IDIAP RESEARCH REPORT

# MULTICLASS TRANSFER LEARNING FROM UNCONSTRAINED PRIORS

Jie Luo Tatiana Tommasi Barbara Caputo

Idiap-RR-25-2011

AUGUST 2011

**Centre du Parc**, Rue Marconi 19, P.O. Box 592, CH - 1920 Martigny
T +41 27 721 77 11 F +41 27 721 77 12 info@idiap.ch www.idiap.ch

# Multiclass Transfer Learning from Unconstrained Priors

Luo Jie[*] Tatiana Tommasi[*†] Barbara Caputo
Idiap Research Institute, CH-1920 Martigny, Switzerland
Swiss Federal Institute of Technology in Lausanne (EPFL), CH-1015 Lausanne, Switzerland

## Abstract

*The vast majority of transfer learning methods proposed in the visual recognition domain over the last years addresses the problem of object category detection, assuming a strong control over the priors from which transfer is done. This is a strict condition, as it concretely limits the use of this type of approach in several settings: for instance, it does not allow in general to use off-the-shelf models as priors. Moreover, the lack of a multiclass formulation for most of the existing transfer learning algorithms prevents using them for object categorization problems, where their use might be beneficial, especially when the number of categories grows and it becomes harder to get enough annotated data for training standard learning methods.*

*This paper presents a multiclass transfer learning algorithm that allows to take advantage of priors built over different features and with different learning methods than the one used for learning the new task. We use the priors as experts, and transfer their outputs to the new incoming samples as additional information. We cast the learning problem within the Multi Kernel Learning framework. The resulting formulation solves efficiently a joint optimization problem that determines from where and how much to transfer, with a principled multiclass formulation. Extensive experiments illustrate the value of this approach.*

## 1. Introduction

The visual recognition community has shown a growing interest in transfer learning algorithms in the last few years. Indeed, this type of algorithms allows to exploit prior knowledge when learning a new class, which reduces the need for annotated training data. As the frontiers in object categorization move from systems able to categorize $10^2$ objects (*e.g*. Caltech256 [15]) to systems aiming to recognize $10^4$ categories (*e.g*. ImageNet [9]), there is a growing demand for techniques able to learn robust categorization models from few labeled samples.

Transfer learning has been studied in multiple domains and under various perspectives. Many works address the issue of what to transfer (samples [3], feature representation [23], model parameters [13, 29, 31]), some focus on how to transfer (generative approaches [13, 29], boosting [36], KNN [27] and Support Vector Machine (SVM) [10, 31]), while others concentrate on how to avoid negative transfer, evaluating when and how much to transfer (different source selection approaches [31] or methods to measure the task relatedness [11]). Some knowledge transfer strategies propose to exploit sets of unlabeled target samples [23, 24] or alternative sources of extra information as attributes [12, 17].

As diverse as these approaches are, they all assume a strong control over the priors, whether in the form of constraining how the prior models are built [13, 31], or in the way of preserving the priors training samples [7, 8], or in the form of imposing the same feature representation for all priors and for the new target class [8, 31]. These constraints become particularly strict when the target problem is multiclass [25, 30].

The contribution of this paper is a multiclass transfer learning algorithm from unconstrained priors. We assume to have no control on the features from which prior models are learned, nor on the learning methods used to build the corresponding classifiers. This is achieved by using the prior knowledge as experts evaluating the new incoming data and transferring their confidence output. These outputs are used to augment the feature space of the new target data. The learning process is defined solving an optimization problem which considers both from where and how much to transfer using a principled multiclass formulation. We model our learning algorithm using the structural risk minimization principle, with a group norm regularization term which allows to tune the level of sparsity in the domain of the prior models. We show that it is possible to cast the problem within the Multi Kernel Learning (MKL) framework, and to solve it efficiently with off-the-shelf MKL solvers. We build on recent work [21] that solves the problem in the primal, resulting in a computationally efficient method that scales well with respect to the

[*] Luo Jie and Tatiana Tommasi contributed equally to this paper.
[†] Primary contact: `tatiana.tommasi@idiap.ch`

number of priors. We call the proposed method Multi Kernel Transfer Learning (MKTL).

We performed thorough experiments on two databases, studying the behavior of the algorithm in three different situations: (1) in the object category detection scenario, with priors and new models learned using the same features and learning methods; (2) in the multiclass object categorization scenario, with limited priors and few annotated samples for the target class, where priors and new models are learned using different algorithms and features, and (3) in the same scenario and setting described in (2), but scaling w.r.t. the number of available priors and w.r.t. the number of labelled samples for the new classes. For all these experiments, we compared against an existing state of the art transfer learning method, and baseline algorithms designed by us, which use (or not) the available priors. Results clearly indicate that MKTL outperforms all the other considered methods, in all the experimental settings described above. Moreover, it is able to boost significantly performance when relevant priors are available, taking advantage of the principled multiclass formulation.

In the following we introduce the notation and the transfer learning framework used in the paper (section 2). Section 3 presents the learning algorithm, discusses its properties and its connections and advantages w.r.t. existing approaches. Section 4 describes the experimental setting adopted in the paper and reports on the obtained results.

## 2. Problem Definition

This section introduces formally the notation and the transfer learning framework 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 $\bar{\boldsymbol{w}} = [\boldsymbol{w}^1, \boldsymbol{w}^2, \cdots, \boldsymbol{w}^K]$.

**Prior Knowledge.** Consider the scenario where we know $F (F \geq 2)$ categories, modeled via a classifier which is a function $f : \mathbb{X} \rightarrow \mathbb{Z}$, where $\mathbb{X}$ is the input feature space. In the binary case $\mathbb{Z} = \{-1, +1\}$, while for multiclass problems $\mathbb{Z} = \{1, \ldots, F\}$. Without loss of generality, we consider a function $f$ of the following form:

$$f(x) = \underset{z \in \mathbb{Z}}{\operatorname{argmax}}\, s_{\text{p}}(\boldsymbol{x}, z)$$

where $s_{\text{p}}(\boldsymbol{x}, z)$ is the value of the score function when the instance $\boldsymbol{x}$ is assigned to the class $z$. The score function can be interpreted as a measure of how confident the classifier is about assigning the label $z$ to the instance $\boldsymbol{x}$. In the case of binary classification, the function can be further simplified as $f(x) = \text{sign}(s(\boldsymbol{x}))$. In the rest of the paper, we will only describe our model for the multiclass situation, as its modification to the binary case is straightforward.

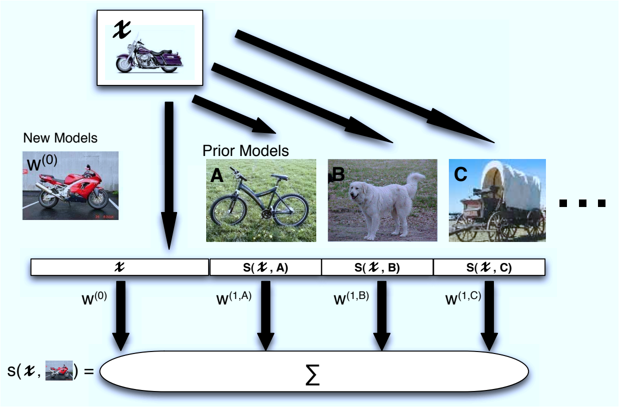

Figure 1. A graphical representation of how to use the outputs from the prior models as auxiliary features when computing the score of a new class.

**The Transfer Learning Framework.** We are interested in the task of learning a classifier for $F'$ categories, different from the $F$ categories already known (prior knowledge). Given the new training set $\{\boldsymbol{x}_i, y_i\}_{i=1}^{N}$, traditional supervised learning methods, *e.g.* SVM, minimize an upper bound of the generalization error, without taking advantage of the existing models $f$. However, when the number of training samples is small, this upper bound may become very loose and the learned model becomes unreliable. One way to improve performance is to exploit existing priors. Here, we propose to incorporate the predictions of prior knowledge models with the training samples as auxiliary features. In addition to the training sample $\boldsymbol{x}_i$, we also gather the scores $s_{\text{p}}(\boldsymbol{x}_i, z)$, $z = 1, \ldots, F$, predicted by the prior models. In this paper, we focus on the standard linear model. Therefore, when learning a new category the score function is:

$$s(\boldsymbol{x}, y) = \bar{\boldsymbol{w}} \cdot \bar{\phi}(\boldsymbol{x}, y) = \boldsymbol{w}^{(0)} \cdot \phi^{(0)}(\boldsymbol{x}, y) + \sum_{z=1}^{z=F} \boldsymbol{w}^{(y,z)} \cdot \phi^{(y,z)}\left(s_{\text{p}}\left(\boldsymbol{x}, z\right), y\right) \qquad (1)$$

where $\boldsymbol{w}^{(\cdot)}$ is a hyperplane, $\phi^{(\cdot)}(\cdot, \cdot) : \mathbb{X} \times \mathbb{Y} \rightarrow \mathbb{H}$ is the joint feature mapping function [33], which maps the samples into some high, possibly infinite dimensional space. Here, $s_{\text{p}}\left(\boldsymbol{x}, z\right)$ is the score of $\boldsymbol{x}$ labeled as class $z$ predicted by the prior models.

We use the index $0$ to indicate the feature mapping function $\phi^{(0)}(\boldsymbol{x}, y)$ for the original input features $\boldsymbol{x}$ and their corresponding model parameters $\boldsymbol{w}^{(0)}$. The indices $(y, z)$ correspond to the feature mapping of $s_{\text{p}}\left(\boldsymbol{x}, z\right)$ to the $y$-th new class, where $y = 1, \ldots, F'$ and $z = 1, \ldots, F$. In other words, given the score $s_{\text{p}}(\boldsymbol{x}, z)$ produced by a prior, $\boldsymbol{w}^{(y,z)}$ represents the contribution of the $z$-th prior model in predicting that $\boldsymbol{x}$ belongs to class $y$. Intuitively, if prior 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 prior knowledge a higher weight. On the contrary, we expect lower weights for classes which are not very relevant, such as dogs. Figure 1 illustrates the approach when computing the score for one class. Again, the predicted label is the class achieving the highest score.

Ideally, we would like to build the auxiliary feature representation using all the prior knowledge we have, and let the learning algorithm decide automatically from where to transfer and how much to transfer. Nevertheless, from a machine learning point of view, the more priors are considered, the higher is the risk for overfitting, especially when the number of training samples is limited. Moreover, among the $F$ prior models, we expect only few of them to be relevant w.r.t. a specific new class, while the rest can even add noise to the problem producing negative transfer. Both factors need to be taken in consideration when designing the learning algorithm.

**Learning the Objective Function.** The supervised learning optimization problem here is to find the modeling parameter $\bar{\boldsymbol{w}}$ that minimizes the structural risk:

$$\min_{\bar{\boldsymbol{w}}} \ \Omega(\bar{\boldsymbol{w}}) + C \sum_{i=1}^{N} \ell\left(\bar{\boldsymbol{w}}, \boldsymbol{x}_i, y_i\right) \ , \tag{2}$$

where $\Omega(\bar{\boldsymbol{w}})$ is a regularizer which avoids overfitting, $C$ is the regularization coefficient that controls the bias-variance tradeoff, and $\ell$ is some convex, non negative loss function. As stated above, we would like to encourage sparsity on the level of prior models, such that out of all the models, only a few of them are actually taking part in the scoring function. For this purpose we select the squared $(2,p)$ group norm [37] as our regularizer, $\Omega(\bar{\boldsymbol{w}}) = \frac{1}{2}\|\bar{\boldsymbol{w}}\|_{2,p}^2 = \frac{1}{2}\left\|\left[\|\boldsymbol{w}^{(0)}\|_2, \|\boldsymbol{w}^{(1,1)}\|_2, \cdots, \|\boldsymbol{w}^{(F',F)}\|_2\right]\right\|_p^2$, with $p \in (1,2]$. Each $\boldsymbol{w}^{(y,z)}$ forms its own group, and minimizing $\Omega(\bar{\boldsymbol{w}})$ corresponds to minimize the norm of each $\boldsymbol{w}^{(\cdot)}$ jointly. The parameter $p$ allows to tune the level of sparsity on the norms – increasing it if $p$ is close to 1.

**Loss Function.** Our learning problem is flexible, and we can use any convex Lipschitz loss function. For the binary case, we choose the most popular hinge loss:

$$\ell^{\mathrm{HL}}\left(\bar{\boldsymbol{w}}, \boldsymbol{x}, y\right) = |1 - y\bar{\boldsymbol{w}} \cdot \bar{\phi}(\boldsymbol{x})|_+, \tag{3}$$

where $|t|_+$ is defined as $\max(t,0)$. For the multiclass case, we use the following loss function [6, 33]:

$$\ell^{\mathrm{MC}}\left(\bar{\boldsymbol{w}}, \boldsymbol{x}, y\right) = \max_{y' \neq y} |1 - \bar{\boldsymbol{w}} \cdot (\bar{\phi}(\boldsymbol{x}, y) - \bar{\phi}(\boldsymbol{x}, y'))|_+ \ . \tag{4}$$

This function is convex and it upper bounds the multiclass misclassification loss.

# 3. Multiple Kernel Transfer Learning

## 3.1. Multiple Kernel Learning

The MKL algorithm was first proposed in [1]. It solves a joint optimization problem while also learning the optimal weights for combining the kernels. This method is theoretically sound, and it gives the possibility to integrate different data representations in a principled manner. The original MKL uses a $l_1$ norm regularization to induce sparsity in the domain of the kernels. Recently, it has been extended to $l_p$ norm regularization in [16, 21] for tuning the level of sparsity with the additional parameter $p$. This leads to better performance when the problem is not sparse. By using a generic group norm and a generic convex function, the $l_p$ norm MKL optimization can be written as:

$$\min_{\bar{\boldsymbol{w}}} \ \frac{1}{2}\|\bar{\boldsymbol{w}}\|_{2,p}^2 + C \sum_{i=1}^{N} \ell\left(\bar{\boldsymbol{w}}, \boldsymbol{x}_i, y_i\right), \tag{5}$$

where $\bar{\boldsymbol{w}} = [\boldsymbol{w}^1, \boldsymbol{w}^2, \cdots, \boldsymbol{w}^K]$, and $K$ is the number of kernels. When $p = 1$, this formulation is equivalent to the $l_1$ norm MKL optimization problem [1], and a sparse solution is obtained by solving it. However, this problem is very difficult to optimize due to the non smooth nature of the $l_1$ norm. It has been shown that when $p$ is larger than 1, the problem (5) becomes much easier to optimize [21]. Meanwhile, when $p$ tends to 1, the solution still gets extremely close to the sparse solution of $p = 1$.

## 3.2. Multi Kernel Transfer Learning

The original learning problem (2) can be converted into an $l_p$-norm MKL, which can be solved with off-the-shelf implementations [16, 21]. To transform (2), we first set

$$\bar{\boldsymbol{w}} \ = \ [\, \boldsymbol{w}^{(0)}, \ \boldsymbol{w}^{(1,1)}, \ \cdots, \ \boldsymbol{w}^{(y,z)}, \ \cdots, \ \boldsymbol{w}^{(F',F)} \,],$$

and

$$\begin{aligned} \bar{\phi}(\boldsymbol{x}, y) \ = \ [\, & \phi^{(0)}, \ \phi^{(1,1)}(s_{\mathrm{p}}(\boldsymbol{x}, 1), y), \ \cdots, \\ & \phi^{(y,z)}((s_{\mathrm{p}}(\boldsymbol{x}, z), y), \ \cdots, \ \phi^{(F',F)}(s_{\mathrm{p}}(\boldsymbol{x}, F), y) \,]\,. \end{aligned}$$

Therefore, in total, we will have $(F \times F' + 1)$ feature mapping functions $\phi^{(\cdot)}(\cdot,\cdot)$, and the same number of kernels $K^j((\boldsymbol{x}, y), (\boldsymbol{x}', y')) = \phi^j(\boldsymbol{x}, y) \cdot \phi^j(\boldsymbol{x}', y')$. This definition includes the particular case of training $F'$ different hyperplanes, one for each new class. In fact, we have that $\phi^{(0)}(x, y)$ is equal to

$$\phi^{(0)}(\boldsymbol{x}, y) = [\mathbf{0}, \cdots, \mathbf{0}, \underbrace{\psi^{(0)}(\boldsymbol{x})}_{y}, \mathbf{0}, \cdots, \mathbf{0}], \tag{6}$$

where $\psi^{(0)}(\cdot)$ is a transformation that depends only on the data. Similarly, $\boldsymbol{w}^{(0)}$ will be composed by $F'$ blocks, with each block corresponding to the hyperplane for each class, as used in [21]. The feature mapping function for the $z$-th prior model output can now be written as:

$$\phi^{(y',z)}(\boldsymbol{x},y) = \begin{cases} [\mathbf{0},\cdots,\underbrace{\psi(s_{\mathrm{p}}(\boldsymbol{x},z))}_{y},\cdots,\mathbf{0}]\ , & \text{if } y = y' \\ \qquad\qquad \mathbf{0} \qquad\qquad\quad , & \text{otherwise .} \end{cases}$$

Again, $\boldsymbol{w}^{(y',z)}$ will be composed by $F'$ blocks. However, with this construction, all the blocks of $\boldsymbol{w}^{(y',z)}$ are $\mathbf{0}$ except for the $y'$-th block. Hence, $\boldsymbol{w}^{(y',z)}$ only appears in the score functions $s(\boldsymbol{x},y')$ predicting if $\boldsymbol{x}$ belongs to the class $y'$.

### 3.3. MKL Solver and Efficient Implementations

We solve the MKTL problem using the OBSCURE [21] framework. OBSCURE is a fast stochastic subgradient descent algorithm which solves the $l_p$ norm MKL problem in the primal. Its training complexity is linear in the number of training examples. It has also been proven theoretically that OBSCURE has a faster convergence rate as the number of kernels grows, which somehow mitigates the problem that the number of kernels grows linearly with the number of priors. Moreover, the framework minimizes the primal objective function directly, even though it uses Mercer kernels. It makes the learning algorithm more memory and computationally efficient, when we can write the explicit form of feature mapping $\psi(\boldsymbol{x})$ (*e.g.* a linear kernel or polynomial kernel with a low degree).

In this paper, we will only consider a linear mapping function $\psi(\boldsymbol{x}) = \boldsymbol{x}$ (i.e. linear kernel) for the scores of prior models. Therefore, the algorithm does not need to use kernel caching for the extra $(F \times F')$ kernels coming from the prior knowledge. Similarly, the algorithm could also store $\boldsymbol{w}^{(y,z)}$ directly in its primal representation. Hence, compared to the original supervised learning problem without prior knowledge, the algorithm will use $\mathcal{O}(F \times F')$ extra memory space, and additional computational complexity at each iteration is also $\mathcal{O}(F \times F')$. In the experiments we modified the OBSCURE algorithm[1] to incorporate the auxiliary prior features and learn them efficiently, using both a binary and a multiclass loss function. For the binary version, we also modified the algorithm to obtain a weighted version for unbalance data [5], which considers a different value of $C$ for positive and negative examples.

The value of the parameter $p$ is usually defined through cross-validation, and its optimal value depends on the sparseness of the data. According to the theorems in [21], it is also possible to set $p$ equals to $\frac{2\log K}{2\log K-1}$ to get a convergence rate that depend logarithmically on the total number of kernels, which is denoted by $K$. With this setup of $p$, we have only one free parameter $C$.

### 3.4. Comparison with Existing Methods

In this section we briefly discuss other related existing approaches, emphasizing the connections and differences between them and our method.

**Using model outputs as auxiliary features.** The idea of using the output of other classifiers as basic feature representation has been well-explored in various AI domains. It recently gained popularity in the computer vision community, thanks to a large amount of annotated object image datasets that become available on the web. Several papers demonstrated that the outputs of object detectors [18], visual attributes [12, 17] and semantic visual concept [32, 35] can be used to define a good feature representation and to improve recognition performance. Our transfer learning approach follows this line of thoughts. The novelty lies in using the outputs of object classifiers as additional feature representations combined with sample features from the new target class. This makes it possible to exploit these ideas within the transfer learning framework. Moreover, we differ from these methods, as we use prediction outputs from models of similar object categories (*e.g.*, when transfers from bicycle to motorbike). This is in contrast with, for instance Object Bank [18] where the output of semantic part detectors (*e.g.*, sky, tree) are used.

Most works [12, 17, 32, 35] use features computed from the entire image. Notably different, Object Bank [18] uses a localized representation where features are extracted at different spatial pyramid levels. This is more suited for representing cluttered images composed of many objects, such as nature scenes. Although in our experiments we also use outputs computed from the entire images, the algorithm we propose can handle various multi-dimensional representations, *e.g.*, representations like Object Bank. Furthermore, MKTL takes advantage of the MKL machinery, which allows to group freely information from different unconstrained sources including the new training data into different kernels.

Finally, MKTL has a principled multiclass formulation. Each class learns from which auxiliary features to transfer in a joint optimization problem. This multiclass formulation could be generalized to other similar problems, such as those described above. It also allows to define different kernels for the new and the prior knowledge.

**Multi Model Knowledge Transfer (Multi-KT) [31].** A transfer learning algorithm close to ours is Multi-KT, which modified the $l_2$ square norm regularizer in the classical Least-Square-SVM objective function, constraining the new hyperplane $\boldsymbol{w}$ to be close to some of the hyperplanes $\boldsymbol{u}^j$ of the $F$ prior models. Its regularization term can be written as $\|\boldsymbol{w} - \sum_{j=1}^{F} \beta^j \boldsymbol{u}^j\|^2$, where $\beta^j$ is a parameter to be learned which defines the reliability of known models

[1] Available at http://dogma.sourceforge.net/

for the new learning problem, subject to the constraint that $\|\boldsymbol{\beta}\|_2 \leq 1$. The algorithm is binary, and its final decision function for a given sample $\boldsymbol{x}$ can be written as:

$$s(\boldsymbol{x}) = \boldsymbol{w} \cdot \phi(\boldsymbol{x}) + \sum_{j=1}^{F} \beta^j \boldsymbol{u}^j \cdot \phi(\boldsymbol{x}).$$

This is very similar to the binary version of the score function defined in (1). However, Multi-KT is solved based on two separate optimization problems, while our algorithm finds both the best hyperplane's parameter and the weights to be assigned to each prior knowledge model in a joint optimization process. Moreover, Multi-KT requires that each prior model $\boldsymbol{u}$ is constructed using the same type of classifier of the new model. All the models (priors and new) must also use the same type of feature descriptors. On the other hand, MKTL has neither of these constrains. It is capable of *heterogeneous transfer from unconstrained priors*: we can freely combine different learning methods and different features to boost performance. Finally, Multi-KT can not be extended to principle multiclass formulation using the multiple class loss function $\ell^{\text{MC}}$,

## 4. Experiments

We present here three sets of experiments[2] designed for studying the behavior of MKTL: (a) in the object category detection scenario, with priors and new model learned using the same features and learning methods (Section 4.1); (b) in the multiclass object categorization scenario, with limited priors and few annotated samples for the target class, where priors and new model are learned using different algorithms and features (Section 4.2), and (c) where the problem is again multiclass, but scaling w.r.t. the number of available priors and w.r.t. the number of labelled samples for the new classes. In all our experiments, the regularization parameter $C$ is selected from the set $\{0.1, 1, 10, 100, 1000\}$, and the parameter $p$ is chosen from the set $\{1.01, 1.05, 1.10, 1.15, 1.20, 1.25, 1.30, 1.40, 1.50\}$.

We compare MKTL against the following baselines:

**[No-Transfer]** It corresponds to the standard supervised learning task without considering prior knowledge. We train SVM classifiers using the 1-vs-All scheme for the multiclass extension. Ideally, the performance of a transfer learning algorithm should not be worse than this baseline, to avoid negative transfer that might hurt performance.

**[Prior-Features]** We also test the performance when using only the outputs of prior models as feature descriptors. We concatenated the outputs of the prior models into a vector representation, then use a linear SVM classifier to test their performance. This idea is similar to the classemes feature proposed in [32]. This baseline will help us understand the role of the prior models in the performance. For example, if the performance of all the prior models is very low compared to No-Transfer, we may expect to see an improvement in performance relatively small compared to standard supervised learning, and vice versa. This kind of baseline has often been ignored in previous transfer learning literature. Here we argue that it should be considered as an obligatory competitor, since sometimes using the prior models alone could lead to higher accuracy.

**[Multi-KT]** We also compared against the Multi-KT transfer learning algorithm. This method assumes that all the prior models and the new model use the same type of feature descriptors and learning method. Thus, for Multi-KT, we did train all our prior models with the same feature descriptors and kernel parameters using SVM classifier. Since this algorithm has been presented only in a binary version, we implemented the multiclass extension using the 1-vs-All scheme.

**[Average-TL]** MKTL learns the weights to combine the outputs of each prior models with the new knowledge representation. Thus, a natural baseline is to consider the information coming from the priors and the new knowledge as equally relevant. This is equivalent to train a SVM classifier using the average of all the available kernels. This method often performs as good as many MKL algorithms [14].

For all the baseline methods, we use the LIBSVM [5] package for training and testing the SVM classifier. The regularization parameter $C$ is selected from the same range as MKTL, and the best results are reported. For No-Transfer and Average-TL, we use the RBF kernel.

### 4.1. Binary Transfer Learning

We consider the same binary experimental setup proposed in [31][Section 5.3] on a subset of 30 classes plus the background class extracted from the Caltech-256 database [15]. Here we just repeat briefly the experimental procedure, for a detailed description of the setup we refer the readers to the original paper. The task is to recognize if a test image belongs to the target object class or not (*i.e.* belonging to a pre-defined background class). In turns, a small number of labelled training examples are available for a target object class and all the 29 remaining classes are used for training the prior models. We use the same four image descriptors as [31] and combine the features through concatenation. In the experiments, the number of negative examples are far larger than the number of positive examples in the training data, leading to an unbalanced data problem. This is very common in the object category detection scenario, and a popular solution to it is to give different importance weights to the positive and negative examples [31]. We modified our algorithm for this purpose. Here the weights are defined to be $w^+ = N^-/N^+$ and $w^- = 1$, where $N^+$ and $N^-$ are the number of positive and negative

[2] The code for MKTL and all the scripts used for the experiments are available online http://www.idiap.ch/~ttommasi/source_code_ICCV11.html.

samples. Both the normal ($w^+ = w^- = 1$) and weighted MKTL are considered in our experiments.

The average results of all the 30 categories as well as the average results for each class are shown in Figure 2. It can be observed that all the transfer learning methods outperform the No-Transfer approach for different numbers of training samples. Weighted MKTL achieves better performance compared to Multi-KT except for the cases with only 3 positive sample. MKTL without weights is slightly worse at the beginning, but it beats Multi-KT when the number of positive training sample reaches 15. We expect prior models to achieve high accuracy on the target task as both the prior and the target problem consist in distinguishing different objects from a common background class. It is surprising to find that using Prior-Features alone outperforms Multi-KT when the number of positive samples grows, which seems to suggest that Multi-KT is not able to combine the prior models and the new knowledge as desired (in oder to minimize the error) when the prior models are very strong. On the other hand MKTL guarantees a performance at least as good as what has been transferred. It is also interesting to look into the results obtained from each single class. Killer-whale and duck seem to exploit at the best the priors, while fern is the only case where all the transfer learning methods fail to avoid negative transfer. In most of the classes we observe that MKTL is better (or at least equal) than using Prior-Features alone.

### 4.2. Multiclass Transfer Learning

We perform multiclass classification experiments on two different datasets: subsets of the Caltech-256 [15] and the Animals with Attributes (AwA) dataset [17]. Precomputed features are available for both the databases[3].

For the experiments on the Caltech-256 dataset, we consider 9 new classes (bonsai, sunflower, mushroom, horse, skunk, gorilla, motorbikes, snowmobile, segway), and we randomly extract a maximum of 30 samples per class for training and 50 samples for testing. Twenty-three classes are considered as possible prior knowledge sources, which can be divided into four groups, plants (palm-tree, cactus, fern, hibiscus), animals (bat, bear, leopards, zebra, dolphin, killer-whale), vehicles (mountain-bike, fire-truck, car-side, bulldozer) and mix (grapes, tomato, camel, dog, raccoon, chimp, school-bus, touring-bike, covered-wagon), and we use different feature descriptors[4] for each group. For the first two groups, we concatenate the feature descriptors together, and train the prior models with Multiclass AdaBoost [28]. Then, for vehicles and mix group, we compute RBF kernels for each feature descriptor, and train SVM using the average of the RBF kernels with 1-vs-All extension. In the end we use a RBF kernel for the new training images described with PHOG [4] features. The $\gamma$ parameters of the RBF kernels were fixed to the mean of the pairwise distances among the samples as done in [14, 17]. Our choice of features descriptors and prior models are arbitrary, as we want to show that the prior models could be constructed using various features descriptors and learning algorithms. For comparison, we first consider transfer learning from the first 14 classes (from palm-tree to bulldozer). Then we progressively add the remaining 9 classes (from grapes to covered-wagon) to the prior models. Meanwhile, we also experiment with $p = \frac{2\log K}{2\log K-1}$ to test if it is possible to set the parameter $p$ automatically (MKTL-$p_{\text{auto}}$). These results are reported in Figure 3 [left].

We performed similar experiments on the AwA dataset. We consider the same 10 test classes in [17] as new classes to learn, randomly extracting a maximum of 100 samples from each class for training and 50 samples for test. The remaining 40 classes are used to build prior knowledge sources. We use the average of two RBF kernels computed using color histogram and SURF features [2] for describing all the prior classes, and train these models using SVM with 1-vs-All extension. Again, we use PHOG [4] feature with a RBF kernel for describing the new training images, and the $\gamma$ parameters are computed using the same method discussed above. These results are reported in Figure 3 [right].

MKTL clearly shows a gain in performance. It can be observed that MKTL achieves better results compared to No-Transfer, and other baseline methods, especially when the number of training samples grows (Figure 3 [left & right], after receiving 5-10 training examples per class), and more prior models are used (Figure 3 [left], 23 priors compared to 14 priors). Here the expected *higher start* effect [26] with few training samples is not as significant as in the binary case. It suggests that the multiclass problem is substantially more difficult compared to the binary object categorization task. Thus, we could expect that we need more samples for each class in order to learn the tasks. Moreover, although the performance of Prior-Features alone is relatively low, MKTL still achieves significant improvement in performance by combining the prior outputs with the new knowledge. We also see that the improvement is consistent even after receiving 100 training samples per class (Figure 3 [right]). This demonstrates the *higher asymptote* advantage for knowledge transfer [26]. This advantage is theoretically guaranteed by the fact that the knowledge transfer problem is solved in a higher dimensional feature space than the original No-Transfer. The same performance can not be expected for Multi-KT: when the number of training samples grows, the regularization term $\|\boldsymbol{w} - \sum_{j=1}^{F} \beta^j \boldsymbol{u}^j\|^2$ looses its relevance and the problem reduces to learning from scratch.

[3] AwA: http://attributes.kyb.tuebingen.mpg.de/; Caltech-256: http://www.vision.ee.ethz.ch/~pgehler/projects/iccv09/.

[4] Plants & Mix: SIFT [19] and LBP [20]; Vehicles: SIFT; Animals: REGCOV [34], SIFT and V1S+ [22]. Since Multi-KT is limited to use only one type of feature descriptor, we use PHOG [4] features for all the groups.

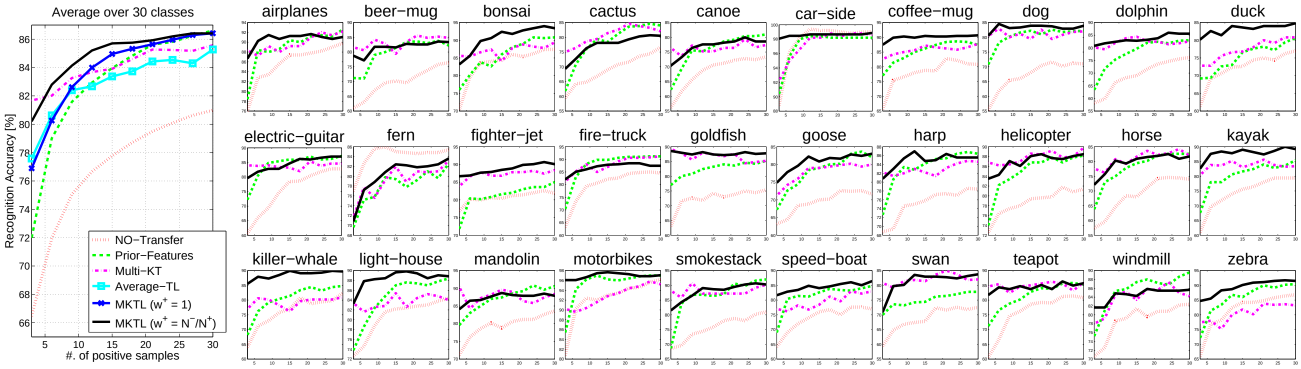

Figure 2. (Best viewed in colors and magnification.) Results obtained on the object category detection scenario, when learning one out of 30 categories with the rest categories as prior models. Classification performance is shown as a function of the number of object training images. For each class, we repeat the experiment 5 times using different random permutation. Class by class results are shown on the right. For the sake of clarity, we only plot the results of "No-Transfer", "Prior-Feature", "Multi-KT" and "MKTL ($w^+ = N^-/N^+$)" on these figures.

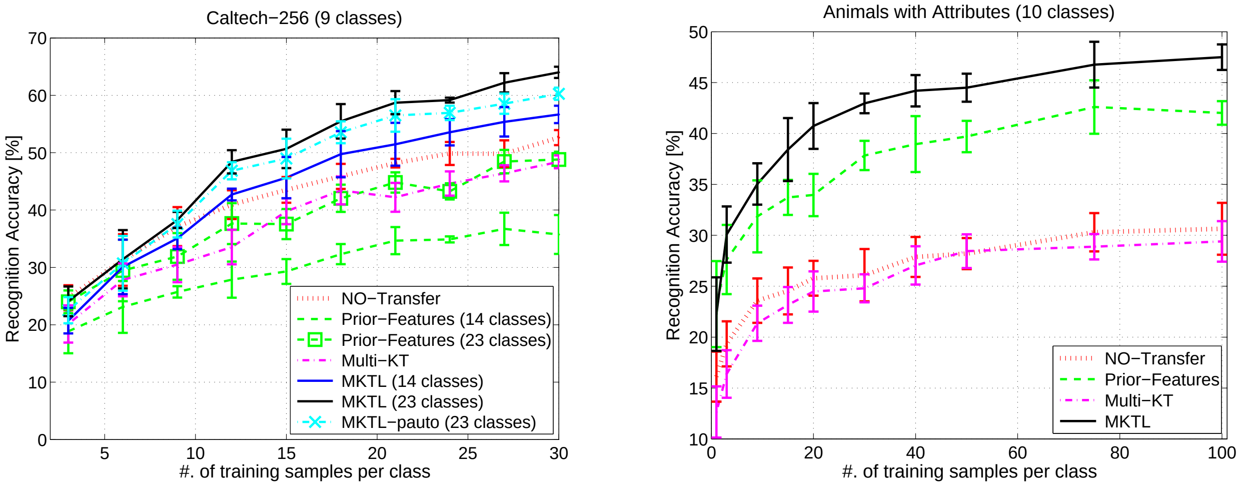

Figure 3. Results obtained on multiclass object categorization scenario. Classification performance is shown as a function of the number of object training images. [left] average results obtained using subset of the the Caltech-256 dataset; [right] average results obtained using the AwA dataset. For both datasets, each experimental setup is repeated for 10 times, and their standard deviations are also reported.

The results for MKTL using the automatic setup of the $p$ parameter is comparable to the results we obtained with cross validation on $p$. This suggests a possible way to eliminate one free parameter in practice when validation data are not available. We also tested Multi-KT on both datasets using the 1-vs-All extension. In this case, Multi-KT does not improve over the No-Transfer baseline. One possible explanation may be that the 1-vs-All scheme may induce confusion when combining the binary results over multiple classes, as the special optimization scheme used in Multi-KT does not guarantee that the output for each binary classification problem will be in a similar range. It is also worth mentioning that our learning algorithm is very efficient and takes less than 1 minute to finish, on the AwA dataset with 100 training sample per categories and 40 prior models.

## 5. Conclusions

This paper presents a multiclass transfer learning algorithm for learning object categories from few examples. The algorithm uses the output of pre-trained models as extra feature inputs, and uses a learning based approach to automatically decide from which prior models to transfer and how much to transfer. The proposed approach has no constraint on the pre-trained prior models and their features representation, as they can be built from different types of learning methods and using different types of feature representations. Furthermore, our algorithm uses a principled multiclass formulation and solves the multiclass problem in a joint optimization process. The optimization algorithm is modified from a recently proposed $l_p$-norm MKL framework which solves the optimization problem in the primal.

It thus scales well w.r.t. the number of prior models. Experiments show that our algorithm outperforms all the baseline methods, and is able to boost the performance when more relevant priors are given. Thanks to the principled multiclass formulation, the performance gain is more significant for multiclass scenarios, where the tasks are substantially more difficult than the more studied binary case.

## Acknowledgments

L. J. was supported by PASCAL Pump Priming SS2-Rob Project and T. T. by the EMMA Hasler Project. We are thankful to Francesco Orabona for his helpful comments and suggestions.

## References

[1] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan. Multiple kernel learning, conic duality, and the SMO algorithm. In *ICML*, 2004. 3

[2] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. *CVIU*, 110:346–359, 2008. 6

[3] S. Bickel, M. Brckner, and T. Scheffer. Discriminative learning for differing training and test distributions. In *ICML*, 2007. 1

[4] A. Bosch, A. Zisserman, X., and Munoz. Representing shape with a spatial pyramid kernel. In *CIVR*, pages 401–408, 2007. 6

[5] C. C. Chang and C. J. Lin. *LIBSVM: A Library for Support Vector Machines*, 2001. Software available at www.csie.ntu.edu.tw/~cjlin/libsvm. 4, 5

[6] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *JMLR*, 2:265–292, 2002. 3

[7] W. Dai, O. Jin, G.-R. Xue, Q. Yang, and Y. Yu. Eigentransfer: a unified framework for transfer learning. In *ICML*, 2009. 1

[8] H. Daumé III. Frustratingly easy domain adaptation. In *ACL*, 2007. 1

[9] J. Deng, A. C. Berg, K. Li, and L. Fei-Fei. What does classifying more than 10,000 image categories tell us? In *ECCV*, pages 71–84, 2010. 1

[10] L. Duan, I. Tsang, D. Xu, and S. Maybank. Domain transfer svm for video concept detection. In *CVPR*, 2009. 1

[11] E. Eaton, M. desJardins, and T. Lane. Modeling transfer relationships between learning tasks for improved inductive transfer. In *ECML*, 2008. 1

[12] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, 2009. 1, 4

[13] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *PAMI*, 28:594–611, 2006. 1

[14] P. Gehler and S. Nowozin. On feature combination for multiclass object classification. In *ICCV*, 2009. 5, 6

[15] G. Griffin, A. Holub, and P. Perona. Caltech 256 object category dataset. Technical Report UCB/CSD-04-1366, California Institue of Technology, 2007. 1, 5, 6

[16] M. Kloft, U. Brefeld, S. Sonnenburg, P. Laskov, K.-R. Müller, and A. Zien. Efficient and accurate $l_p$-norm multiple kernel learning. In *NIPS*. 2009. 3

[17] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009. 1, 4, 6

[18] L.-J. Li, H. Su, E. P. Xing, and L. Fei-Fei. Object Bank: A High-Level Image Representation for Scene Classification & Semantic Feature Sparsification. In *NIPS*, 2010. 4

[19] D. G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, 1999. 6

[20] T. Ojala, M. Pietikinen, and T. Menp. Multiresolution grayscale and rotation invariant texture classification with local binary patterns. *PAMI*, 24:971–987, 2002. 6

[21] F. Orabona, L. Jie, and B. Caputo. Online-batch strongly convex multi kernel learning. In *CVPR*, 2010. 1, 3, 4

[22] N. Pinto, D. D. Cox, and J. J. DiCarlo. Why is Real-World Visual Object Recognition Hard? *PLoS Comput Biol*, 4(1), 2008. 6

[23] A. Quattoni, M. Collins, and T. Darrell. Transfer learning for image classification with sparse prototype representations. In *CVPR*, 2008. 1

[24] R. Raina, A. Battle, H. Lee, and B. P. A. Y. Ng. Self-taught learning: transfer learning from unlabeled data. In *ICML*, 2007. 1

[25] M. Rohrbach, M. Stark, G. Szarvas, I. Gurevych, and B. Schiele. What helps where? and why? semantic relatedness for knowledge transfer. In *CVPR*, 2010. 1

[26] M. Rosenstein, Z. Marx, and L. P. Kaelbling. To transfer or not to transfer. In *NIPS*, 2005. 6

[27] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *ECCV*, 2010. 1

[28] R. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37:297–336, 1999. 6

[29] M. Stark, M. Goesele, and B. Schiele. A shape-based object class model for knowledge transfer. In *ICCV*, 2009. 1

[30] K. Tang, M. Tappen, R. Sukthankar, and C. Lampert. Optimizing one-shot recognition with micro-set learning. In *CVPR*, 2010. 1

[31] T. Tommasi, F. Orabona, and B. Caputo. Safety in numbers: Learning categories from few examples with multi model knowledge transfer. In *CVPR*, 2010. 1, 4, 5

[32] L. Torresani, M. Szummer, and A. Fitzgibbon. Efficient object category recognition using classemes. In *ECCV*, pages 776–789, 2010. 4, 5

[33] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *ICML*, 2004. 2, 3

[34] O. Tuzel, F. Porikli, and P. Meer. Human detection via classification on riemannian manifolds. In *CVPR*, 2007. 6

[35] J. Vogel and B. Schiele. Semantic modeling of natural scenes for content-based image retrieval. *IJCV*, 2008. 4

[36] Y. Yao and G. Doretto. Boosting for transfer learning with multiple sources. In *CVPR*, 2010. 1

[37] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *J. Roy. Stat. Society*, 68:49–67, 2006. 3