j \ . \end{aligned}$$

This indicates that MultiK-KT is formally equivalent to the original Multi-KT with the kernel chosen as in (16). As a consequence the computational complexity of MultiK-KT is again $\mathcal{O}(N^3 + JN^2)$ (see Section 5.3).

7 Experiments

In this section we show empirically the effectiveness of our transfer algorithm[1] on three datasets: Caltech-256 [37], Animals with Attributes (AwA) [24] and IRMA [38].

The *Caltech-256* contains images of 256 object classes plus a clutter category that can be used as negative class in object-vs-background problems. The objects are organized in a hierarchical ontology that makes it easy to identify the related and unrelated categories. We downloaded[2] the pre-computed features of [39] and we selected four different image descriptors: PHOG Shape Descriptors [40], SIFT Appearance Descriptors [41], Region Covariance [42], and Local Binary Patterns [43]. They were all computed in a spatial pyramid [44] and we considered just the first level (i.e. information extracted from the whole image).

The *AwA* dataset contains 50 animal classes and it has been released with several pre-extracted feature sets for each

1. We implemented it in MATLAB, the code is available online http://www.idiap.ch/~ttommasi/source_code_CVPR10.html

2. http://files.is.tue.mpg.de/pgehler/projects/iccv09/

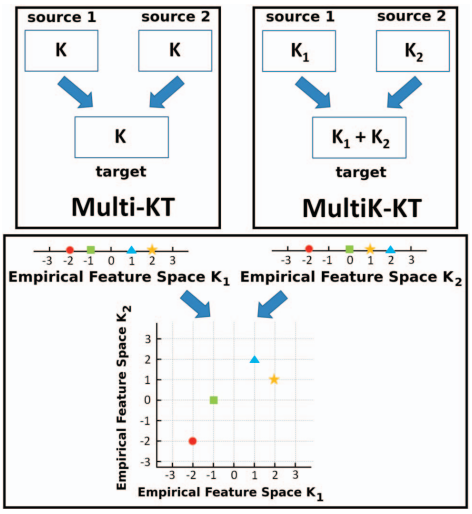

Fig. 4. For Multi-KT the source and target models must live in the same space identified by the kernel K. For MultiK-KT all the sources can be defined independently in their own space and the target solution lives in the space obtained by orthogonal combination. We show also a geometrical interpretation of the kernel combination.

image[3]. From the full set of categories we extracted the six sea mammals (killer whale, blue whale, humpback whale, seal, walrus and dolphin) and used them to define the background class. We used three of the precomputed descriptors for our experiments: color histogram, PHOG and SIFT.

The *IRMA* database is a collection of x-ray images presenting a large number of classes defined according to a four-axis hierarchical code [45]. We decided to work on the 2008 IRMA database version [38], just considering the third axis of the code which identifies the depicted body part. A total of 23 classes with more than 100 images were selected from various sub-levels of the third axis, 3 of them were used to define the background class. Following [46], we used as features the global pixel-based and local SIFT-based descriptors.

We performed all the experiments in a leave-one-class-out approach, that is considering in turn each class as target and all the others as sources. The number of negative training samples is kept fixed while the number of positive training samples increases in subsequent steps till reaching the same amount of the negative set. The samples are extracted randomly 10 times for an equal number of experimental runs. Each prior knowledge model is built with standard LS-SVM. We use the Gaussian kernel both on the source and on the target for all the experiments $K(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\gamma\|\boldsymbol{x}-\boldsymbol{x}'\|^2)$. To integrate multiple (F) features we calculate one kernel for each of them and we use the average kernel $K(\boldsymbol{x}, \boldsymbol{x}') = 1/F \sum_{f=1}^{F} K_f(\boldsymbol{x}, \boldsymbol{x}')$. All the transfer results are benchmarked against *no transfer*: this corresponds to learning from scratch with weighted-LS-SVM , i.e. solving the optimization problem in (5) with $\boldsymbol{\beta} = \mathbf{0}$.

Regarding the parameters, a unique common value for γ was chosen for all the kernels by cross validation on the source sets. In particular, we trained a model for each class in the source set and we used it to classify on the remaining $J-1$ source classes. Finally, we selected the γ value producing on average the best recognition rate. The value of C is instead determined as the one producing the best result when learning from scratch. There is no guarantee that the obtained C value is the best for the transfer approach; still in this way we compare against the best performance that can be obtained by learning only on the available training samples, without exploiting the source knowledge. We used this setup for all the experiments; specific differences are otherwise mentioned.

7.1 Setting the Constraints

To fully define the Multi-KT algorithm it is necessary to choose the p value in the constraint of (14). We evaluate empirically three cases with $p = 1, 2, \infty$ and we compare the obtained results over three groups of data that differ in the level of relatedness among source and target knowledge. Specifically, we extracted 6 unrelated classes (harp, microwave, fire-truck, cowboy-hat, 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 Caltech-256. We refer to a class as the combination of 80 object and 80 background images. For each class used as target, we extracted 20 training and 100 testing samples with half positive and half negative data.

The results in Figure 5 (top line) show the clear gain obtained by Multi-KT with respect to no transfer[4]. The advantage is maximum in case of related classes (the difference between Multi-KT L_2 and no transfer is 39% in recognition rate for 1 positive sample), it is just a little bit smaller for mixed classes (34%) and drops more in case of sources unrelated to the target task (29%). However, regardless of the relatedness level, the choice of the constraint on $\boldsymbol{\beta}$ does not produce significantly different results, apart for a slightly lower performance of the L_1 case with respect to the others. Hence, in the following we will always use the L_2 norm constraint.

7.2 Transfer Weights and Semantic Similarity

The Multi-KT algorithm defines automatically the relevance of each source model to the current target task. We analyze here the $\boldsymbol{\beta}$ vector obtained as a byproduct of the transfer process, to verify if its elements have a correspondence with the real visual and semantic relation among the tasks.

We start from the results obtained in the previous section with the L_2 norm constraint and we consider the intermediate training step with 5 positive samples. We average the $\boldsymbol{\beta}$ vectors obtained over the 10 runs defining a matrix of weights with one row for each class used as target. By simple algebra we can transform it to a fully symmetric matrix containing measures of class dissimilarities evaluated as $(1-\beta_j)$ and apply multidimensional scaling on it [48], with two dimensions. We obtain plots where each point represents a class, and the distance among the points is proportional to the input dissimilarities.

3. http://attributes.kyb.tuebingen.mpg.de/

4. Using SVM for learning from scratch produces slightly better results than LS-SVM. However in all our experiments this difference is not significant and it does not change the conclusions on the proposed transfer approach. We use the sign test [47] for our statistical evaluations.

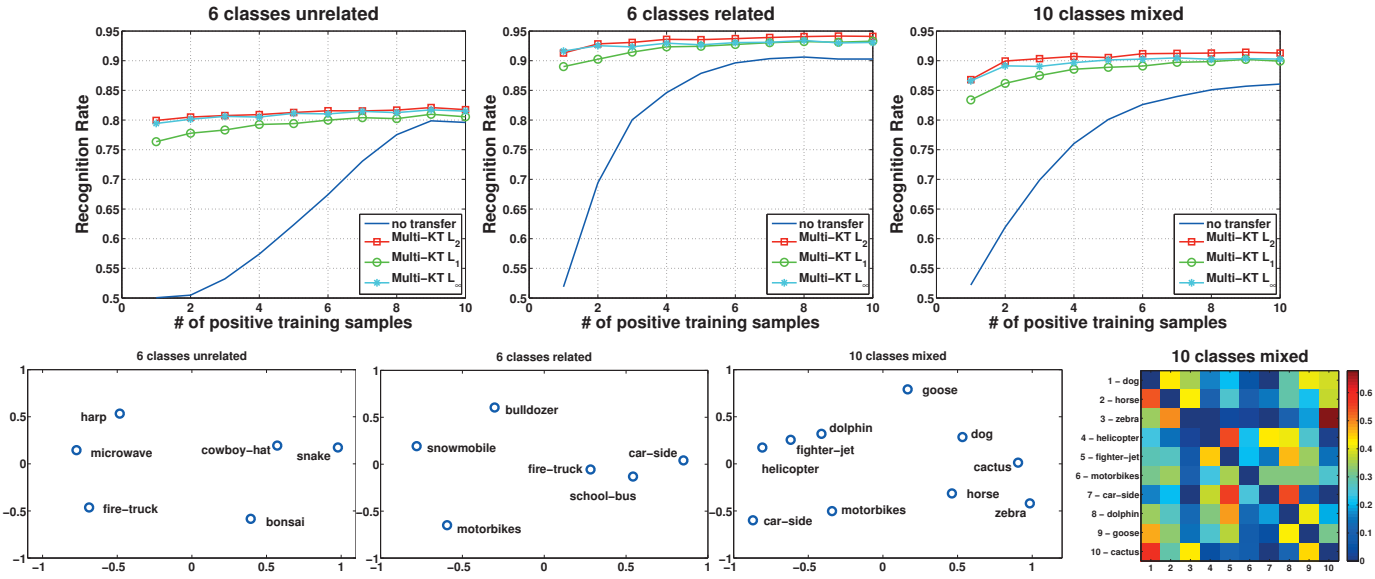

Fig. 5. Top line: Performance of the Multi-KT method with various settings for the constraint on the source knowledge weights. The results correspond to average recognition rate over the categories, considering each class out experiment repeated ten times. Bottom line: output of the bidimensional scaling applied on the $\boldsymbol{\beta}$ vector values. For 10 mixed classes we also show the assigned weigths with a heat map where each row corresponds to a target class and each column to a source class.

Figure 5 (bottom line) shows the obtained results. It can be seen that in the case of unrelated classes the corresponding points tend to be far from each other. On the other hand, among the related classes extracted from the general category *motorized-ground-vehicles*, the *four wheels* vehicles form a cluster, leaving aside motorbikes (two wheels), snowmobile (skis) and bulldozer (tracks). Finally, among the mixed classes, helicopter and fighter-jet appear close to each other and to dolphin. Probably this is due to the shape appearance of these object classes and to the common uniformity of the sky and water background. Moreover, all the four legged animals (zebra, horse and dog) appear on the right side of the plot, while the vehicles (car-side and motorbikes) are in the left bottom corner. The heat map of the $\boldsymbol{\beta}$ weights also shows that Multi-KT does not leverage over the source models in a flat way, but chooses properly which part of the available knowledge can be reused.

Globally all the results indicate that the $\boldsymbol{\beta}$ vectors actually contain meaningful values in terms of semantic relation between the object classes.

7.3 Comparison and Evaluation

Here we evaluate our Multi-KT algorithm in comparison with several state of the art transfer learning approaches. We briefly review them before discussing the experimental results.

Single Source. Most of the existing knowledge transfer methods suppose the availability of a single source knowledge. Among the approaches listed below, the first two are based on transferring model parameters as our Multi-KT, while the last one is an instance transfer technique and exploits directly the source samples.

Adaptive SVM (A-SVM). This method has been originally presented in [25] and is based on substituting the usual regularizer of the SVM formulation with the adaptive version

$$\min_{\boldsymbol{w}} \|\boldsymbol{w} - \beta\hat{\boldsymbol{w}}\|^2 + C\sum_{i=1}^{N} \ell^H(\boldsymbol{w}^\top \phi(\boldsymbol{x}_i), y_i) \ . \qquad (17)$$

Projective Model Transfer SVM (PMT-SVM). Maximizing the projection of $\boldsymbol{w}$ onto $\hat{\boldsymbol{w}}$ corresponds also to minimizing the projection of $\boldsymbol{w}$ onto the source separating hyperplane (orthogonal to $\hat{\boldsymbol{w}}$). Following this idea the objective function of PMT-SVM [11] is

$$\min_{\boldsymbol{w}} \ \|\boldsymbol{w}\|^2 + \beta\|\mathbf{R}\boldsymbol{w}\|^2 + C\sum_{i=1}^{N} \ell^H(\boldsymbol{w}^\top \phi(\boldsymbol{x}_i), y_i)$$
$$\text{s. t.} \quad \boldsymbol{w}^\top \hat{\boldsymbol{w}} \geq 0, \qquad (18)$$

here $\mathbf{R}$ is the projection matrix and $\|\mathbf{R}\boldsymbol{w}\|^2 = \|\boldsymbol{w}\|^2 \sin^2\theta$, where θ is the angle between $\boldsymbol{w}$ and $\hat{\boldsymbol{w}}$.

TrAdaBoost: boosting for Transfer Learning. This extension to the Adaboost learning framework was proposed in [18]. It is based on mechanism which, starting from the combination of source and target samples, iteratively decreases the weights of the source data in order to weaken their impact on the learning process.

Experiments. We benchmark here our Multi-KT algorithm against the described A-SVM, PMT-SVM and TrAdaBoost. Since these baseline methods were defined in the hypothesis of a single available source set, we considered two cases: a pair of unrelated and a pair of related classes. Both the pairs were extracted from Caltech-256 and each of the classes is considered in turn as target while the other represents the source task.

We used the MATLAB code of PMT-SVM provided by its authors, together with their implementation of A-SVM[5] slightly modifying them to introduce the weights ζ_i for $i =$

5. http://www.robots.ox.ac.uk/~vgg/software/tabularasa/

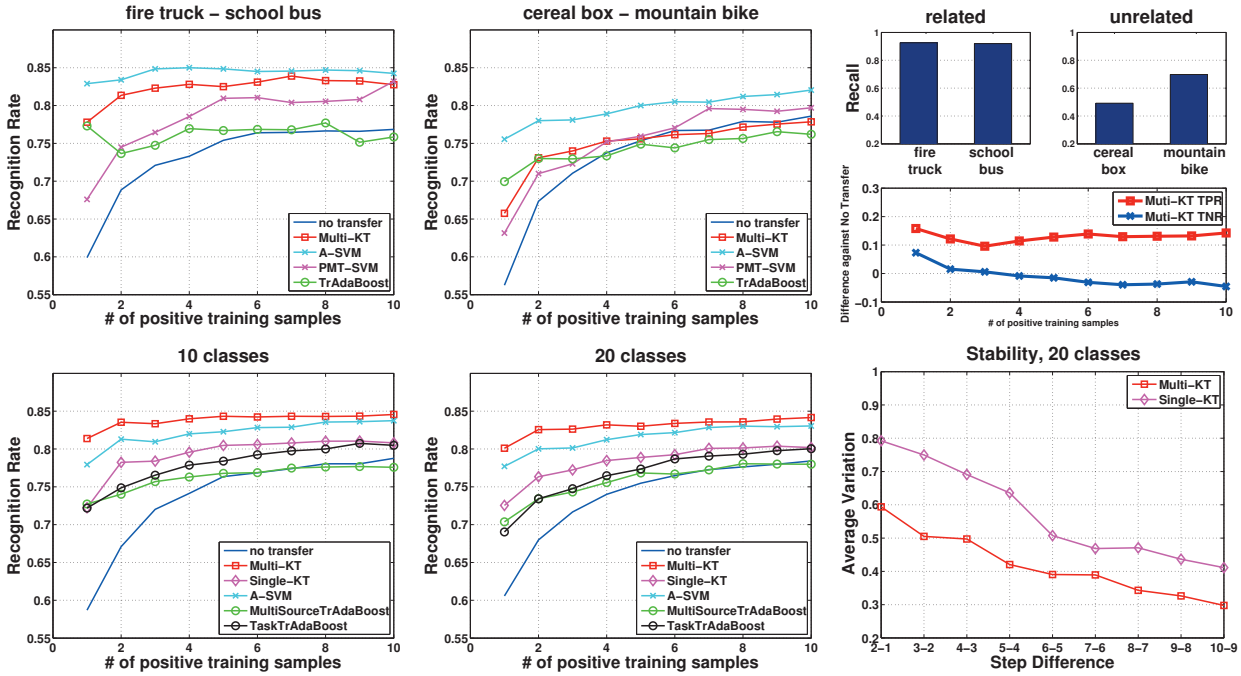

Fig. 6. Left and middle columns: recognition rate as a function of the number of positive training samples. In each experiment we consider in turn one of the classes as target and the others as source, on ten random training sets. The final results are obtained as average over all the runs. Top Right: (up) the histogram bars represent the recall produced by the source model (indicated on the x-axis) when used to classify on the target class; (down) we compute separately the true positive and true negative recognition rate of Multi-KT and we show the value of their difference with no transfer in the related case. Bottom Right: average norm of the difference between two β vectors obtained for a pair of subsequent training steps.

$1, \ldots, N$ in the corresponding loss function, so to have a fair comparison with our Multi-KT. The original formulation considered the linear kernel, thus we chose $K(\boldsymbol{x}, \boldsymbol{z}) = \boldsymbol{x}^\top \boldsymbol{z}$ for all the experiments together with the SIFT feature descriptors. In [11] the β value is defined by cross validation on extra validation target samples. Here we decided to simply tune it on the test set, showing the best result that could be obtained. The same approach was adopted to choose the number of boosting iterations for TrAdaBoost.

The results are shown in Figure 6 (top line). In the related (left plot) case all the transfer learning methods show better performance than learning from scratch with different extent. The results of our Multi-KT are significantly better than those of no transfer and PMT-SVM ($p \leq 0.01$). Only for 10 positive training samples PMT-SVM and Multi-KT produce comparable results. Multi-KT also outperforms TrAdaBoost for all the training steps ($p \leq 0.01$) except the first one, where they are statistically equivalent. Finally, the difference between Multi-KT and A-SVM is not significant: since the β parameter for A-SVM is tuned on the test set, this indicates that Multi-KT is autonomously able to optimally weight the source knowledge. The bias of A-SVM towards the best possible recognition rate is evident in the case of unrelated classes (middle plot) where it is the only method to outperform no transfer along all the steps. The other knowledge transfer approaches show better results than no transfer only for less than three positive training samples ($p \leq 0.05$), becoming then statistically equivalent to learning from scratch.

The histogram bars on the right in Figure 6 (top right - up) show the recall produced by each source model when used directly to classify on the target task. This indicates how good is the source in recognizing the target object without adaptation and it is clearly lower for unrelated than for related classes. For a deeper understanding of the method we also calculated separately the true positive and true negative recognition rate of Multi-KT in the case of related classes. We show the value of their difference with no transfer in Figure 6 (top right - down). From the plot we can conclude that the main advantage in transferring is in fact due to the relation between the source and target positive classes rather than to the joint background class.

Multiple Sources. When more than one source set is available, there are three main strategies that a transfer learning method can consider. Two extreme solutions consist in either selecting only one source, evaluated as the best for the target problem, or averaging over all of them supposing that they are all equally useful. The third strategy considers the intermediate case where only some of the source sets are helpful for the target task and consists in selecting them by assigning to each a proper weight. To our knowledge, only our Multi-KT method is based on the third selective technique.

MultiSourceTrAdaBoost: boosting by transferring samples. An extension to the TrAdaBoost approach in the case of multiple available sources has been presented in [9]. The method *MultiSourceTrAdaBoost* considers one source set at the time, combining it with the target set and defining a candidate

weak classifier. The final classifier is then chosen as the one producing the smallest training target classification error by automatically selecting the corresponding best source.

TaskTrAdaBoost: boosting by transferring models. This is a parameter transfer approach consisting of two steps. Phase I deploys traditional AdaBoost separately on each source task to get a collection of candidate weak classifiers. Only the most discriminative are stored. Phase II is again an AdaBoost loop over the target training data where at each iteration the weak classifier is extracted from the set produced in the previous phase.

Single KT. Our Multi-KT algorithm chooses the best set of weights for all the prior knowledge models at once on the basis of the loss function defined in (13). An alternative approach can be defined adopting a logistic loss function [49]:

$$\ell(\tilde{y}_i, y_i) = \zeta_i \frac{1}{1 + \exp\{-10(\tilde{y}_i - y_i)\}} \ . \tag{19}$$

If we consider one single source knowledge j at the time, the corresponding loss $\ell_j(\tilde{y}_i, y_i)$ will depend on the difference $(\tilde{y}_i - y_i) = \left(\frac{a'_i}{P_{ii}} - \beta_j \frac{A''_{ji}}{P_{ii}}\right)$ for all $i = 1, \ldots, N$. Although this formulation results in a non convex objective function with respect to β_j, it is always possible to evaluate (19) for a finite set $\mathcal{S}$ of weights[6]. We can store for each source the value $\min_{\mathcal{S}}\{\sum_i \ell_j(\tilde{y}_i, y_i)\}$, and then compare all the results to identify the best prior knowledge model and its best weight. We call this variant of our method *Single-KT*.

Average Prior Knowledge. As already mentioned in the introduction, the first knowledge transfer approach able to perform one-shot learning on computer vision problems was presented in [7]. This approach does not make any assumption on the reliability of the prior knowledge, which is always considered as an average over all the known classes. The algorithm structure is strictly related to the part-based model descriptors and neither the code nor the feature used for the experiments in [7] have ever been publicly released. However, by following the proposed main idea, any single source transfer learning method can be extended to the case of multiple sources by relying on their average model.

Experiments. Here we show a benchmark evaluation of our Multi-KT algorithm against its Single-KT version, MultiSourceTrAdaBoost and TaskTrAdaBoost. We also consider A-SVM as baseline using the average of all the prior models as source knowledge, thus $\hat{\boldsymbol{w}} = \frac{1}{J}\sum_{j=1}^{J} \hat{\boldsymbol{w}}_j$ and $\beta = 1$.

We adopted the same experimental setting of the previous section with SIFT features, linear kernel and two randomly extracted sets of 10 and 20 classes from Caltech-256. In particular, the second set is obtained by adding an extra random group of 10 classes to the first one. For the boosting approaches all the learning parameters where tuned on the test set and only the best results are presented. From Figure 6 it is clear that in both the experiments Multi-KT clearly outperforms Single-KT and the two boosting methods ($p \leq 0.01$), besides producing better results than learning from scratch ($p \leq 0.01$). Moreover, for very few samples, properly weighting each prior knowledge source with Multi-KT is better ($p \leq 0.05$) than averaging over all the known models as done by A-SVM: the two approaches are equivalent only after five positive training samples with 10 classes and respectively three positive training samples for 20 classes.

The behavior of any method that chooses only one source model in transferring may vary significantly every time there is a change in the selected source. This indicates low stability. Recent work has shown that the more stable is an algorithm, the better is its generalization ability [50]. The plot on the bottom right in Figure 6 shows the comparison of Multi-KT with its Single-KT version in terms of stability. The best β_j value chosen by Single-KT can be considered as an element of the full $\boldsymbol{\beta}$ vector where all the remaining elements are zero. For each pair of subsequent steps in time, corresponding to a new added positive training sample, we calculate the difference between the obtained $\boldsymbol{\beta}$ both for Multi-KT and Single-KT. From the average norm of these differences it is evident that choosing a combination of the prior known models for transfer learning is more stable than relying on just a single source (lower average variation in the vector $\boldsymbol{\beta}$).

7.4 Heterogeneous Knowledge

In this section we consider an heterogeneous setting where each source knowledge lives in its own feature space and we compare the performance of MultiK-KT with that of Multi-KT applied on a restricted homogeneous condition. We show that the space enlarging trick at the basis of MultiK-KT, not only allows to overcome the problem due to the source variability, but it also produces better results than Multi-KT in the corresponding single space case.

We ran experiments on the same subset of data used in section 7.1. Here we considered SIFT as unique descriptor together with the generalized Gaussian kernel: $K(\boldsymbol{x}, \boldsymbol{z}) = \exp(-\gamma d_{\rho,\delta}(\boldsymbol{x}, \boldsymbol{z}))$, where $d_{\rho,\delta}(\boldsymbol{x}, \boldsymbol{z}) = \sum_i |x_i^\rho - z_i^\rho|^\delta$. Each source knowledge is defined by using the best set $\{\gamma, \rho, \delta\}$ obtained by cross validation on the corresponding object category. We learn on the target class considering the sum over the source kernels. We name *no transfer multiK* the baseline corresponding to learning from scratch in this combined space. Figure 7 presents the obtained results in comparison with the case of using a single standard Gaussian kernel, with fixed γ for sources and target tasks (*no transfer* and *Multi-KT* curves in the plot): *MultiK-KT* always performs significantly better than *Multi-KT* ($p \leq 0.002$).

Among the baseline methods considered in the homogeneous experiments, the only one that allows heterogeneous sources is TaskTrAdaBoost. We compare it with MultiK-KT over the random set of ten classes already used in the previous section. For each source we suppose to have already learned an SVM model with SIFT descriptors and Gaussian kernel where the γ parameter is set to the mean of the pairwise distances among the samples. This means that each source model lives in its own specific feature space. TaskTrAdaboost in each boosting iteration simply chooses one of the source models, while MultiK-KT learns the target task in the composed space defined by all the sources and obtained on the basis of the sum kernel. Figure 8 shows that multiK-KT outperforms TaskTrAdaBoost ($p \leq 0.01$) besides obtaining better results than learning from scratch.

6. We considered a fine tuning varying β in $\{0.01, 1\}$ with step of 0.01.

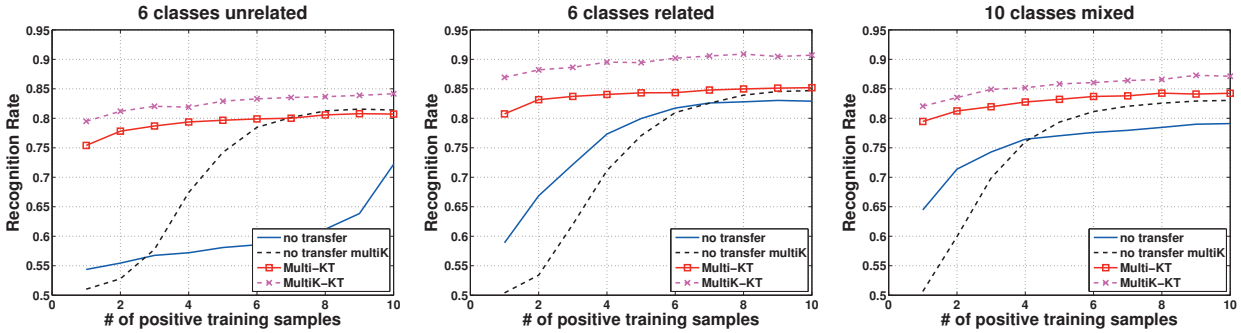

Fig. 7. Performance of the MultiK-KT method in comparison with the single kernel Multi-KT formulation. The curves identified by *no transfer* and *no transfer multiK* corresponds respectively to learning from scratch by using only the Gaussian kernel or the combination of generalized Gaussian kernels.

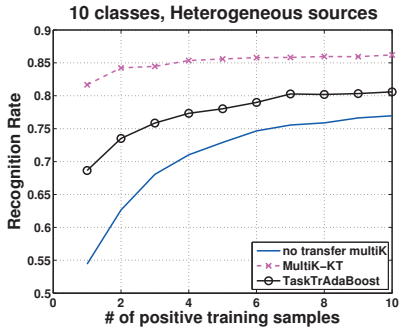

Fig. 8. Recognition rate as a function of the number of positive training samples. Each source model is defined by using a Gaussian kernel with a different γ parameter.

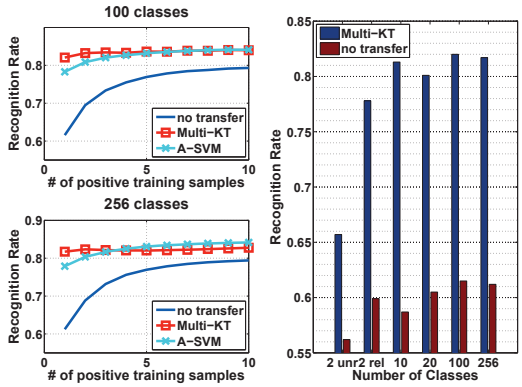

Fig. 9. Multi-KT performance for high number of source knowledge sets. Right: one shot learning recognition rate results when varying the number of prior known object categories.

7.5 Increasing Number of Sources

For any open-ended learning agent the number of known object categories is expected to grow in time. An increasing number of sources may give rise to a scalability problem in transfer learning due to the necessity of checking the reliability of each known model for the new task. Specifically, for 10^2 sources the boosting methods described in Section 7.3 become extremely expensive in computational terms (indeed in [9] they considered a maximum of 5 sources).

We performed experiments with 100 and 256 object classes from Caltech-256 dataset, reporting the result of Multi-KT, no transfer and A-SVM with average prior knowledge in Figure 9. In both cases, properly choosing the weights to assign to each source pays off with respect to average over all the sources for very few training samples: Multi-KT outperforms A-SVM ($p \leq 0.05$) for less than three positive samples. With enough training samples and a rich prior knowledge set, the best choice is to not neglect any source information.

We can expect that with a growing prior knowledge set, also the probability to find a useful source for the target task increases. To verify this behavior we focus on the Multi-KT results obtained with a single positive image. The one-shot performance for 2 unrelated classes, 2 related classes and increasing sets of 10, 20, 100 classes plus the final full set of 256 objects are summarized in Figure 9 (right). Specifically for each group of J classes we show the average one-shot recogniton result over all the possible source/target $(J-1)$classes/(1)class combinations. In this way the number of evaluations grows with the class group dimension and this may cause small oscillations in the final average results. Nevertheless, from the bars it is it is clear that by increasing the number of available sources the one-shot recognition rate obtained with Multi-KT grows. After an evident gain obtained by passing from 10^0 to 10^1 classes, the difference becomes less evident from 10^1 to 10^2 classes.

7.6 Increasing Number of Samples

Transfer learning has its maximum effectiveness in the small sample scenario in comparison to learning from scratch. However, it is also interesting to evaluate the performance of a knowledge transfer approach when the number of available training instances increases, thus checking its asymptotic behavior (see Figure 2).

We repeated the experiments on the full Caltech-256 dataset considering $\{1, 5, 10, 30, 50\}$ positive training samples with a fixed set of 50 negative training samples. We also run analogous experiments on the AwA and IRMA dataset, considering respectively 60 (60) and 70 (70) positive (negative) training samples. For all the datasets the test set contains 60 (30 positive and 30 negative) instances.

All the results are reported in Figure 10 (top line). Although it is clear the gain of Multi-KT with respect to learning from scratch for limited available data, in general this advantage

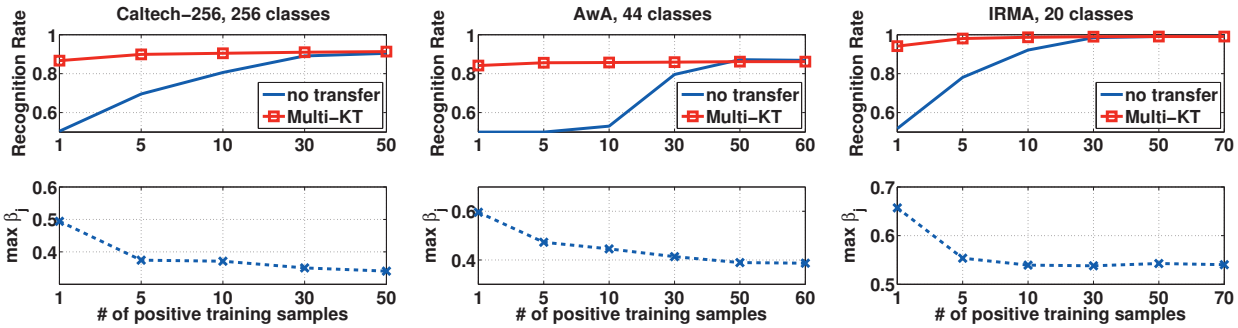

Fig. 10. Top line: recognition rate as a function of the number of positive training samples. Each experiment is defined by considering in turn one of the classes as target and the others as sources. The final results are obtained as average over ten runs. Bottom line: Maximum value over the elements of the β vector averaged over the classes and the splits.

disappears when the number of positive training samples reaches 30. The absence of the asymptotic advantage was to be expected for Multi-KT and can be justified in theoretical terms. Any universally consistent classifier will converge to the target optimal solution for an infinite number of training samples. As discussed in [51], this is the case for SVM with universal kernels, thus we expect that both our Multi-KT and the no transfer curve will reach the same asymptotic value when the amount of data increases. The only possible advantage in performance can be obtained for a reduced set of target training samples where the effect of the adaptive regularization in Multi-KT is relevant and advantageous over learning from scratch. It does not exists a general rule to establish when the small samples regime ends and the large sample regime starts, for our algorithm we showed that the limit appears around 30 target training samples.

As a final remark we underline the overall smooth behavior of Multi-KT in assigning the relevance weights to the source knowledge. In case of a single positive target training sample the prediction is strongly supported by the sources, but their importance progressively decreases when the number of training samples grows (see Figure 10, bottom line).

8 Conclusion

A learning system able to exploit prior knowledge when learning something new should rely only on the available target information for choosing from where and how much to transfer. To be autonomous it should not need an external teacher providing either information on which is the best source to use, or extra target training samples. In this paper we presented our Multi-KT algorithm, a LS-SVM based transfer learning approach with a principled technique to rely on source models and avoid negative transfer. The results of extensive experiments demonstrated the effectiveness of Multi-KT for object categorization problems with respect to other existing transfer learning methods. Moreover the weight assigned to the source knowledge set proved to be meaningful in terms of the semantic relation among the considered classes. We also extended our algorithm to the heterogeneous setting.

Recently the computer vision literature has seen an increasing interest towards high scale (10^4) object problems [5]. Most of the proposed transfer learning algorithms in this setting have been developed for object detection [52] and segmentation [53], while how to scale up the classification problem is still an open issue. Introducing a structure on the source knowledge while learning something new might be a promising strategy to use Multi-KT in this condition. Moreover the associated scalability problem due to the increasing number of training examples can be overcome by casting Multi-KT in an online learning framework [54]. All this clearly indicates possible directions for future research.

References

[1] I. Biederman, "Recognition-by-components: A theory of human image understanding," *Psychological Review*, vol. 94, pp. 115–147, 1987.

[2] D. R. Hofstadter, *Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought*. Basic Books, Inc., 1996.

[3] N. Intrator and S. Edelman, "Learning to learn," ch. Making a low-dimensional representation suitable for diverse tasks, pp. 135–157, Kluwer Academic Publishers, 1996.

[4] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results." http://www.pascal-network.org/challenges/VOC/.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[6] S. Thrun, "Is learning the n-th thing any easier than learning the first?," in *NIPS*, 1996.

[7] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, vol. 28, pp. 594–611, 2006.

[8] A. Quattoni, M. Collins, and T. Darrell, "Transfer learning for image classification with sparse prototype representations," *CVPR*, 2008.

[9] Y. Yao and G. Doretto, "Boosting for transfer learning with multiple sources.," in *Computer Vision and Pattern Recognition Conference (CVPR)*, 2010.

[10] M. Stark, M. Goesele, and B. Schiele, "A shape-based object class model for knowledge transfer," in *International Conference on Computer Vision (ICCV)*, 2009.

[11] Y. Aytar and A. Zisserman, "Tabula rasa: Model transfer for object category detection," in *International Conference on Computer Vision (ICCV)*, 2011.

[12] J. Suykens, T. V. Gestel, J. D. Brabanter, B. D. Moor, and J. Vanderwalle, *Least Squares Support Vector Machines*. World Scientific, 2002.

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

[14] V. N. Vapnik, *The nature of statistical learning theory*. Springer-Verlag New York, Inc., 1995.

[15] 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.

[16] L. Torrey and J. Shavlik, "Transfer learning," in *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*, IGI Global, 2009.

[17] http://socrates.acadiau.ca/courses/comp/dsilver/NIPS95_LTL/transfer.workshop.1995.html.
[18] W. Dai, Q. Yang, G. Xue, and Y. Yu, "Boosting for transfer learning," in *International Conference on Machine Learning (ICML)*, 2007.
[19] J. J. Lim, R. Salakhutdinov, and A. Torralba, "Transfer learning by borrowing examples for multiclass object detection," in *Advances in Neural Information Processing Systems (NIPS)*, 2011.
[20] E. Bart and S. Ullman, "Cross-generalization: Learning novel classes from a single example by feature replacement," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.
[21] M. Fink, "Object classification from a single example utilizing class relevance metrics," in *Advances in Neural Information Processing Systems (NIPS)*, pp. 449–456, 2004.
[22] U. Rückert and S. Kramer, "Kernel-based inductive transfer," in *European conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD)*, 2008.
[23] G. Mesnil, Y. Dauphin, X. Glorot, S. Rifai, Y. Bengio, I. Goodfellow, E. Lavoie, X. Muller, G. Desjardins, D. Warde-Farley, P. Vincent, A. Courville, and J. Bergstra, "Unsupervised and transfer learning challenge: a deep learning approach," in *Journal of Machine Learning Research*, vol. 27, pp. 97–110, 2012.
[24] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between class attribute transfer," in *Computer Vision and Pattern Recognition Conference (CVPR)*, 2009.
[25] J. Yang, R. Yan, and A. G. Hauptmann, "Adapting SVM classifiers to data with shifted distributions," in *International Conference on Data Mining Workshops (ICDM)*, 2007.
[26] Y. Zhang and D. Yeung, "Transfer metric learning by learning task relationships," in *ACM SIGKDD International Conference on Knowledge discovery and data mining*, 2010.
[27] J. Davis and P. Domingos, "Deep transfer via second-order markov logic," in *International Conference on Machine Learning (ICML)*, 2009.
[28] W. Dai, O. Jin, G. Xue, Q. Yang, and Y. Yu, "Eigentransfer: a unified framework for transfer learning," in *International Conference on Machine Learning (ICML)*, 2009.
[29] Y. Aytar and A. Zisserman, "Enhancing exemplar svms using part level transfer regularization," in *Proc. of British Machine Vision Conference (BMVC)*, 2012.
[30] M. E. Taylor, G. Kuhlmann, and P. Stone, "Accelerating search with transferred heuristics," in *ICAPS-07 workshop on AI Planning and Learning*, 2007.
[31] M. M. Mahmud and S. R. Ray, "Transfer learning using Kolmogorov complexity: Basic theory and empirical evaluations.," in *Advances in Neural Information Processing Systems (NIPS)*, 2007.
[32] B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2001.
[33] H. Daumé III, "Frustratingly easy domain adaptation," in *Association for Computational Linguistics Conference (ACL)*, 2007.
[34] G. C. Cawley, "Leave-one-out cross-validation based model selection criteria for weighted LS-SVMs," in *In Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, (Vancouver, BC, Canada), pp. 1661–1668, 2006.
[35] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, "Efficient projections onto the l1-ball for learning in high dimensions," in *International Conference on Machine Learning (ICML)*, 2008.
[36] D. Hush, P. Kelly, C. Scovel, and I. Steinwart, "QP algorithms with guaranteed accuracy and run time for support vector machines," *JMLR*, 2006.
[37] G. Griffin, A. Holub, and P. Perona, "Caltech 256 object category dataset," Tech. Rep. UCB/CSD-04-1366, California Institue of Technology, 2007.
[38] T. Deselaers and T. Deserno, "Medical image annotation in ImageCLEF 2008," in *working notes CLEF*, 2008.
[39] P. Gehler and S. Nowozin, "On feature combination for multiclass object classification," in *ICCV*, 2009.
[40] A. Bosch, A. Zisserman, and X. Munoz, "Representing shape with a spatial pyramid kernel," in *CIVR*, 2007.
[41] D. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.
[42] O. Tuzel, F. Porikli, and P. Meer, "Human detection via classification on riemannian manifolds," in *CVPR*, 2007.
[43] T. Ojala, M. Pietikinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognition*, vol. 29, pp. 51–59, Jan. 1996.
[44] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *CVPR*, 2006.
[45] T. Lehmann, H.Schubert, D. Keysers, M. Kohnen, and B. Wein, "The IRMA code for unique classification of medical images," in *International Society for Optical Engineering (SPIE)*, 2003.
[46] T. Tommasi, F. Orabona, and B. Caputo, "An SVM confidence-based approach to medical image annotation," in *Evaluating Systems for Multilingual and Multimodal Information Access – Proceedings of CLEF*, 2008.
[47] J. Gibbons, *Nonparametric Statistical Inference*. New York: Marcel Dekker, 1985.
[48] T. F. Cox and M. A. A. Cox, *Multidimensional Scaling*. Chapman and Hall, 2001.
[49] T. Tommasi and B. Caputo, "The more you know, the less you learn: from knowledge transfer to one-shot learning of object categories," in *BMVC*, 2009.
[50] O. Bousquet and A. Elisseeff, "Stability and generalization," *Journal of Machine Learning Research*, vol. 2, pp. 499–526, 2002.
[51] I. Steinwart, "Consistency of support vector machines and other regularized kernel classifiers," *IEEE Transactions on Information Theory*, 2005.
[52] M. Guillaumin and V. Ferrari, "Large-scale knowledge transfer for object localization in imagenet," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR '12, 2012.
[53] D. Kuettel, M. Guillaumin, and V. Ferrari, "Segmentation propagation in imagenet," in *European Conference on Computer Vision (ECCV)*, 2012.
[54] T. Tommasi, F. Orabona, M. Kaboli, and B. Caputo, "Leveraging over prior knowledge for online learning of visual categories," in *British Machine Vision Conference (BMVC)*, 2012.

Tatiana Tommasi Is a Research Assistant at KU Leuven. Her research interests include machine learning and computer vision with a focus on knowledge transfer and object categorization using multimodal information. She received her MS degree in physics (2004) and the Dipl. degree in medical physics (2008) from the University of Rome, La Sapienza, Italy. She completed her PhD in electrical engineering at the École Polytechnique Fédérale de Lausanne, in 2013.

Francesco Orabona is a Research Assistant Professor at the Toyota Technological Institute at Chicago. His research interests are in the area of theoretically motivated and efficient learning algorithms, with emphasis on online learning, kernel methods, and computer vision. He received the PhD degree in Bioengineering and Bioelectronics at the University of Genoa, in 2007. He is (co)author of more than 40 peer-reviewed papers.

Barbara Caputo is Associate Professor at the University of Rome La Sapienza since 2013, where she leads the Visual and Multimodal Applied Learning Laboratory. She received her PhD in Computer Science from the Royal Institute of Technology (KTH) in Stockholm, Sweden, in 2004. Her main research interests are in computer vision, machine learning and robotics, where she has been active since 1999. Prof. Caputo has edited 4 books, and she is (co)author of more than 70 peer-reviewed papers.