Leveraging over prior knowledge for online learning of visual categories

Tatiana Tommasi[12]
ttommasi@idiap.ch
Francesco Orabona[3]
francesco@orabona.com
Mohsen Kaboli[1]
mkaboli@idiap.ch
Barbara Caputo[1]
bcaputo@idiap.ch

[1] Idiap Research Institute, Martigny, CH
[2] École Polytechnique Fédérale EPFL Lausanne, CH
[3] Toyota Technological Institute Chicago, USA

Abstract

Open ended learning is a dynamic process based on the continuous analysis of new data, guided by past experience. On one side it is helpful to take advantage of prior knowledge when only few information on a new task is available (transfer learning). On the other, it is important to continuously update an existing model so to exploit the new incoming data, especially if their informative content is very different from what is already known (online learning). Until today these two aspects of the learning process have been tackled separately. In this paper we propose an algorithm that takes the best of both worlds: we consider a sequential learning setting, and we exploit the potentiality of knowledge transfer with a computationally cheap solution. At the same time, by relying on past experience we boost online learning to predict reliably on future problems. A theoretical analysis, coupled with extensive experiments, show that our approach performs well in terms of the online number of training mistakes, as well as in terms of performance on separate test sets.

1 Introduction

The underlying main goal of all research in visual recognition is to enable vision-based artificial systems to operate autonomously in the real world. However, even the best system we can currently engineer is bound to fail whenever the setting is not heavily constrained. This is because the real world is generally too nuanced, too complicated and too unpredictable to be summarized within a limited set of specifications. There will be inevitably novel situations and the system will always have gaps, conflicts or ambiguities in its own knowledge and capabilities. This calls for algorithms able to support open ended learning of visual classes.

The open ended learning issue, i.e. the ability to learn a new detected class continuously over time, has been typically addressed in a fragmented fashion in the literature. A first component is that of transfer learning, i.e. the ability to leverage over prior knowledge when learning a new class, especially in presence of few training data [10, 11, 15, 17, 18]. A second component is that of updating continuously the learned visual class, as new samples

 http://dx.doi.org/10.5244/C.26.87

arrive sequentially. The dominant approach in the literature here is that of online learning: predictions are made on the fly and the model is progressively updated at each step, on the basis of the given true label. An attractive feature of this family of algorithms is that they aim at minimizing the number of total mistakes on the incoming samples (mistake-bound).

In this paper we propose to merge together these two components, using the prior knowledge sources for initializing the online learning process on a new target task through transfer learning. This has two main advantages: (1) by using a principled transfer learning process we can study the relation between the old sources and the new target. Within this framework, few samples might be sufficient to indicate in which part of the original space the correct solution (the best in term of generalization capacity) should be sought. (2) we show theoretically that a good initialization for the online learning process produces a tighter mistake bound compared to previous work [18], while empirically improving the recognition performance on an unseen test set. Globally an expensive transfer learning approach is used only at the beginning, therefore limiting the computational burden. Then, a fast and efficient online approach is applied. We choose the Passive Aggressive online learning algorithm [4], and we show how to initialize it in two different fashions with a state of the art transfer learning method [17]. For each of the two versions of the algorithm, we derive the relative mistake bound, which provide us with a deeper understanding of the methods. Experiments on the object categorization domain show the potential of our approach.

The paper is organized as follows: after a brief review of related works in this section, we outline the Passive Aggressive method (section 2.1), and describe two existent batch and online transfer learning algorithms (sections 2.2 and 2.3) on which we build. The new derived algorithms, together with their theoretical analysis are introduced in section 2.4. Section 3 reports our findings and comments the implications of the results. We conclude with an overall discussion.

Related Work To our knowledge, the most similar approach to ours presented in the literature is Online Transfer Learning (OTL) [18]. Based on ensemble learning, it builds online a prediction function on the data of the target domain, and mix it with the old prediction function learned on the source domain. The weights for the combination are adjusted dynamically on the basis of a loss function which evaluates the difference among the current prediction and the correct label of any new incoming sample. This method does not consider the case of transfer from multiple sources: for a single prior knowledge model a theoretical analysis demonstrates the existence of a mistake bound for OTL.

It has been shown that for recommender problems [1] and robotics applications [5] learning online on a new task with a good initialization based on source knowledge can be very helpful. [13, 14] present active learning techniques which, by leveraging over different but related source domains, get advantage on a new target, querying experts for more labeled data only when necessary. Recently [9] introduced an online approach based on Gaussian process regression for rapidly adapting pre-trained classifiers to a new test domain improving the performance in face detection problems.

2 The Learning Approach

In this section we introduce formally the notation and we briefly recap the techniques on which we build before presenting our learning algorithm.

We consider the case in which each instance is represented by a vector $\boldsymbol{x} \in \mathbb{R}^d$ associated to a unique label $y \in \{-1, 1\}$ and the prediction mechanism is based on a hyperplane which

divides the instance space into two parts. This hyperplane is defined by its orthogonal vector $\boldsymbol{w} \in \mathbb{R}^d$ and the predicted label is given by $\text{sign}(\boldsymbol{w}\cdot\boldsymbol{x})$. We will also assume without loss of generality that $\|\boldsymbol{x}_t\| \leq 1$. We also define the hinge loss with margin 1 of a classifier $\boldsymbol{w}$ over an instance/label pair $(\boldsymbol{x}, y)$ as

$$\ell^H(\boldsymbol{w}\cdot\boldsymbol{x}, y) = \max\{0, 1 - y\boldsymbol{w}\cdot\boldsymbol{x}\} \,. \tag{1}$$

2.1 Online Learning

In the online learning framework a learner is presented with a sequence of instances $\boldsymbol{x}_t$, $t = 1,\ldots,T$. After each instance $\boldsymbol{x}_t$ it generates the corresponding prediction. Then, the true label y_t is given to the learner, that uses this feedback to update its hypothesis for future trials. The aim of an online algorithm is to minimize its cumulative loss on the sequence of data, measured using an arbitrary loss function. For a thorough introduction to the online learning theory we refer the reader to [3].

In the linear setting defined above, at each step we estimate the hyperplane $\boldsymbol{w}_t$ and predict with $\text{sign}(\boldsymbol{w}_t\cdot\boldsymbol{x}_t)$, while the quantity $\boldsymbol{w}_t\cdot\boldsymbol{x}_t$, that corresponds to the distance between the instance and the hyperplane, can be roughly seen as the confidence on the prediction.

We focus here on the Passive Aggressive algorithm (PA) [4]. It learns an online classifier which is updated at each step minimizing an objective function that trades off the maximum closeness to the current classifier and the hinge loss on the most recent example[1]. Starting from an arbitrary hypothesis, $\boldsymbol{w}_1$, at the $t-$th round PA is updated solving the following optimization problem

$$\boldsymbol{w}_{t+1} = \underset{\boldsymbol{w}}{\text{argmin}} \frac{1}{2}\|\boldsymbol{w} - \boldsymbol{w}_t\|^2 + C\xi \quad \text{s.t.} \quad \ell^H(\boldsymbol{w}\cdot\boldsymbol{x}_t, y_t) \leq \xi \quad \text{and} \quad \xi \geq 0 \,, \tag{2}$$

that results in a simple closed form

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \gamma_t y_t \boldsymbol{x}_t \quad \text{where} \quad \gamma_t = \min\left\{C, \frac{\ell^H(\boldsymbol{w}_t\cdot\boldsymbol{x}_t, y_t)}{\|\boldsymbol{x}_t\|^2}\right\}, \tag{3}$$

where C is the aggressiveness parameter that trades off the two quantities in (2). Hence the hypothesis is updated each time there is a prediction error, or the prediction is correct but the magnitude of the prediction is too low, i.e. the algorithm is not confident enough. When PA is implemented in dual variables [9], each update requires the computation of γ_t which costs $\mathcal{O}(t)$ where t indicates the number of samples seen till that moment. Considering a full set of T instances, the total computational complexity of PA is $\mathcal{O}(T^2)$.

2.2 Transfer Learning

Among the existing transfer learning approaches we consider the Multi-KT algorithm [17]. Its main advantage is that it proposes a principled solution to evaluate automatically the relatedness among multiple sources and the new target task. A discriminative model for the target task is then learned with the condition of closeness to a weighted combination of prior knowledge models.

More formally, we suppose to have k binary source tasks each containing N_k samples $(\boldsymbol{x}_i, y_i)$ $i = 1,\ldots,N_k$. A discriminative model is learned for each task in terms of a linear

[1] We consider the Passive Aggressive version defined as PA-I in [9] that for simplicity we denote by PA.

function $h_j(\boldsymbol{x}) = \hat{\boldsymbol{w}}_j \cdot \boldsymbol{x}$ for $j = 1, \ldots, k$. For a novel target problem with T available training samples living in the same data space of the sources $(\boldsymbol{x}_t, y_t)$ $t = 1, \ldots, T$, Multi-KT solves the following optimization problem [17]:

$$\min_{\boldsymbol{w},b} \frac{1}{2} \left\| \boldsymbol{w} - \sum_{j=1}^{k} \beta_j \hat{\boldsymbol{w}}_j \right\|^2 + \frac{C}{2} \sum_{t=1}^{T} (y_i - \boldsymbol{w} \cdot \boldsymbol{x}_t - b)^2 \, . \tag{4}$$

This problem has the same objective function of LS-SVM [16] where the second term is the square loss, while the regularizer has been modified to impose closeness between the new target model and a linear combination of source models. Here the weights β_j assigned to each prior knowledge are found by minimizing $\sum_{t=1}^{T} \ell^H(\tilde{y}_t, y_t)$ subject to $\|\boldsymbol{\beta}\|_2 \leq 1$, where $\tilde{y}_t$ is the leave-one-out prediction for the $t-$th sample, and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_k)$. With this formulation the final prediction function on the target task is

$$f(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x} + b = \left(\sum_{j=1}^{k} \beta_j \hat{\boldsymbol{w}}_j + \sum_{t=1}^{T} \alpha_t \boldsymbol{x}_t \right) \cdot \boldsymbol{x} + b \, , \tag{5}$$

where α_t are the coefficients of the support vectors for the new target problem. From the computational point of view the runtime for Multi-KT is $\mathcal{O}(T^3 + kT^2)$ where the first term is related to solving the problem in (4) while the second term is the cost of calculating $y_t \tilde{y}_t$. We are not taking into consideration the learning process on the source tasks, supposing that the prior knowledge models are given as input to the Multi-KT algorithm.

2.3 OTL: Online Transfer Learning

The OTL algorithm proposed in [18] is a two stages online learning approach which combines a source classifier $h(\boldsymbol{x})$ with a prediction function $f(\boldsymbol{x})$ learned online on the target domain. Specifically f is learned from a sequence of samples $(\boldsymbol{x}_t, y_t)$ $t = 1, \ldots, T$. At the $t-$trial the learner receives an instance $\boldsymbol{x}_t$ and the prediction function f_t is updated to f_{t+1} according to the PA rule (3) with $f_t(\boldsymbol{x}_t) = \boldsymbol{w}_t \cdot \boldsymbol{x}_t$. In addition, the corresponding class label is predicted by the following ensamble function [18]:

$$\hat{y}_t = \text{sign} \left(\sigma_t \Pi(h(\boldsymbol{x}_t)) + \tau_t \Pi(f_t(\boldsymbol{x}_t)) - \frac{1}{2} \right) \, , \tag{6}$$

where $\Pi(x) = \max\{0, \min\{1, \frac{x+1}{2}\}\}$ is a normalization function. The weights are initialized as $\sigma_1 = \tau_1 = \frac{1}{2}$ and at each step they are adjusted dynamically according to [18]

$$\sigma_{t+1} = \frac{\sigma_t s_t(h)}{\sigma_t s_t(h) + \tau_t s_t(f_t)} \, , \qquad \tau_{t+1} = \frac{\tau_t s_t(f_t)}{\sigma_t s_t(h) + \tau_t s_t(f_t)} \, , \tag{7}$$

where $s_t(g) = \exp\{-\frac{1}{2} \ell^S(\Pi(g(\boldsymbol{x}_t)), \Pi(y_t))\}$ and $\ell^S(z, y) = (z - y)^2$ is the loss function.

The proposed method originally supposes the existence of one unique source domain. In case of multiple source tasks we suggest the naïve solution of averaging all the prior knowledge models and use the mean classifier as $h(\boldsymbol{x})$. A different solution consists in assigning one weight to each source knowledge. In this case we start from $\sigma_1 = \sum_{j=1}^{k} \sigma_{j,1} = \tau_1 = \frac{1}{2}$ with $\sigma_{j,1} = \frac{1}{2k}$ for $j = 1, \ldots, k$ and then we update the weights with

$$\sigma_{j,t+1} = \frac{\sigma_{j,t} s_t(h_j)}{\sum_{j=1}^{k} \sigma_{j,t} s_t(h_j) + \tau_t s_t(f_t)} \, , \qquad \tau_{t+1} = \frac{\tau_t s_t(f_t)}{\sum_{j=1}^{k} \sigma_{j,t} s_t(h_j) + \tau_t s_t(f_t)} \, . \tag{8}$$

If we neglect the prior knowledge learning process as before, the total computational complexity of OTL matches the one of the online learning method used, since the cost of (7) (and (8)) is $\mathcal{O}(1)$. Thus we have $\mathcal{O}(T^2)$ as for PA.

Theoretical Analysis In the particular case of one single source task, for the OTL algorithm is possible to prove a bound on the number of mistakes M made during the online learning process (see Theorem 1 and its proof in [18]):

$$M \leq 4 \min\left\{\Sigma_h, \Sigma_f\right\} + 8 \ln 2 \,, \tag{9}$$

where $\Sigma_h = \sum_{t=1}^{T} \ell^S(\Pi(h(\boldsymbol{x}_t)), \Pi(y_t)))$ and $\Sigma_f = \sum_{t=1}^{T} \ell^S(\Pi(f_t(\boldsymbol{x}_t)), \Pi(y_t)))$. Note that the first stage in OTL is based on the PA algorithm, that uses the hinge loss, while the second stage uses the square loss. Hence in [18] the Authors observe that, if we denote by M_h and M_f the mistake bound of the model h and f_t respectively, and we assume $\ell^S(\Pi(h(\boldsymbol{x}_t)), \Pi(y_t))) \approx \frac{1}{4} M_h$ and $\ell^S(\Pi(f_t(\boldsymbol{x}_t)), \Pi(y_t))) \approx \frac{1}{4} M_f$, then $M \leq \min\{M_h, M_f\} + 8 \ln 2$.

2.4 TROL: TRansfer initializes Online Learning

The main issue faced by OTL is how to combine online the source and the target knowledge that are learned independently in an initial stage. On the other hand Multi-KT provides a model for the new target problem on the basis of very few training samples exploiting a reliable combination of prior models. This is a batch approach directly meant to minimize the generalization error of the obtained target model. Since Multi-KT operates in the small setting scenario, we can use it to define an hybrid batch-online learning approach different from OTL. It is based on two phases: at the beginning n target training samples are given as input to Multi-KT which outputs the corresponding target model, and as second step, this model is used to initialize the online learning process. Using PA, the updated solution will be at each step close to the previous one: this helps keeping the advantage given by Multi-KT together with the proper introduction of new information when necessary. We name this algorithm TROL: TRansfer initializes Online Learning.

Formally, training Multi-KT on n target samples consists in solving the optimization problem in (4). The obtained model $\boldsymbol{w}_1 = (\sum_{j=1}^{k} \beta_j \hat{\boldsymbol{w}}_j + \sum_{i=1}^{n} \alpha_i \boldsymbol{x}_i)$ is then introduced in (2) as initialization when learning from the $(n+1)-$th training sample on. The Multi-KT algorithm is applied on n (typically $n \leq 10$) training samples and k prior knowledge sources, before starting the online learning process. Hence the final cost is $\mathcal{O}(T^2 + n^3 + kn^2)$, that for enough samples T is dominated by the complexity of PA. In other words the added complexity of using Multi-KT on n samples is negligible.

Theoretical Analysis With respect to PA, a good initialization model can improve the mistake bound. In fact it is easy to generalize the mistake bound in [4] to the case of using a $\boldsymbol{w}_1$ different from the null vector. In this case we have that the number M of prediction mistakes satisfies

$$M \leq 2 \max\left\{1, \frac{1}{C}\right\} \left(\frac{1}{2} \|\boldsymbol{u} - \boldsymbol{w}_1\|^2 + C \sum_{t=1}^{T} \ell^H(\boldsymbol{u} \cdot \boldsymbol{x}_t, y_t) \right) . \tag{10}$$

From this bound we have that it is possible to improve the performance of the algorithm, at least in the worst case scenario, by initializing it with a classifier that is close to the optimal one.

2.4.1 Update the transfer weights

The learning solution described above to integrate old and new knowledge is based on a proper initialization of the online process. Still, the old knowledge is not directly reweighted during the learning process. We show here that it is possible to use a simple feature augmentation trick to have the same starting condition of TROL together with a progressive update of the source and the target knowledge weights in time. We call this algorithm TROL+.

Given the model $\boldsymbol{w}_1$, we can evaluate its prediction on each new training samples as $\boldsymbol{w}_1 \cdot \boldsymbol{x}_t$. Cropping the obtained value between -1 and 1, similarly to OTL, to limit the norm of the added dimension, we use this prediction as the $(d+1)$-th element in the feature vector descriptor of $\boldsymbol{x}_t$. So we define

$$\boldsymbol{x}'_t = (\boldsymbol{x}_t, v_t) \in \mathbb{R}^{d+1} \quad \text{where} \quad v_t = \max\{-1, \min\{1, \boldsymbol{w}_1 \cdot \boldsymbol{x}_t\}\} \; .$$

The samples with such a modified representation enter the PA algorithm initialized now with $\boldsymbol{w}'_1 = (0, \ldots, 0, 1) \in \mathbb{R}^{d+1}$. At $t = 1$ PA predicts with $\text{sign}(\boldsymbol{w}'_1 \cdot \boldsymbol{x}'_1) = \text{sign}(v_1)$ while for any t the updating rule in (3) results in

$$\boldsymbol{w}'_{t+1} = \boldsymbol{w}'_t + \gamma_t y_t \boldsymbol{x}'_t \quad \text{where} \quad \gamma_t = \min\left\{C, \frac{\ell^H(\boldsymbol{w}'_t \cdot \boldsymbol{x}'_t, y_t)}{\|\boldsymbol{x}'_t\|^2}\right\}, \tag{11}$$

and the predictions are

$$\boldsymbol{w}'_t \cdot \boldsymbol{x}'_t = \sum_{i=1}^{t-1} \gamma_i y_i (\boldsymbol{x}_i \cdot \boldsymbol{x}_t + v_i v_t) \; . \tag{12}$$

Hence the hyperplane $\boldsymbol{w}'_t$ can be thought as composed by two parts, one for the old knowledge and one for the knowledge coming from the new instances. Of course this approach can be generalized to allow the use of k different prior models $\boldsymbol{w}_1^j \; j = 1, \ldots, k$, expanding the input vectors with k new dimensions

$$\boldsymbol{x}'_t = (\boldsymbol{x}_t, v_{1,t}, \ldots, v_{k,t}) \in \mathbb{R}^{d+k} \quad \text{where} \quad v_{j,t} = \max\{-1, \min\{1, \boldsymbol{w}_1^j \cdot \boldsymbol{x}_t\}\} \; . \tag{13}$$

Theoretical Analysis A theoretical support to TROL+ is given by the following theorem

Theorem 1 *Let $(\boldsymbol{x}'_t, y_t)$, $t = 1, \ldots, T$ be a sequence of transformed instances as in (13), $y_t \in \{+1, -1\}$ and $\|\boldsymbol{x}_t\| \leq 1$ for all t. Then, for any vector $\boldsymbol{u} \in \mathbb{R}^{d+k}$ the number of prediction mistakes made by TROL+ on this sequence of examples is bounded from above by*

$$M \leq 2 \max\left\{(1+k), \frac{1}{C}\right\} \left(\frac{1}{2}\|\boldsymbol{u} - \boldsymbol{w}'_1\|^2 + C \sum_{t=1}^{T} \ell^H(\boldsymbol{u} \cdot \boldsymbol{x}'_t, y_t)\right),$$

where C is the aggressiveness parameter provided to TROL+.

The proof follows immediately by considering the bound in (10) and the increased dimensionality of the instances. To compare this bound to the one of OTL, let us set $C = 1$ and use only one prior knowledge, i.e. $k = 1$. Given that the bound in (10) holds for any $\boldsymbol{u}$, we can worsen the bound by setting $\boldsymbol{u}$ to be the optimal one for the new knowledge alone or the prior knowledge alone, to have that $M \leq 4 \min\left\{\Sigma_h, \Sigma_f\right\}$, where $\Sigma_h = \sum_{t=1}^{T} \ell^H(v_t, y_t) \leq \sum_{t=1}^{T} \ell^H(w_1 \cdot \boldsymbol{x}_t, y_t)$, and $\Sigma_f = \min_{\boldsymbol{u}} \frac{1}{2}\|\boldsymbol{u}\|^2 + \sum_{t=1}^{T} \ell^H(\boldsymbol{u} \cdot \boldsymbol{x}_t, y_t)$. Hence, as in OTL, the performance of TROL+ is always close to the best between the performance of the prior and the

performance of the best batch classifier over the new knowledge. However here we have the hinge loss ℓ^H and not the square loss ℓ^S as in OTL. It is known that the first one approximates the real 0/1 loss better than the second [2, 12]. Moreover, as discussed in section 2.3, the OTL bound does not directly link the performance to the two stages of the algorithm, while in TROL+ there is only one layer so we do not have this problem. Another difference with OTL is that TROL+ will make only a finite number of mistakes if there is an hyperplane $\boldsymbol{u}$ that correctly classifies all the samples. In the next section we will show that this theoretical advantage is also evident in the empirical experiments.

3 Experiments

We ran experiments on the Caltech-256 dataset [8], following the setting chosen in [17]. Besides considering the full database, we focused on some selected classes to show the performance of transfer learning in case of different level of relatedness between the source and target tasks and to evaluate the eventual negative transfer (decrease in performance w.r.t learning from scratch [11]). Feature-wise, we used the publicly available SIFT descriptors of [7]. The training (test) set for each class consisted of 60 (100) samples. Each set contains an equal number of positive (object class) and negative (background) examples. We considered 10 random orderings of the samples for each class and we present the average results on these ten splits both in terms of the average error rate for the online methods and of the recognition rate produced by the current training solution on the test set. The training set are organized such that any positive sample is always followed by a negative one and vice-versa. For all experiments we used the Gaussian kernel $K(\boldsymbol{x},\boldsymbol{x}') = \exp(-\frac{1}{\delta}\|\boldsymbol{x}-\boldsymbol{x}'\|^2)$, fixing δ to the mean of the pairwise distances among the samples. For the particular feature augmentation technique used in TROL+, we considered the linear combination of two kernels $(K_1 + K_2)$ where the first is Gaussian and deals with the SIFT feature descriptor, while the second is a linear kernel applied on the extra feature elements obtained by the prediction of the priors.

We benchmarked TROL and TROL+ against PA trained on the target samples, Multi-KT and OTL, where in case of multiple priors we considered the average of all the available models as source classifier. We also defined other three baselines:

NOTR This is a batch strategy corresponding to learn using only the target samples (no transfer). It uses LS-SVM on the available set of training samples at each step.

M-OTL This is our modified version of OTL able to assign a different weight to each prior knowledge in case of multiple sources, with the update rule defined in (8).

TR-OTL This method considers as source knowledge for OTL the same Multi-KT output that we use as initialization in TROL.

All the online techniques initialized with Multi-KT use its output model learned over $n = 6$ training images, corresponding to three positive and three negative samples. All the source models have been learned with LS-SVM. The value of the C parameter is chosen by cross validation on the sources and we used the same for the batch methods (Multi-KT and NOTR) applied on the new task. The C value for all the online methods is instead fixed to 1.

Single source This is the setting in which OTL was originally presented and evaluated. We ran experiments on different couple of classes, chosen inside the macro categories defined by the dataset taxonomy (*e.g.* related objects in food-containers) or extracted randomly. For all the couples we consider one of the classes as target task and the other as source knowledge, repeating the experiments twice switching the role of the two classes. Three representative results are reported in Figure 1. For the unrelated couple fireworks-treadmill (left column),

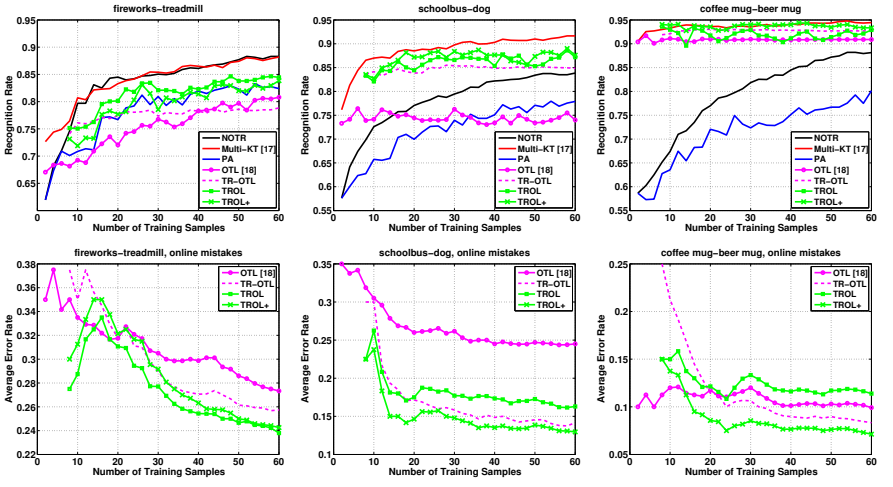

Figure 1: Single source experiments: couples of classes with increasing relatedness from left to right, as empirically shown by the growing advantage of Multi-KT over NOTR. Top line: recognition rate results on the test set as a function of the number of training samples. Bottom line: corresponding rate of mistakes for the online learning methods.

TROL and TROL+ matches the recognition performance of the corresponding no transfer online learning method PA, while OTL and TR-OTL suffer for negative transfer. The case schoolbus-dog (middle column) represents an intermediate condition, where transfer learning can be helpful. Here TROL and TROL+ present a small advantage over TR-OTL in terms of recognition on the test set, while TROL+ and TR-OTL show the best performance on the mistake rate. Finally for coffee-mugs and beer-mugs (right column) all the transfer learning methods perform much better than learning from scratch with a particular advantage of TROL+ in terms of online mistake rate.

Multiple sources We focus here on the case where multiple priors are available. Figure 2 shows the results on a group of four unrelated classes (originally used in [6]). Each of them is considered in turn as target task while the remaining ones define three source knowledges. Despite the difference among the object categories, Multi-KT is able to define a good combination of priors and exploit it when learning on the target, obtaining extremely good results in classification. All the online methods initialized with Multi-KT (TROL, TROL+, TR-OTL) matches its recognition performance after 10 training samples. OTL considering the average source knowledge shows instead negative transfer. M-OTL, based on different weights for each source classifier, does not have any advantage w.r.t learning from scratch (PA), but at least it is not worse. TROL, TROL+ and TR-OTL have the best performance with respect to all the other baselines in terms of average rate of mistakes. A special remark is necessary here for the method named "TROL+ (priors)". This refers to the case in which each prior knowledge model is considered as a separate source, so we are augmenting the feature space with $k = 3$ new elements. This method outperforms OTL and M-OTL both in terms of mistake rate and recognition on the test set, roughly matching the batch performances of Multi-KT after 20 training samples.

The four plots on the right in Figure 2 show how the weights given to source and target knowledge change in the OTL-related methods. The information obtained as output from Multi-KT, used as source in TR-OTL, maintains a high weight in time. This demonstrates

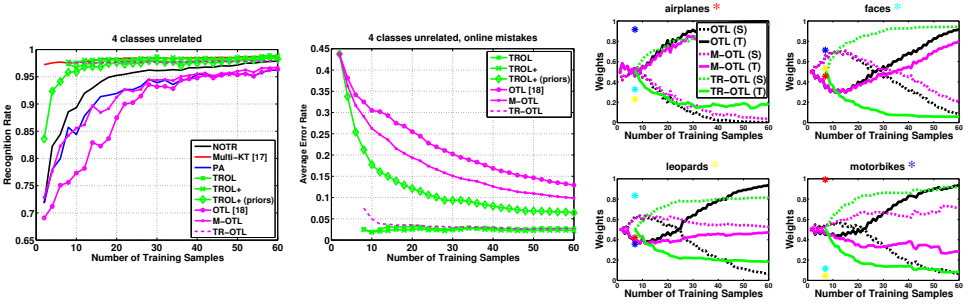

Figure 2: Four unrelated classes: airplanes, motorbike, faces, leopards. Left: recognition rate results on the test set as a function of the number of training samples. Middle: corresponding rate of mistakes for the online learning methods. Right: value of the weights given to source (S) and target (T) knowledge by the OTL-related methods for one split. The line "M-OTL (S)" corresponds to the sum of all the weights separately given to the sources. The stars indicate the weight given to each of the source classes by Multi-KT and used in the input model to TR-OTL.

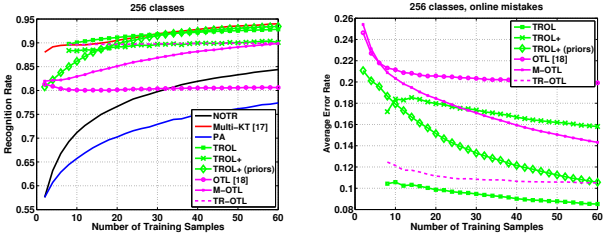

Figure 3: Recognition rate results on the test set as a function of the number of training samples and corresponding average error rate for the online methods on the whole Caltech-256 dataset. All the results are obtained as average over each of the classes considered as target task with the remaining 255 used as sources.

its usefulness for the learning process. On the other hand, the source knowledge looses its importance when the number of training samples increase, or show a small weight, for OTL and M-OTL.

Figure 3 presents the results for the full Caltech-256 dataset. Here the online method TROL performs as the batch algorithm Multi-KT and shows the best results w.r.t. all the other baselines in terms of mistake rate. Both OTL and TR-OTL do not seem able to use properly the new information given by the incoming training samples, showing almost a flat performance on the test set. We see again the good results of "TROL+ (priors)" that directly using and reweighting multiple prior knowledges outperforms OTL and M-OTL, and matches TR-OTL in terms of average error rate with 60 training samples.

In summary, over all the experiments TROL and TROL+ are better or at least as good as PA, never showing negative transfer, and can match the batch performance of Multi-KT on the test set. In terms of online mistakes, they outperform all the other baselines.

4 Conclusions

In this paper we addressed the issue of open ended learning of visual categories, casting a state of the art transfer learning method, Multi-KT [17], into the online learning framework.

This results into an algorithm where the available priors are used in a principled manner to initialize the online learning process. This allows us to exploit the potentiality of the transfer method without paying the computational cost of a batch approach, possibly limited to an initial budget. We call our algorithm TRansfer initialized Online Learning (TROL). We presented two different version of the approach, for each deriving the relative mistake bound, and we showed with extensive experiments on the visual categorization problem the value of our method.

Acknowledgments This work was supported by the EMMA project thanks to the Hasler foundation (www.haslerstiftung.ch).

References

[1] D. Agarwal, B.C. Chen, and P. Elango. Fast online learning through offline initialization for time-sensitive recommendation. In *Proceedings of the International conference on Knowledge discovery and data mining (KDD)*, 2010.

[2] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[3] N. Cesa-Bianchi and G. Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006.

[4] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585, 2006.

[5] J. S. de la Cruz, D. Kulić, and W. Owen. Online incremental learning of inverse dynamics incorporating prior knowledge. In *Proceedings of the International conference on Autonomous and intelligent systems (AIS)*, 2011.

[6] Li Fei-Fei, R. Fergus, and P. Perona. A bayesian approach to unsupervised one-shot learning of object categories. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2003.

[7] P. Gehler and S. Nowozin. Let the kernel figure it out: Principled learning of preprocessing for kernel classifiers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

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

[9] V. Jain and E. Learned-Miller. Online domain adaptation of a pre-trained cascade of classifiers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[10] F. Orabona, C. Castellini, B. Caputo, E. Fiorilla, and G. Sandini. Model adaptation with least-squares SVM for hand prosthetics. In *Proceedings of ICRA - International Conference on Robotics and Automation*, pages 2897–2903, 2009.

[11] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, October 2010.

[12] L. Rosasco, E. De Vito, A. Caponnetto, M. Piana, and A. Verri. Are loss functions all the same? *Neural Computation*, 16(5):1063–1076, 2004.

[13] A. Saha, P. Rai, H. Daumé, S. Venkatasubramanian, and S. L. DuVall. Active supervised domain adaptation. In *Proceedings of the European conference on Machine learning and knowledge discovery in databases (ECML PKDD)*, 2011.

[14] X. Shi, W. Fan, and J. Ren. Actively transfer domain knowledge. In *Proceedings of the European conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD)*, 2008.

[15] E. Soria, J. Martin, R. Magdalena, M. Martinez, and A. Serrano. *Handbook of Research on Machine Learning Applications*, chapter L. Torrey and J. Shavlik, Transfer Learning. IGI Global, 2009.

[16] J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vanderwalle. *Least Squares Support Vector Machines*. World Scientific, 2002.

[17] T. Tommasi, F. Orabona, and B. Caputo. Safety in numbers: Learning categories from few examples with multi model knowledge transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[18] P. Zhao and S. C. H. Hoi. OTL: A Framework of Online Transfer Learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2010.