Safety in Numbers: Learning Categories from Few Examples with Multi Model Knowledge Transfer

Tatiana Tommasi[1,2] , Francesco Orabona[3] , Barbara Caputo[1]
[1]Idiap Research Institute, Martigny CH
[2]Ecole Polytechnique Federale de Lausanne, EPFL, Lausanne CH
[3]DSI, Universita' degli Studi di Milano, Milan IT
ttommasi@idiap.ch, orabona@dsi.unimi.it, bcaputo@idiap.ch

Abstract

Learning object categories from small samples is a challenging problem, where machine learning tools can in general provide very few guarantees. Exploiting prior knowledge may be useful to reproduce the human capability of recognizing objects even from only one single view. This paper presents an SVM-based model adaptation algorithm able to select and weight appropriately prior knowledge coming from different categories. The method relies on the solution of a convex optimization problem which ensures to have the minimal leave-one-out error on the training set. Experiments on a subset of the Caltech-256 database show that the proposed method produces better results than both choosing one single prior model, and transferring from all previous experience in a flat uninformative way.

1. Introduction

The ability to learn from few samples is a hallmark of human intelligence. We rapidly and reliably learn many kinds of regularities and this enables us to make inductive inference even from only small amount of data [1].

Although current state of the art categorization methods reach impressive results on difficult datasets [6], they don't handle well small training sets. Without additional information, learning from few examples always reduces to an ill-posed optimization problem. A possible solution is exploiting prior knowledge, a strategy known in the literature as *learning to learn, knowledge transfer* or *transfer learning*. The basic intuition is that, if a system has already learned k categories, learning the $k + 1$ should be easier, even from one or few training samples [21]. Besides boosting the learning process, knowledge transfer can give three other advantages ([19], see Figure 1): *(1) higher start:* the initial performance is higher (one-shot learning); *(2) higher slope:* performance grows faster, and *(3) higher asymptote:* the final performance is better.

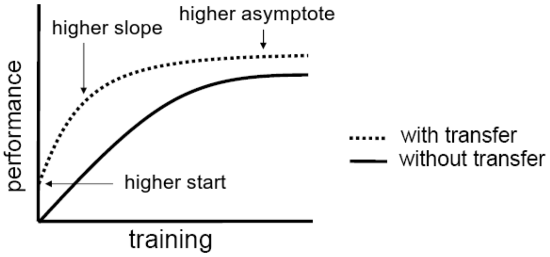


Figure 1. Three ways in which transfer might improve learning [19].

The contribution of this paper is a method for learning object categories from few examples. We focus on three key issues for knowledge transfer: *how to transfer, what to transfer* and *when to transfer* [16]. We propose a discriminative method based on Least Square Support Vector Machine (LS-SVM) [20] (*how to transfer*) that learns the new class through adaptation. We define the prior knowledge as the hyperplanes of the classifiers $\mathbf{w}'_\mathbf{j}$, $j = 1, \cdots, k$ of the k classes already learned (*what to transfer*). Hence knowledge transfer is equivalent to constrain the hyperplanes $\mathbf{w}$ of the $k + 1$ new category to be close to those of a subset of the k classes. This strategy is in between the choice of transferring acritically from all previously learned models [7] and transferring from one single model [22]. We learn the sub-set of classes from where to transfer, and how much to transfer from each of them, via the Leave-One-Out (LOO) error on the training set. Determining how much to transfer helps avoiding negative transfer. Therefore, in case of non-informative prior knowledge, transfer might be disregarded completely (*when to transfer*).

Experiments on various subsets of the Caltech-256 [12] database show that our approach consistently reproduces the curve depicted in Figure 1 with a higher start, and higher slope compared to what is obtained by not exploiting prior knowledge, and to current state of the art knowledge tran-

978-1-4244-6985-7/10/$26.00 ©2010 IEEE

sfer approaches [22, 7]. Furthermore, when the number k of known classes grows, our algorithm presents a one-shot learning behaviour.

The rest of the paper is organized as follows: we give an overview of previous work in Section 2. Section 3 describes LS-SVM and the knowledge transfer algorithm in [22], on which we build. Section 4 describes our new algorithm and discusses its properties. Experimental results are reported in Section 5. We conclude with an overall discussion and pointing out possible avenues for future research.

2. Related Work

Several authors addressed in the past the issues of what, how and when to transfer. We review below the most prominent approaches.

What to Transfer. We can find three answers to this question in the literature (see [16] for a survey). The first is the *instance-transfer approach*: although the source domain data cannot be reused directly, there are certain parts of them that can still be considered together with a few labelled data in the target domain. A second solution is defined by *transferring feature representations*. It means learning a common feature structure, *e.g.* a kernel in SVM-based approaches, from different domains that can bridge related tasks. The third strategy can be described as *parameter-transfer approach*. It assumes that the source task and the target tasks share some parameters of their model or priors.

How to Transfer. Wu and Dietterich transferred source training examples either as support vectors or as constraints (or both) and demonstrated improved image classification by SVMs [24]. Fei-Fei *et al.* [7] proposed a Bayesian transfer learning approach for object recognition that learns a common prior over visual classifier parameters. Zweig and Weinshall [25] investigated transfer learning with a method based on combining object classifiers from different hierarchical levels into a single classifier. Using discriminative (maximum margin) object models, Fink [9] developed a method that learns distance metrics from related problems. Quattoni *et al.* [17] proposed to use knowledge transfer in an unsupervised setting learning a representation based on kernel distances to available unlabelled data.

When to Transfer. Works focusing on when to transfer evaluate the limit of transfer learning power. Rosenstain *et al.* [18] showed empirically that if two tasks are dissimilar, then the transferring hurts the performance on the target task. Ideally, a transfer method should produce positive transfer between appropriately related tasks while avoiding negative transfer when the tasks are not a good match. However it might be easier to avoid negative transfer if, given multiple source tasks, one transfers from several or all of them.

Research on knowledge transfer is still in its infancy, especially applied to object recognition. Although there are many publication in this area, given the slightly different tasks defined in each paper, none of them compare against the others. Moreover there is not an official testbed database, nor a standard experimental setup. In this work we propose a reproducible experimental setting that can be used in the future to test new knowledge transfer algorithms and we benchmark our algorithm against two other methods in literature [22, 7]. We address three open problems of [8]: (1) the possibility that a sophisticated multimodal prior is beneficial in learning; (2) if it is easier to learn new categories which are similar to some of the “prior” categories; (3) if exist another point of view besides the Bayesian one that allows to incorporate prior knowledge. We present a discriminative method which exploits a combination of multiple visual features and selects automatically the most useful prior knowledge models to use when learning a new category.

3. Problem Statement

Consider the following scenario. We have k visual categories and a classifier trained to distinguish each of them from background. This corresponds to define k functions $f_j(\mathbf{x}) \rightarrow \{1, -1\}, j = 1, ..., k$, such that the image $\mathbf{x}$ is assigned to the j^{th} category if and only if $f_j(\mathbf{x}) = 1$. Now suppose that we want to learn a new $k + 1$ category from just one or few instances, plus some background examples. To obtain f_{k+1} we can train using only the available data, or we can take advantage of what already learned. In the following we briefly review the LS-SVM theory and how it can be used in a model adaptation framework [15]. We review how this approach can be formulated to derive a knowledge transfer algorithm that exploits prior knowledge from *only one* of the k classes [22] (Section 3.1). The contribution of this paper is how to extend this method to exploit *all* the suitable prior knowledge. The used strategy is presented in Section 4.

3.1. LS-SVM Adaptation Method: learning from small samples

Suppose to have a binary problem and a set of l samples $\{\mathbf{x}_i, y_i\}_{i=1}^{l}$, where $\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^d$ is an input vector describing the i^{th} sample and $y_i \in \mathcal{Y} = \{-1, 1\}$ is its label. We want to learn a linear function $f(\mathbf{x}) = \mathbf{w} \cdot \phi(\mathbf{x}) + b$ which assigns the correct label to an unseen test sample $\mathbf{x}$. $\phi(\mathbf{x})$ is used to map the input samples to a high dimensional feature space, induced by a kernel function $K(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x}) \cdot \phi(\mathbf{x}')$ [5]. In LS-SVM the model parameters $(\mathbf{w}, b)$ are found by solving the following optimisation problem:

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{2}\sum_{i=1}^{l}[y_i - \mathbf{w} \cdot \phi(\mathbf{x}_i) - b]^2 \,. \qquad (1)$$

It can be shown [20] that the optimal $\mathbf{w}$ is expressed by $\mathbf{w} = \sum_{i=1}^{l} \alpha_i \phi(\mathbf{x}_i)$, and $(\boldsymbol{\alpha}, b)$ are found solving

$$\begin{bmatrix} \mathbf{K} + \frac{1}{C}\mathbf{I} & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ b \end{bmatrix} = \begin{bmatrix} \mathbf{y} \\ 0 \end{bmatrix}, \tag{2}$$

where $\mathbf{K}$ is the kernel matrix. Let us call $\mathbf{G}$ the first term in left-hand side of (2). The least-square optimisation problem can be solved by simply inverting $\mathbf{G}$. Another advantage of the LS-SVM formulation is that it gives the possibility to write the LOO error in closed form [4]. The LOO error is an unbiased estimator of the classifier generalization error and can be used for model selection [4].

Slightly changing the classical LS-SVM regularization term, it is possible to define a learning method based on adaptation [15]. The idea is to constrain a new model to be close to one of a set of pre-trained models:

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w} - \beta\mathbf{w}'\|^2 + \frac{C}{2}\sum_{i=1}^{l}[y_i - \mathbf{w}\cdot\phi(\mathbf{x}_i) - b]^2, \tag{3}$$

where $\mathbf{w}'$ is the parameter describing the old model and β is a scaling factor in $(0, 1)$ necessary to control the degree to which the new model is close to the old one. The LOO error in the modified formulation is:

$$r_i^{(-i)} = \frac{\alpha_i}{\mathbf{G}_{ii}^{-1}} - \beta\frac{\alpha'_i}{\mathbf{G}_{ii}^{-1}}, \tag{4}$$

where $\alpha'_i = \mathbf{G}_{(-i)}^{-1}[\hat{y}_1, \ldots, \hat{y}_{i-1}, \hat{y}_{i+1}, \ldots, \hat{y}_l, 0]^T$, $\mathbf{G}_{(-i)}$ is the matrix obtained when the i^{th} sample is omitted in $\mathbf{G}$ and $\hat{y}_i = (\mathbf{w}'\cdot\phi(\mathbf{x}_i))$, *i.e.* $\hat{y}_i$ is the prediction of the old model on the i^{th} sample. $r_i^{(-i)}$ is then used to obtain an estimate of the Weighted Error Rate (WER) [4]:

$$WER = \sum_{i=1}^{l}\zeta_i\Psi\{y_i r_i^{(-i)} - 1\} \tag{5}$$

$$\text{with} \quad \Psi\{z\} = \frac{1}{1 + exp\{-10 * z\}} \tag{6}$$

$$\text{and} \quad \zeta_i = \begin{cases} \frac{l}{2l^+} & \text{if } y_i = +1 \\ \frac{l}{2l^-} & \text{if } y_i = -1\,. \end{cases} \tag{7}$$

Here l^+ and l^- represent the number of positive and negative examples respectively. Introducing the weighting factors ζ_i is asymptotically equivalent to re-sampling the data so that object and non-object examples are balanced [4]. Hence, without explicitly running cross validation experiments, the best learning parameters which maximise the LS-SVM model generalisation performance can be found as those minimising WER. Since $r_i^{(-i)}$ depends on β, for each known model it is possible to find the best β producing the lowest WER. Then, comparing all the criterion errors, the lowest one will identify the best prior knowledge model to use for adaptation.

To further increase robustness to unbalanced distributions of the data, the model parameters $(\mathbf{w}, b)$ can be found via minimisation of a regularised weighted least-square loss function [20]:

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{2}\sum_{i=1}^{l}\zeta_i[y_i - \mathbf{w}\cdot\phi(\mathbf{x}_i) - b]^2. \tag{8}$$

This introduces just a small variation in the LS-SVM solution: the optimal model parameters $(\boldsymbol{\alpha}, b)$ are defined by a modified system of linear equations [4]:

$$\begin{bmatrix} \mathbf{K} + \frac{1}{C}\mathbf{W} & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ b \end{bmatrix} = \begin{bmatrix} \mathbf{y} \\ 0 \end{bmatrix}, \tag{9}$$

where $\mathbf{W} = diag\{\zeta_1^{-1}, \zeta_2^{-1}, \ldots, \zeta_l^{-1}\}$ and ζ_i are defined as in (7). Hence the model adaptation method changes to its weighted formulation [22]:

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w} - \beta\mathbf{w}'\|^2 + \frac{C}{2}\sum_{i=1}^{l}\zeta_i[y_i - \mathbf{w}\cdot\phi(\mathbf{x}_i) - b]^2\,. \tag{10}$$

In this way the weighting factors ζ_i take into account that the proportion of positive and negative examples in the training data are known not to be representative of the operational class frequencies. In particular they help to balance the contribution of the sets of positive and negative examples to the data misfit term [22].

Experiments show that this method is able to learn new visual categories from few examples. However, the algorithm can choose only one prior known model. As we will show in Section 5, this is not always the best solution. Moreover this approach can suffer for instability in time, *i.e.* when the number of training images increases.

4. Multi Model Knowledge Transfer

Consider the following situation. Suppose to be given the task to learn from few examples the class motorbike, having already learned the categories bicycle, car, dog and cat. We would expect to achieve better results by transferring from bicycle *and* car, rather than transferring from bicycle *or* car. Also, we would expect better results compared to transferring equally from *all* known categories, as cat and dog might induce negative transfer.

This kind of scenario motivates us to design a knowledge transfer algorithm able to find autonomously the best subset of known models from where to transfer. In the rest of the Section we define the new model (Section 4.1) and we discuss its properties (Section 4.2).

4.1. Multi Model Knowledge Transfer: Definition

We start from Equation (10) and we rewrite it substituting the single coefficient β with a vector $\boldsymbol{\beta}$ containing as many elements as the number of prior models, k:

$$\min_{\mathbf{w},b} \frac{1}{2}\left\|\mathbf{w}-\sum_{j=1}^{k}\beta_j\mathbf{w}'_{\mathbf{j}}\right\|^2+\frac{C}{2}\sum_{i=1}^{l}\zeta_i(y_i-\mathbf{w}\cdot\phi(\mathbf{x}_i)-b)^2 \,. \tag{11}$$

Here $\boldsymbol{\beta}$ has to be chosen in the unitary ball, *i.e.* $\|\boldsymbol{\beta}\|_2 \leq 1$. It is similar to the regularization term used in LS-SVM in Equation (1), and it is a natural generalization of the original constraint $0 \leq \beta \leq 1$. This term is necessary to avoid overfitting problems. They can happen when the number of known models is large compared to the number of training samples. With this new formulation the optimal solution is

$$\mathbf{w}=\sum_{j=1}^{k}\beta_j\mathbf{w'}_j+\sum_{i=1}^{l}\alpha_i\phi(\mathbf{x}_i) \,. \tag{12}$$

Hence $\mathbf{w}$ is expressed as a weighted sum of the pre-trained models scaled by the parameters β_j, plus the new model built on the incoming training data.

To find the optimal $\boldsymbol{\beta}$ we use again the possibility of LS-SVM to write the LOO error in closed form:

$$r_i^{(-i)}=y_i-\tilde{y}_i=\frac{\alpha_i}{\mathbf{G}^{-1}_{ii}}-\sum_{j=1}^{k}\beta_j\frac{\alpha'_{i(j)}}{\mathbf{G}^{-1}_{ii}}, \tag{13}$$

where $\alpha'_{i(j)}=\mathbf{G}^{-1}_{(-i)}[\hat{y}^j_1,\ldots,\hat{y}^j_{i-1},\hat{y}^j_{i+1},\ldots,\hat{y}^j_l,0]^T$, $\hat{y}^j_i=(\mathbf{w}'_{\mathbf{j}}\cdot\phi(\mathbf{x}_i))$ and $\tilde{y}_i$ are the LOO predictions. By multiplying by y_i we obtain:

$$y_i\tilde{y}_i=1-y_i\left(\frac{\alpha_i}{\mathbf{G}^{-1}_{ii}}-\sum_{j=1}^{k}\beta_j\frac{\alpha'_{i(j)}}{\mathbf{G}^{-1}_{ii}}\right) \,. \tag{14}$$

The best values of β_j are those minimizing the LOO error, *i.e.* the values producing positive values for $y_i\tilde{y}_i$, for each i. However minimizing directly the sign of those quantities would result in a non-convex formulation with many local minima. We propose instead the following loss function:

$$\begin{aligned} loss(y_i,\tilde{y}_i) &= \zeta_i \max\left[1-y_i\tilde{y}_i,0\right] \\ &= \max\left[y_i\zeta_i\left(\frac{\alpha_i}{\mathbf{G}^{-1}_{ii}}-\sum_{j=1}^{k}\beta_j\frac{\alpha'_{i(j)}}{\mathbf{G}^{-1}_{ii}}\right),0\right] \,. \end{aligned} \tag{15}$$

This loss function is similar to the hinge loss used in Support Vector Machines. It is a convex upper bound to the LOO misclassification loss and favours solution in which $\tilde{y}_i$ has a value of 1, beside having the same sign of y_i. Moreover it has a smoothing effect, similar to the function in (6). Finally, the objective function is:

$$J=\sum_{i=1}^{l}loss(y_i,\tilde{y}_i) \quad \text{s.t.} \quad \|\boldsymbol{\beta}\|_2\leq 1 \,. \tag{16}$$

Notice that this formulation is equivalent to the more common optimization problem $(1/2)\|\boldsymbol{\beta}\|_2^2+CJ$ for a proper choice of C [5]. By minimizing J we can find the best values of β_j to weight the known prior models in the transfer learning process. The scaling factors ζ_i are introduced in the loss function to take care of the data unbalance between positive and negative samples in the training set, as in [22].

We implement the optimization process using a simple projected sub-gradient descent algorithm, where at each iteration $\boldsymbol{\beta}$ is projected onto the l_2-sphere, $\|\boldsymbol{\beta}\|_2 \leq 1$.

4.2. Multi Model Knowledge Transfer: Properties

The main advantage of our approach is the ability to transfer from multiple prior model, instead of choosing just one. At the same time, the knowledge is not transfered in a flat, uninformative way, but we evaluate the importance of each model and their interaction. The loss used is convex and the constraint in (16) is convex too, hence the minimizer of (16) is unique. This is opposed to the formulation proposed in [22], where (7) is non-convex. This means that the algorithm in [22] can have many local minima.

An important property of this new formulation is also its "stability". Stability here means that the behaviour of the algorithm does not change much if a point is removed or added. This notion is closely related to the LOO error, which is exactly calculated measuring the performance of the model every time a point is removed. Recent works have shown that a stable algorithm has a better generalization ability [3]. The algorithm in [22] can choose only one model at each time step, to be used to transfer knowledge. This means that everytime the algorithm "changes its mind", *i.e.* it chooses a different prior model on two consecutive time steps, the behaviour of the algorithm will change completely. On the other hand, our method selects more than one prior model at each time step, so we expect that differences between steps in the vector $\boldsymbol{\beta}$ will be small. The regularization is also important in this sense [3]. In Section 5.2 we show empirically that this is true.

From a computational point of view the current algorithm's runtime is $\mathcal{O}(l^3+kl^2)$, with l the number of training samples (of the order of 10 images) and k the number of known prior models. The first term is related to inverting $\mathbf{G}$, while the second term is the computational complexity of (13). We match the complexity of a plain SVM, which in the worst case is known to be $\mathcal{O}(l^3)$ [13], and is the standard out-of-the-shelf classification method commonly used on datasets with more than 10^3 images. The computational complexity of each step of the projected sub-gradient descent is $\mathcal{O}(kl)$ and it is extremely fast. For instance, our MATLAB implementation takes just half a second with $l = 12$ and $k = 3$.

5. Experiments

We present here three sets of experiments designed to illustrate how our algorithm performs (a) when the prior knowledge is related/unrelated to the new class (Section 5.2); (b) when prior knowledge increases (Section 5.3); (c) compared to the current state of the art [7, 8] (Section 5.4). We first describe the experimental setup (Section 5.1) and then we report our findings in the three scenarios described above. The algorithms presented here were implemented in MATLAB. The code for the Multi Model Knowledge Transfer method and all the scripts used for the experiments are available online[1].

5.1. Experimental Setting

Our working assumption is to have k category models stored in memory, built using LS-SVM. We used the Gaussian kernel $K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma\|\mathbf{x} - \mathbf{x}'\|^2)$ for all our experiments; the parameters C and γ were chosen by cross-validation. When the new $k + 1$ category comes, the system starts learning.

All experiments are run on subsets of the Caltech-256 database [12]. We selected in total 41 classes + background class, obtaining a data set with a fair amount of clutter and scale variation. We didn't perform any image selection or preprocessing. The training data consists of m images from the background dataset and an increasing number of instances of the new category from 1 to m. The test set consists of 100 images, half from the background and half from the new category. Images are chosen randomly by splitting each class into two disjoint sets: m training images are drawn randomly for the first, a set of 50 are taken from the second. As we focus on learning from small samples, we varied m from 1 to 6, repeating the experiments 10 times for each value and using different sets of training and test images. To get a reliable estimate of the performance on all the categories, we used a leave-one-class-out approach, considering in turn each class for adaptive learning and using all the rest as prior knowledge.

We used the pre-computed features of [10] which the authors made available on their website[2]. Specifically, we used four different image descriptors: PHOG Shape Descriptors [2], Appearance Descriptors [14], Region Covariance [23] and Local Binary Patterns. All of the image descriptors were computed in a spatial pyramid, we considered just the first level (*i.e.* informations extracted from the whole image) and combined the features using the average kernel.

In the following we will compare the performance of our Multi Model Knowledge Transfer aglorithm (*Multi-KT*) to that obtained with a flat average mixture of prior knowledge (*Average-KT*) and to the method presented in [22] that we call here *Single-KT*. We also benchmark all the results against *No Adapt*. This corresponds to learn from scratch using weighted-LS-SVM, *i.e.* solving the optimization problem in Equation (10) with $\beta = 0$. The significance of the comparisons are evaluated through the sign test [11].

5.2. Related/Unrelated Prior Knowledge

In the first set of experiments we considered different groups of related and unrelated categories. The goal is to study how *Multi-KT* chooses the reliable prior knowledge, and its impact on performance.

Related Classes. We considered two sets of 6 classes belonging respectively to the Caltech-256 general classes "transportation, ground, motorized" (bulldozer, firetruck, motorbikes, schoolbus, snowmobile, car-side) and "food edibles" (cake, hamburger, hot-dog, ice-cream-cone, spaghetti, sushi). Figure 2(a)-(d) show the respective classification results. In both cases we see that all KT algorithms obtain an impressive advantage over starting from scratch. As Figure 2(c)-(f) shows, *Multi-KT* performs clearly better than *Single-KT*, with ($p < 0.02$) for less than four images in both cases. This confirms the intuition that it pays off to transfer from multiple sources, as opposed to one, when they all bring useful information. There is no significant difference in accuracy between *Multi-KT* and *Average-KT* (Figure 2(b)-(e)). This suggests that, when all prior knowledge is useful, learning the weights does not give a real advantage over a flat average.

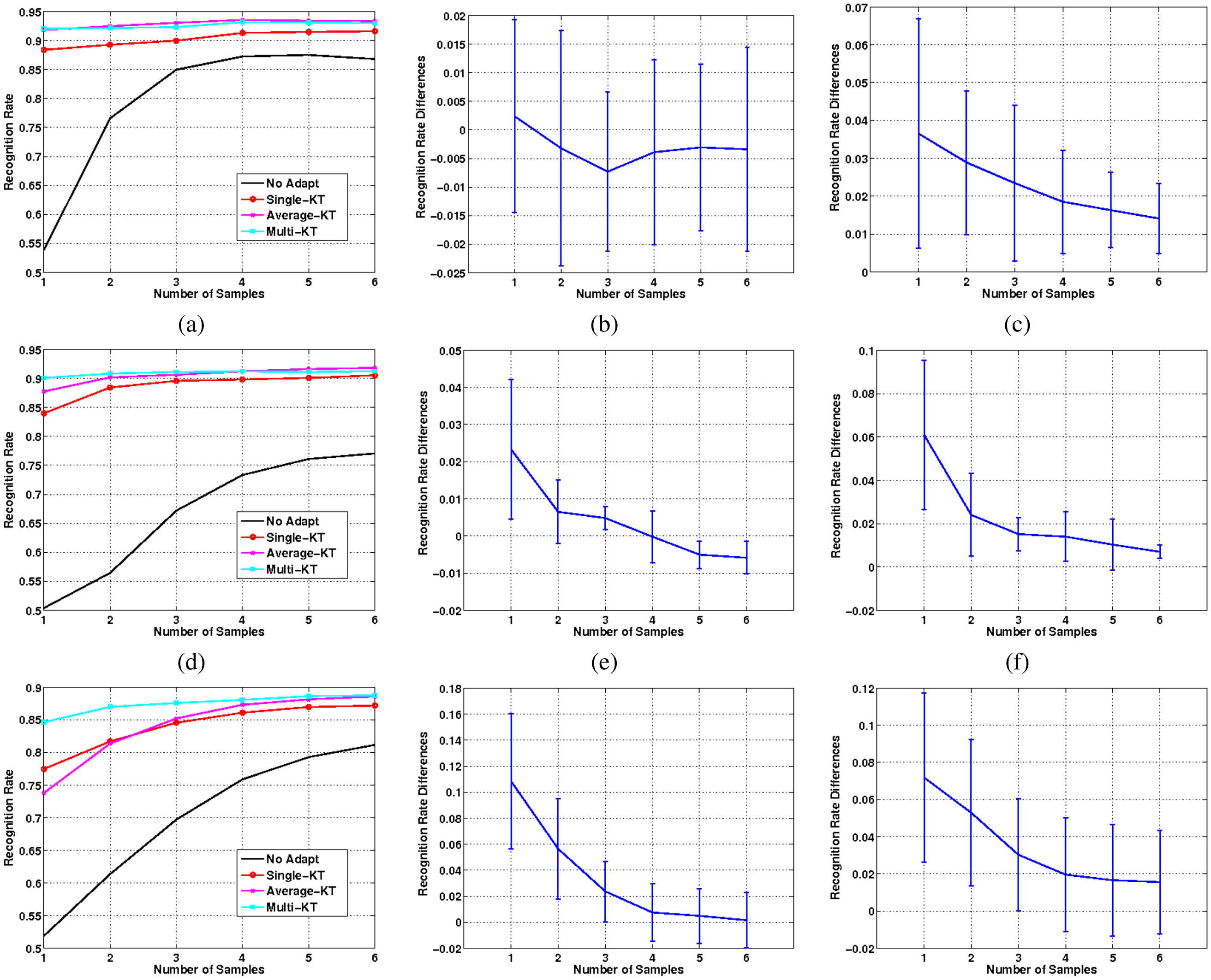
Mixed Classes. To consider what happens in a more confused situation, we selected the following 10 classes: dog, horse, zebra, helicopter, fighter-jet, motorbikes, car-side, dolphin, goose and cactus. The classification results in Figure 2(g) show that here *Multi-KT* performs better both than *Average-KT* and *Single-KT* (Figure 2(h)-(i), in both cases $p < 0.02$ for less than four images). This experiment illustrates very clearly the power of our approach: when the prior knowledge is partially related to the new class, transferring from only one model does not exploit fully previous experience. At the same time, using acritically all the prior knowledge induces partial negative transfer behaviours, that affect the overall performance. Notice that the situation of knowledge transfer from a mixture of related and unrelated classes is the most common.

We can also compare *Multi-KT* to *Single-KT* in terms of stability. Let us consider the unique β used by *Single-KT* as an element of the $\boldsymbol{\beta}$ vector where all the remaining elements are zero. There are 6 steps in time corresponding to a new

---

[1] `http://www.idiap.ch/~ttommasi/source_code.html`

[2] `http://www.vision.ee.ethz.ch/~pgehler/projects/iccv09/`



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 2. (a-d-g) Classification performance as a function of the number of object training images, when learning respectively one out of six related categories "transportation, ground, motorized", "food, edibles", and one out of ten mixed categories. The results shown correspond to average recognition rate over the categories, considering each class-out experiment repeated 10 times; (b-e-h) average difference in classification performance ± stand. dev., obtained by *Multi-KT* with respect to *Average-KT*; (c-f-i) average difference in classification performance ± stand. dev., obtained by *Multi-KT* with respect to *Single-KT*.

positive sample entering the training set. For each couple of subsequent steps we calculated the difference between the obtained $\boldsymbol{\beta}$ vectors of *Single-KT*. We did the same with the $\boldsymbol{\beta}$ vectors produced by the *Multi-KT* algorithm. Figure 3 shows the average norm of these differences. It is evident that choosing a combination of the prior known models for transfer learning is more stable (lower average variations in the $\boldsymbol{\beta}$ vectors) than relying on just a single known category.

5.3. Increasing Prior Knowledge

Here we studied how performance varies when the number of known category grows. We are especially interested in how *Multi-KT* behaves when learning from a single pos-

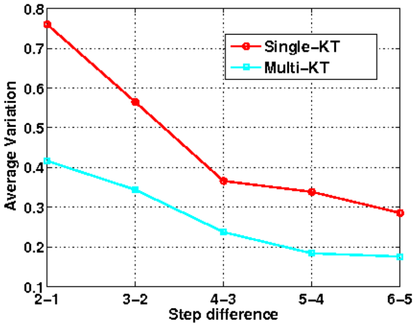
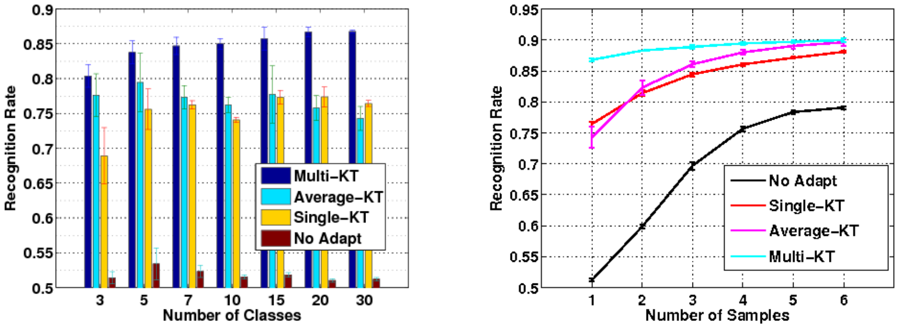


Figure 3. Norm of the differences between two $\boldsymbol{\beta}$ vectors correspondent to two subsequent steps in time. The norms are averaged both on the classes and on the splits. These results are obtained considering 10 randomly chosen classes.



(a) (b)

Figure 4. (a) One-shot learning performance of *Multi-KT*, *Average-KT* and *Single-KT* respect to *No Adapt* when varying the number of prior known categories; (b) classification performance as a function of the number of training images when learning on 30 object categories. The results correspond to average recognition rate over the 30 categories (each class out repeated 10 times), we run this experiment 3 times, the error bars denote ± standard deviation.

itive image (one-shot learning). We selected 30 classes[3], extracting 3 visually related classes from 10 general categories of Caltech-256. We run six set of experiments, considering 3/5/7/10/15/20 categories plus a final one with all the 30 categories. We first extracted three categories through random selection and then we went on adding new ones till covering the whole 30 class dataset. We repeated the experiments three times: Figure 4(a) shows the average recognition rate and the corresponding standard deviations when training only on one object image. We expect that the overall performance will increase along with the number of stored models, since there is a larger probability to have stored useful prior knowledge. This intuition is confirmed by the increasing accuracy of the one-shot learning for *Multi-KT*. *Average-KT* shows a decreasing behaviour, indicating that as the prior knowledge grows, the number of unrelated classes in memory usually outnumbers the related one. The performance of *Single-KT* is more or less constant except for an evident jump in performance passing from 3 to 5 categories. Finally, Figure 4(b) shows the average classification results in case of 30 categories. It is evident here that, when learning from few samples (≤ 4), *Multi-KT* outperforms both *Average-KT* and *Single-KT*. These results, jointly with those reported in the previous section, make us conclude that *Multi-KT* is the most effective knowledge transfer algorithm, compared to *Average-KT* and *Single-KT*.

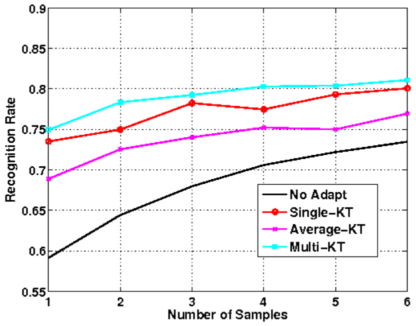


Figure 5. Classification performance as a function of the number of object training images, when learning one out of four unrelated categories. The results showed correspond to average recognition rate over the four categories, considering each class-out experiment repeated 10 times.

Algorithm	Error Rate (%) on 5 pos. images	Best Rec. Rate (%) on 1 pos. image	Remarks
Multi-KT	8-29	airplanes: 90.8	+6 backgr. images
Single-KT [22]	10-29	airplanes: 88.1	+6 backgr. images
[7]	8-22	faces: 82.0	

Table 1. Comparison between our *Multi-KT* algorithm, *Single-KT* [22], and the Bayesian One-Shot learning method presented in [7]. Since both *Multi-KT* and *Single-KT* are a discriminative approaches, besides the positive samples we need few background images in the training set.

5.4. Comparison with previous work

The most famous one-shot learning algorithm in the computer vision literature is [7, 8], where the authors extract a “general knowledge” from previously learned categories. Their approach makes no assumptions on the reliability of prior knowledge, which is always considered as an average of all the known classes. To compare against this method, we repeated the four classes experiment presented in [7]. Unfortunately it was not possible to reproduce exactly their experimental setting, as the features used are no more available[4], and the algorithm was not publicly released. We opted therefore for benchmarking our results against those reported in Table 1 in [7].

---

[3] “transportation, ground, motorized”:car-side, fire-truck, motorbike; “animal,land”: dog, horse, zebra; “animal,water”: goldfish, dolphin, killer-whale; “transportation, water”: canoe, kayak, speed-boat; “music, stringed”: electric-guitar, harp, mandolin; “food, containers”: beer-mug, coffee-mug, teapot; “transportation, air”: airplanes, helicopter, fighter-jet; “animals, air”’: duck, goose, swan; “plants”: bonsai, cactus, fern; “structures, buildings”: light-house, windmill, smokestack.

[4] L. Fei Fei, personal communication.

We considered the classes faces, motorbikes, leopards (originally spotted-cats) and airplanes. The average recognition rate over the categories as a function of the number of object training images is shown in Figure 5. Table 1 compares the results of *Multi-KT* and *Single-KT* to that reported in [7] considering also the best one-shot result per class. This analysis confirms that our method performs better than *Single-KT*, and it obtains results comparable to [7].

6. Conclusions

We presented an SVM-based method for learning object categories from few examples. The algorithm transfers prior knowledge selecting a subset of the known classes and weighting them appropriately. It decides automatically from where and how much to transfer, adapting the old models to the incoming data and solving a convex optimization problem which minimizes an estimate of the generalization error. Experiments show that it outperforms both the results obtained in [22] and those produced using an average of all the previous experience. This last choice can induce negative transfer, in particular when the number of known category increases. On the contrary, when prior knowledge grows our method shows a one-shot learning behaviour. By using the features provided in [10] and making available our code we are offering to the community a reproducible experimental setting that can be used in the future to test new knowledge transfer algorithms. By using several features we also showed that the behaviour of the method is not affected by the feature's choice. Future work will investigate ways to reduce the computational complexity of the algorithm for large number of known categories and analyze its asymptotical behaviour when the number of training samples increases.

7. Acknowledgments

This work was supported by the EU project DIRAC (FP6-0027787), by the EMMA project thanks to the Hasler foundation (www.haslerstiftung.ch) (T.T. & B.C), and by the PASCAL Network of Excellence (F.O.). We are thankful to Nicolò Cesa-Bianchi for his helpful comments and suggestions.

References

[1] I. Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 1987.

[2] A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. In *CIVR*, 2007.

[3] O. Bousquet and A. Elisseeff. Stability and generalization. *JMLR*, 2:499–526, Mar 2002.

[4] G. Cawley. Leave-one-out cross-validation based model selection criteria for weighted LS-SVMs. In *IJCNN*, 2006.

[5] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, 2000.

[6] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2008 Results. http://www.pascal-network.org/challenges/VOC/.

[7] L. Fei-Fei, R. Fergus, and P. Perona. A bayesian approach to unsupervised one-shot learning of object categories. In *ICCV*, 2003.

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

[9] M. Fink. Object classification from a single example utilizing class relevance metrics. In *NIPS*, 2004.

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

[11] J. Gibbons. *Nonparametric Statistical Inference*. New York: Marcel Dekker, 1985.

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

[13] D. Hush, P. Kelly, C. Scovel, and I. Steinwart. QP algorithms with guaranteed accuracy and run time for support vector machines. *JMLR*, 2006.

[14] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[15] F. Orabona, C. Castellini, B. Caputo, E. Fiorilla, and G. Sandini. Model adaptation with least-squares SVM for hand prosthetics. In *ICRA*, 2009.

[16] S. J. Pan and Q. Yang. A survey on transfer learning. Technical report, Hong Kong University of Science and Technology, Hong Kong, China, November 2008.

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

[18] M. Rosenstein, Z. Marx, and L. P. Kaelbling. To transfer or not to transfer. In *NIPS Workshop on Transfer Learning*, 2005.

[19] 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.

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

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

[22] 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.

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

[24] P. Wu and T. G. Dietterich. Improving SVM accuracy by training on auxiliary data sources. In *ICML*, 2004.

[25] A. Zweig and D. Weinshall. Exploiting object hierarchy: Combining models from different category levels. In *ICCV*, 2007.