Learning to Learn by Exploiting
Prior Knowledge

Acknowledgments

Working on the PhD has been a wonderful and often overwhelming experience. When I started I was not even suspecting of how much I would have grown up during the following four years. It is hard to say if it has been facing all the difficulties of the topic itself which has been the real learning experience, or fighting against the computation grid, grappling with how to write papers, prepare fancy figures, give talks, stay up until birds start singing and stay focus...These are just few aspects of my adventure but they all contributed to what I am today. Moreover I was so lucky to share all these experiences with people that in one way or another helped me putting the pieces together and making the PhD an unforgettable period.

First of all I'm deeply grateful to my advisor Barbara Caputo. You have been the first to believe in the capacity of the "match girl" to become a doctor! To work with you has been a real pleasure : you have oriented and supported me with care and have always been patient in times of difficulties. Your skill to select and approach compelling research problems and your high scientific standards set an example. Above all, you made me feel a friend which I appreciate from my heart.

Furthermore I am grateful to my thesis director Prof. Hervé Bourlard for carefully following my work from above : interactions with you have always been fruitful.

I would also like to thank my thesis juries Prof. Dario Floreano, Dr. Vittorio Ferrari, Prof. Jim Little and Prof. Jean-Philippe Thiran, for kindly agreeing to read my thesis and for their useful comments.

I have been very privileged to get to know and to collaborate with many other great people. First and foremost Francesco Orabona who has been my machine learning guru, always ready to share his keen intelligence and warm personality with me. Thanks for the countless hours you spent to introduce me the secrets of statistical learning and making sure I do not forget them along the way. More in general I've learned a lot from you, following the idea that there is no reason to be pleased if you can do better in research and in life.

Many thanks go to Prof. Christoph Lampert for supervising me during my internship at IST Austria. Thank you for sharing with me your research vision and experience. I'm also grateful to Prof. Henning Müller who knows me since my first steps in the medical image research world and has always been encouraging, kind and helpful. In addition I'd like to express my gratitude to my colleagues and talented co-authors : Novi Quadrianto, Jie Luo, Claudio Castellini, Arjan Gijsberts, and Fabian Nater.

Abstract

One of the ultimate goals of open ended learning systems is to take advantage of experience to get a future benefit. We can identify two levels in learning. One builds directly over the data : it captures the pattern and regularities which allow for reliable predictions on new samples. The other starts from such an obtained source knowledge and focuses on how to generalize it to new target concepts : this is also known as *learning to learn*. Most of the existing machine learning methods stop at the first level and are able of reliable future decisions only if a large amount of training samples is available. This work is devoted to the second level of learning and focuses on how to *transfer information from prior knowledge*, exploiting it on a new learning problem with possibly scarce labeled data.

We propose several algorithmic solutions by leveraging over prior models or features. One possibility is to constrain any target learning model to be close to the linear combination of several source models. Alternatively the prior knowledge can be used as an expert which judges over the target samples and considers the obtained output as an extra feature descriptor. All the proposed approaches evaluate automatically the relevance of prior knowledge and decide from where and how much to transfer without any need of external supervision or heuristically hand tuned parameters. A thorough experimental analysis shows the effectiveness of the defined methods both in case of interclass transfer and for adaptation across different domains.

The last part of this work is dedicated to moving forward knowledge transfer towards life long learning. We show how to combine transfer and online learning to obtain a method which processes continuously new data guided by information acquired in the past. We also present an approach to exploit the large variety of existing visual data resources every time it is necessary to solve a new situated learning problem. We propose an image representation that decomposes orthogonally into a specific and a generic part. The last one can be used as an un-biased reference knowledge for future learning tasks.

Keywords : transfer learning, visual object recognition, multiclass classification, regression, domain adaptation, multi-kernel learning, online learning, multi-task learning.

Résumé

Un des objectifs des systèmes d'apprentissage est d'utiliser l'expérience passée pour obtenir un bénéfice futur. Nous pouvons identifier deux niveaux d'apprentissage. L'un se construit directement sur les données : il capture les motifs et régularités qui permettent d'effectuer des prédictions fiables sur de nouveaux échantillons. L'autre niveau d'apprentissage part d'une telle connaissance (source) apprise et essaye de la généraliser à de nouveaux concepts (cible) : on appelle cela *apprendre à apprendre*. La plupart des méthodes existantes d'apprentissage automatique s'arrêtent au premier niveau et ne permettent des décisions futures fiables que si l'on a accès à un jeu d'apprentissage conséquent. Ces travaux abordent le deuxième niveau d'apprentissage et se concentrent sur la manière de *transférer de l'information depuis un savoir pré-existant*, l'exploitant pour de nouveaux problèmes d'apprentissage, notamment dans le cas de manque de données étiquetées.

Nous proposons plusieurs solutions algorithmiques en se basant sur des modèles ou des descripteurs existant. Une façon possible de le faire est de contraindre tout modèle d'apprentissage cible à être proche d'une combinaison linéaire des modèles source. Une alternative consiste à utiliser une connaissance *a priori* comme un expert qui juge les échantillons cibles et considérer la sortie de ce processus comme un nouveau descripteur. Toutes les méthodes proposées évaluent automatiquement la pertinence de la connaissance *a priori* et décident d'où et à quel point transférer sans utiliser de supervision extérieure ou de paramètres réglés de manière heuristique. Une analyse expérimentale approfondie montre l'efficacité des méthodes définies à la fois en termes de transfert inter-classes et d'adaptation entre différents domaines.

La dernière partie de ce travail est dédiée à pousser plus loin le transfert de connaissance vers l'apprentissage à long terme. Nous montrons comment combiner le transfert et l'apprentissage en ligne pour obtenir une méthode capable de traiter en continu des nouvelles données qui arrivent en se basant sur l'information acquise dans le passé. Nous présentons également une approche pour exploiter la grande variété de ressources à chaque fois que ce la est nécessaire pour résoudre un nouveau problème d'apprentissage. Nous proposons représentation de l'image qui se décompose de manière orthogonale entre une partie spécifique et une partie générique. Cette dernière peut être utilisée comme un savoir de référence non biaisé pour des tâches d'apprentissage futures.

Acknowledgments

Sommario

Uno degli scopi finali di ogni sistema di apprendimento é di trarre vantaggio dall'esperienza per ottenere un beneficio in futuro. É possibile identificare due livelli di apprendimento. Uno si basa direttamente sui dati disponibili e consiste nel catturare la struttura e le regolarità presenti per assicurare delle predizioni affidabili su nuovi campioni. Il secondo livello parte dalla conoscenza cosí definita e focalizza su come generalizzarla per nuovi concetti: questo processo é anche noto come *apprendere ad apprendere*. La maggior parte dei sistemi di apprendimento artificiale si fermano al primo livello e risultano affidabili solo in caso di una grande quantità di dati di addestramento. Questo lavoro é dedicato al secondo livello di apprendimento e si incentra su come *trasferire informazioni dalla conoscenza pregressa* quando si affronta un nuovo problema nel caso di pochi dati utili.

Proponiamo diversi algoritmi in grado di sfruttare l'esperienza acquisita sotto forma di modelli o di descrittori. Una possibilità consiste nell'imporre che il modello che descrive il nuovo problema sia vicino ad una combinazione lineare dei modelli noti. Alternativamente, la conoscenza a priori puó essere utilizzata come un esperto il cui giudizio sui nuovi dati viene usato come un descrittore. Tutti i metodi proposti valutano automaticamente la rilevanza della conoscenza a priori senza la necessità di supervisione esterna o di parametri scelti in modo euristico.

L'ultima parte di questo lavoro é dedicata a come passare dal trasferimento di conoscenza a un apprendimento continuo a lungo termine. Mostriamo come combinare il trasferimento con l'apprendimento sequenziale e otteniamo un metodo che processa continuamente nuovi dati guidato dalle informazioni giá acquisite in passato. Presentiamo inoltre un approccio che sfrutta la grande varietá di risorse visive esistenti ogni qual volta sia necessario risolvere un nuovo problema. Proponiamo una rappresentazione dei dati che si decompone ortogonalmente in una parte specifica ed una generica. Quest' ultima puó essere usata come riferimento per ogni nuovo compito di apprendimento.

Parole chiave: trasferimento di conoscenza pregressa, riconoscimento visivo di oggetti, classificazione, regressione, adattamento, apprendimento di Kernels multipli, apprendimento continuo, apprendimento simultaneo su molteplici compiti.

Contents

1 Introduction

As human beings, our learning capacity develops progressively in time as we grow. At the age of six, we recognize around 10^4 object categories and we keep learning more throughout our life [16]. Moreover we tend to organize all our knowledge into useful taxonomies: concepts and categories are grouped on the basis of the common properties acquired through our five senses. This intrinsically means that any new concept is not learned in isolation, but considering connections to what is already known, which makes analogical reasoning (the skill of building analogies) one of the cores of human intelligence [68]. This results in practical advantages: it might be easier to learn French if one already knows Italian and English and it might be easier to learn playing chess if one already knows draughts. Even focusing only on visual tasks, we can give several examples of this cognitive ability: have you ever seen a guava or a serval? The guava is a fruit that can be easily recognized if apples and limes are known, and a serval can be described as an animal similar to a leopard with longer legs and lighter body (see Figure 1.1).
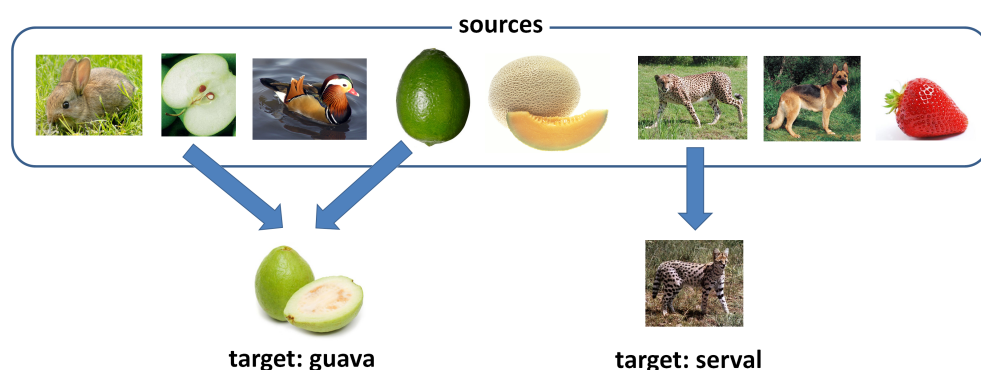
Figure 1.1 – If we already know the appearance of several objects we can use it as reference information when learning something new. Here we give an example considering source knowledge on fruits and animals while *guava* and *serval* are used as target tasks. The first is a fruit which externally looks like a lime, while its inner part is similar to an apple. The second is an African animal, with long legs and a small head in relation to the body. The pattern of its fur is similar to that of a leopard.

In psychology this specific analogical capacity is also known as *transfer learning* and encompasses phenomena ranging from simple (e.g. generalization of conditioned response between familiar and novel stimuli) to extremely complex (e.g. carrying over a solution to a problem in arithmetics to a novel class of problems) behaviors [70]. The degree of generalization between stimuli is governed by their perceptual similarity, while it is hypothesized that more complex processes are mediated by specific cognitive structures. In general this ability makes learning further concepts extremely efficient, allows us to evaluate many kinds of recurrent patterns and regularities, giving us the possibility to make inductive inferences on a new task even with only a small amount of data.

When the learning problem was formalized for artificial intelligent systems, a computer program was said to learn if its performance at a given task was improving with experience [115]. For example, supervised learning addresses the task of approximating an unknown function f with experience in the form of training examples, and the performance is measured by the ratio of correct to incorrect classification. This definition does not consider at all neither the possibility to encounter more than one task, nor their eventual relation or how to induce functions that generalize over all of them. Learning in this more natural scenario, with an improvement in performance both with experience and with the number of tasks has been studied only more recently under the name of *learning to learn* [160]. Towards this goal an artificial learning system should be able to reproduce the human knowledge transfer skills. This implies to define methods able to answer autonomously to questions like how to formalize pre-existent source knowledge and how to evaluate in practice the relation among different tasks.

This work is devoted to the above issues. We propose several algorithmic solutions and we evaluate them considering different applications which ranges from visual object classification to adaptive learning for hand prosthetics. Finally we analyze how to apply knowledge transfer in the more general context of life long learning.

In the following section we describe the main features of the proposed methods and we discuss the contributions of this work. We conclude the chapter with a short outline of the thesis.

1.1   Contribution of this work

The main contribution of this work is the introduction of transfer learning algorithms which exploit automatically prior knowledge. When more than one auxiliary source of information is available, the proposed methods choose which are the most useful for the new task and the degree of adaptation. At the same time they guarantee a performance at least equal to what obtained by learning from scratch on the novel target task in case of unrelated sources.

Specifically we present

a discriminative knowledge transfer method that leverages over prior models [161, 165];

It is based on Least-Square Support Vector Machines [156]; it focuses on binary problems and learns any new class through adaptation by imposing closeness between the target classifier w and a linear combination of the corresponding w'_j already learned on the $j = 1, \ldots, J$ sources. The weights assigned to each prior knowledge in the linear combination are defined by solving a convex optimization problem which minimizes the leave-one-out error on the training set. This is an almost unbiased estimator of the generalization error for a classifier, and minimizing it provides us with a principled solution to choose from where to transfer and how much to rely on each known source.

a discriminative knowledge transfer method that leverages over features [104].
The closeness constraint between the sources and target classifier used in the described adaptive learning method can also be interpreted differently in mathematical terms. Considering each source knowledge as an expert which judges on the new target samples, it assesses the obtained confidence output as extra features. Starting from this view it is possible to enlarge the original method to multiclass problems. We propose to cast this idea in the multi-kernel learning framework, obtaining a final algorithm that solves at the same time the learning problem on the new task and defines from where and how much to transfer with a principled multiclass formulation. This approach also allows us to consider unconstrained prior sources, meaning that they could be originally defined with any learning method and any features.

Moreover, we show

that the proposed methods are largely applicable in several settings [166];
Transfer learning covers the exchange of information across different categories or concepts, while more in general this exchange can happen even between the same entities that has been already learned in the past but under different conditions. This is generally known as domain adaptation. Examples of domain mismatch are the variability in the posture (recorded in terms of biological signals) across multiple subjects performing the same actions, or the difference in point of view, resolution and lighting conditions of object images recorded with different devices and in different environments.

how to move from knowledge transfer to life long learning [167, 168].
In particular we consider two different directions, proposing (1) a method to combine transfer with online learning, limiting the computational burden of the transfer process; (2) a method to integrate transfer, domain adaptation and multi-task learning with the aim to take advantage of several existing visual resources to obtain cross-dataset generalization.

1.2 Outline

The rest of the thesis is organized in three main parts. We start with an overview on the general problem of learning efficiently on a target task by exploiting prior knowledge: chapter 2 reviews existing techniques to overcome the distribution mismatch between different domains and to exploit available sources of information.

Then, in part 1 we introduce a transfer learning algorithm that leverages over prior known models. We describe its theoretical derivation and properties, reporting on several experimental results. More precisely chapter 3 presents the knowledge transfer method for binary problems, while chapter 4 describes its extension to multiclass domain adaptation.

Part 2 (chapter 5) focuses on a feature transfer approach. Its basic structure can be derived directly from the previously defined method, but it presents much more freedom in terms of possible form of source knowledge (feature used and learning method considered) and target tasks (multiclass problems). Thorough experiments show the performance of the proposed approach in different settings.

Finally, part 3 introduces two steps to move forward transfer learning. Chapter 6 shows that it is possible to initialize online learning with knowledge transfer: this helps in identifying in which part of the learning space the correct solution (the best in terms of generalization capacity) should be sought. Chapter 7 focuses on the possibility to have direct access to several pre-collected visual datasets and proposes an image representation that decomposes orthogonally in two subspaces: a part specific to each dataset and a part shared by all of them. This last generic representation can then be used as unbiased reference knowledge for any novel classification task.

The thesis concludes with a summary discussion and possible future direction of research in chapter 8

2 Several Landmarks and Prior Work

In this chapter we review the problem of learning on a target set and exploiting auxiliary sources of information. This can be extremely useful when the training set has few labeled data, but in general, to get this help it is necessary to overcome the distribution mismatch between the source and the target. We describe different aspects of this problem and we give an overview of the related work with particular focus on transfer learning, its advantages and challenges.

2.1 Exploiting the Distribution Mismatch

The main assumption in theoretical models of learning, such as the standard PAC (Probably Approximately Correct [175]) model, is that training instances are drawn according to the same probability distribution as the unseen test examples. This hypothesis permits the estimation of the generalization error and the uniform convergence theory [176] provides basic guarantees on the correctness of future decisions.

However in many real world applications, this assumption does not hold. It often happens that the training data is different from that available for testing, as the two sets are actually drawn from different distributions. This problem arises because labeled data are difficult and expensive to obtain and some times they depend on dynamic factors including time, acquisition device and space, which make them easily outdated.

There are several practical examples of what described untill here. In wifi localization the signal strength depends on many fast evolving parameters. In face recognition systems, training images are often obtained under some set of lighting or occlusion conditions that may change in the test phase. In speech recognition, acoustic models trained on one speaker may be used by another. In natural language processing, part-of-speech taggers, parsers, and documents classifiers are trained on carefully annotated training sets, but then applied on text from different genders or styles.

In general terms, we can have a lot of labeled data on a *source* problem and the need to solve a different *target* problem with few or no labeled data, where source and target present a

distribution mismatch. In this case to reduce the effort of collecting new samples, and at the same time to reduce the lack of robustness issue (risk of overfitting) we can leverage over the existing source knowledge. How to do it and up to which extent it can be useful depends on the relation between source and target. This relation can be empirically evaluated either directly through the performance of a source model on the target, or according to some similarity metrics between the respective probability distributions.

We want to focus on the general problem of exploiting prior knowledge gathered on one or multiple sources when facing a new target despite the distribution mismatch. To formalize it, let's indicate with $X \in \mathcal{X}$ the input variable to a learning system (i.e. an observation) and with $Y \in \mathcal{Y}$ the output variable (i.e. label), where $\mathcal{X}$ and $\mathcal{Y}$ specify respectively the feature and the label space. We use lowercase x and y to denote a specific value of X, Y (i.e. a sample and its label). Furthermore we call *domain* $D = \{\mathcal{X}, P(X)\}$ the pair of feature space and marginal distribution on the data, while a *task* $T = \{\mathcal{Y}, f(X)\}$ is the pair of label space and prediction function, where the last one can be written in probabilistic terms as $P(Y|X)$. Depending on (a) what gives rise to the distribution mismatch in terms of domain and task relations, and (b) if the learning process is symmetric (symultaneous over many tasks) or asymmetric (starts from a source task and then uses the obtained knowledge on a target task), it is possible to consider different aspects of the problem with their respective solutions. We describe them in the following sections and two general schemes with examples are presented in Figures 2.1 and 2.2.

2.1.1 Domain Adaptation

Domain adaptation [73] aims at solving the learning problem on a target domain D^t exploiting information from a source domain D^s, when both the domains and the corresponding tasks T^s, T^t are not the same. In particular, the tasks have identical label sets $\mathcal{Y}^s = \mathcal{Y}^t$ but with slightly different conditional distributions $P^s(Y|X) \sim P^t(Y|X)$. The domains are different in terms of marginal data distribution $P^s(X) \neq P^t(X)$, and/or in feature spaces $\mathcal{X}^s \neq \mathcal{X}^t$.

Domain adaptation has been studied in two main settings: one is the *semi-supervised* case, where the target presents few labeled data, while the other is the *unsupervised* case that considers only unlabeled examples for the target. In both cases, the source set is generally rich in labeled samples.

Many techniques for the *semi-supervised* domain adaptation problem have been developed for text classification. A common approach is to treat the source domain as prior knowledge and to estimate the target domain model parameters under such prior distribution. This solution was developed within the context of generative classifiers (maximum entropy models [31], estimate of the Bayesian divergence [97]), but the idea is more general and was also extended to discriminative methods. By modifying the regularization term it is possible to drive the model trained on the target to prefer parameters similar to that already defined for the source [181]. Other techniques aim at bridging the gap between the source and target distribution by
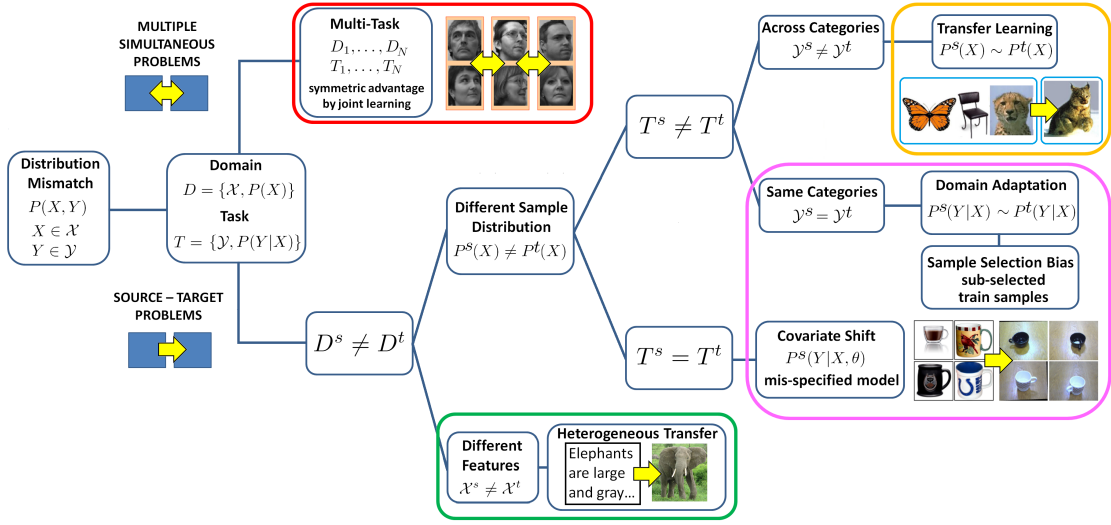
Figure 2.1 – This scheme presents the problem of distribution mismatch and the main techniques to overcome it. They are all based on exploiting the available source knowledge when solving a target learning problem. *Multi-task learning*: a practical example is that of male/female face classification when each task contains images of a different pair of subjects [150]. *Transfer learning*: learning wildcats leveraging over prior knowledge of butterflies, chairs and leopards [135]. *Domain adaptation*: there is a clear domain shift between images of cups downloaded from the web and images of cups acquired with a personal camera [143]. *Sample selection bias* is a specific case of domain adaptation, while *covariate shift* is actually originated by a different task condition, however they result in similar practical problems. In case of different features, it is possible to consider a *heterogeneous transfer*, e.g. from text to images [182].

changing the data representation or re-weighting the patterns. In this context, an interesting approach has been proposed in [43]: the domain adaptation problem is transformed into a standard supervised learning problem by augmenting the size of the feature space of both source and target data by replication. Cross-Domain SVM (CD-SVM) proposed by Jiang et al. [75] uses K-Nearest Neighbors (KNN) from the target domain to define a weight for each auxiliary source pattern and then trains SVM on the combined set of target and source reweighted samples. Apart from the few labeled data, there are always many unlabeled samples in the target set and they may also be used to improve the classification performance [74].

In the *unsupervised* case most of the adaptation approaches rely on defining new features to capture the correspondence between the domains. Blitzer et al. propose in [18] the Structural Correspondence Learning (SCL) method which models heuristically the relation among the domains through some pivot features: they are supposed to behave in the same way for discriminative learning on the source and on the target problem. In [14] Ben-David et al. try to learn directly a new representation which minimizes a bound on the target generalization error. Specifically the algorithm jointly minimizes a trade-off between source-target similarity and
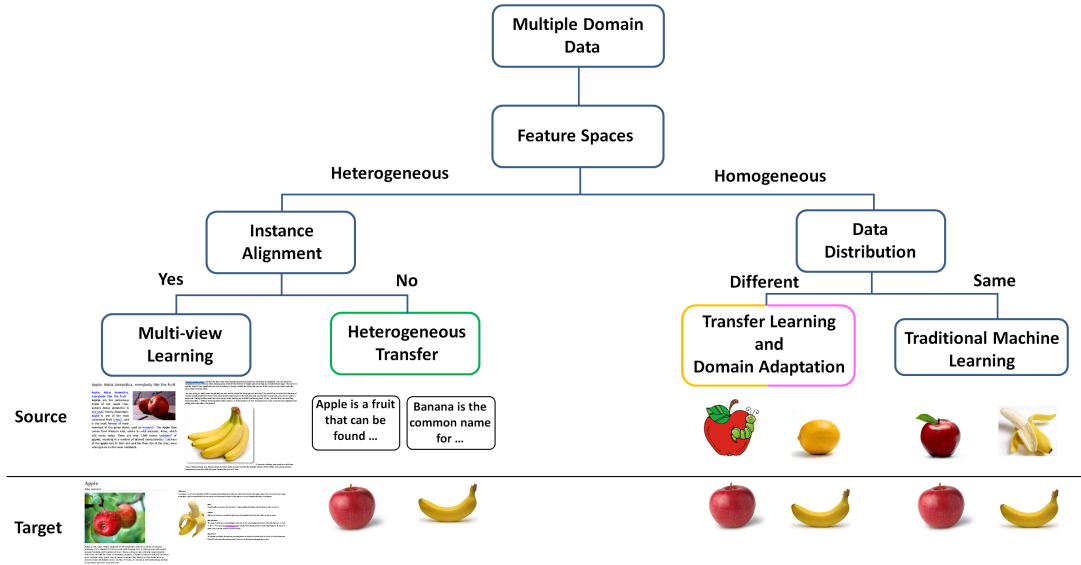
Figure 2.2 – This figure shows different possible conditions of learning over two domains. In the depicted hierarchy, the two final external leaves correspond to cases with no domain mismatch. In classical multi-view learning (left) the feature heterogeneity is not a problem because the instance alignment makes clear the correspondence among the cues. In traditional machine learning (right) the data distribution is the same among source and target. The central leaves are instead the two cases with a difference in train and testing domains: if the mismatch is caused by different features we have the heterogeneous transfer, while if it is caused by a difference in data distribution we can have domain adaptation or transfer learning depending if the labels across the domains are the same or not [182].

source domain training error. Transfer Component Analysis (TCA) [122] uses the maximum mean discrepancy [19] to identify the main components of a space where the data properties are preserved and the distribution of the samples from the two domains are close to each other. Approaches based on combining the reweighted source and target samples are also common in this setting [40, 23].

When more than one source set is available [51] proposes a domain dependent regularizer: it enforces that the target classifier produces decision values similar to those of the source classifiers on the unlabeled instances.

In the computer vision research area, [143] proposed a method to adapt object models across different visual domains. It learns a transformation that minimizes the effect of domain-induced changes in the feature distribution. Recently [65] focused on the unsupervised case presenting a technique inspired by incremental learning. It creates intermediate representations of data between the source and the target domain as points on a manifold. This approach has been casted into an efficient kernel-based method in [64].

Apart from practical solutions to specific problems, there exist also many theoretical studies

on domain adaptation. Mansour et al. [107] extended the domain-adaptation-distance already used in [14], introducing a discrepancy distance to measure the mismatch of data distributions. The same authors considered also the case of multiple available sources. They demonstrated in [106] that for any fixed target function, there exists a distribution weighted combining rule that has a loss of at most ϵ with respect to any target mixture of the source distributions.

2.1.2 Sample Selection Bias

The difference between the source and the target may be intrinsically caused by the fact that we do not have full control on the data gathering process. Medical diagnosis screen tests are often based on training data collected on subjects with a different risk level then the final target population. In econometrics, this happens whenever data are collected through surveys and the people are in some way self-selected so they do not constitute a random sample of the underlying distribution on which in practice we would like to predict accurately. This condition is recognized under the name of *sample selection bias*: the training points are drawn according to the test distribution but not all of them are made available to the learner.

Several recent machine learning publications have dealt with this problem: the main correction technique used consists in scaling the cost of training point errors to more closely reflect the test distribution [185, 148]. A detailed theoretical study on sample selection bias has been presented in [33].

2.1.3 Covariate Shift

Untill here we have considered domain adaptation problems where the mismatch between the source and the target set was due both to a difference in the domains and in the tasks. However, it can happen that the two tasks are exactly the same, namely $\mathcal{Y}^s = \mathcal{Y}^t$ and $P^s(Y|X) = P^t(Y|X)$, but still for the marginal distributions it holds $P^s(X) \neq P^t(X)$. This is known as *covariate shift*. For classification we are interested only in the conditional distribution, thus it may appear that the covariate shift is not a problem, once $P^s(Y|X)$ is known [73]. However, if we consider a parametric model family $P(Y|X,\theta)_{\theta \in \Theta}$ and we select a model $P(Y|X,\theta^*)$ to minimize the expected classification error, it can happen that there does not exist any $\theta \in \Theta$ such that $P(Y|X = x,\theta) = P(Y|X = x)$ for all $x \in X$. It means that we have a mis-specified model family and the optimal model on the source depends on $P^s(X)$ which is different from $P^t(X)$.

To address this problem, similar approaches as the one described for sample selection bias are adopted. Sugiyama et al. [154] proposed a reweighting algorithm known as Kullback–Leibler Importance Estimation Procedure (KLIEP), which is integrated with cross validation to perform model selection automatically. Bickel et al. [15] suggested to integrate the distribution correcting process into a kernelized logistic regression. Kanamori et al. [77] proposed a method called unconstrained Least-Squares Importance Fitting (uLSIF) to estimate the importance

efficiently by formulating a least-squares function fitting problem.

2.1.4 Transfer Learning

Transfer learning [123] focuses on the possibility to pass useful knowledge from a source task to a target task with different label sets $\mathcal{Y}^s \neq \mathcal{Y}^t$, when the corresponding domains are not the same but the marginal distributions of data are related $P^s(X) \sim P^t(X)$. Differently from domain adaptation, now the classes contained in the source and target set are not the same. Thus it is always necessary to evaluate in practice how much the tasks are related and whether it is really worth to rely on the source knowledge when solving a new learning problem.

Multiple transfer learning methods have been developed under specific names depending on the availability of labeled samples in the source (s) and in the target (t). Let's indicate with + and − the availability of labeled and unlabeled data and with n the number of samples.

s+t+. This corresponds to the supervised condition with labeled samples both in the source and in the target. Knowledge transfer results particularly useful to avoid overfitting when $n^{t+} \ll n^{s+}$. In particular the case with $n^{t+} = 1$ is also known as *one-shot learning* [59]. Several generative techniques based on Bayesian learning [59] and Gaussian Process [135] have been developed for this problem with applications both in classification and detection. More recently some approaches have been proposed in the discriminative framework [41].

s+t-. When no labeled samples are available in the target set, we have the so called *transductive transfer learning* [5, 10]. In this setting also the *zero-shot learning* is recently emerged [90]: the idea is to leverage not only the source knowledge but also some extra information about the relation between the source and the target such as textual attributes. For visual problems, this approach allows the identification of object categories never seen before, only through their description [89, 137].

s-t+. This particular condition is named *self-taught learning* [134] and is based on the possibility to extract some useful information from the source, even if its label set it is not known. Source knowledge can be formalized as a high level representation through unsupervised feature construction.

s-t-. This is a completely unsupervised case and the methods proposed in this setting are clustering, dimensionality reduction and density estimation [42, 178].

Transfer learning has been applied in all these different settings for wifi localization [124], natural language processing [49], sentiment classification [24] and visual problems both for still images [152] and videos [100]. More in general, knowledge transfer techniques has been widely studied in binary classification settings across pairs of categories both with a single and multiple sources available [161, 165]; only recently it has been applied to multiclass problems [136, 138, 104].

2.1.5 Heterogeneous Transfer

The case in which the source and the target set present different features is known as *heterogeneous transfer* and it covers both the conditions of fixed and changing label sets across the tasks. This is different from the classical multi-view learning where the data present multiple features but there is a sample alignment, meaning that it is known from which sample the features are extracted, and it is possible to define a correspondence between the cues (see Figure 2.2).

Different examples of heterogeneous transfer are in cross language classification, [38, 12]. Recently several methods have been presented for text-to-images classification eventually using social media (e.g. flickr) to bridge the gap [182, 190].

2.1.6 Multi Task Learning

Multi-task learning focuses on the particular case in which N undersampled sets of data are available at the same time. In this condition it is not possible to specify one source and one target problem, the goal is to learn over all the sets at the same time by exploiting a symmetric share of information. This framework supposes that all the sets have the same feature space $\mathcal{X}^i = \mathcal{X}^j$ but present slightly different domains $P^i(X) \sim P^j(X)$ for $i, j \in \{1, \ldots, N\}$. Traditionally, one either assume that the set of labels for all the tasks are the same ($\mathcal{Y}^i = \mathcal{Y}^j$) or that it is possible to access an oracle mapping function $\mathcal{Y}^i \mapsto \mathcal{Y}^j$ that aligns the labels.

Theoretical and empirical evaluations have shown that the learning performance has a significant improvement when multiple related task are considered simultaneously. In practice each task is used as bias for the others and has a positive generalization effect: the multi-task bias causes the inductive learner to prefer hypotheses that explain more than one task [25]. An extension of the VC-dimension notion has been developed for multi-task learning in [11], and it was used to derive generalization bounds on the average error of N tasks, which it is shown to decrease at best as one single task.

Different solutions have been proposed to evaluate task relatedness. If each task is associated to a model, one possibility is to measure their closeness [55, 99, 187, 125]. It might happen that the available tasks are not all equally related, thus it is important to choose with whom to share knowledge [78]. Finally [139] shows that it is possible to take advantage also from the knowledge of unrelatedness among the tasks.

Some multi-task approaches are Bayesian [44], other techniques are based on discriminative learning methods [54]. Multi-task solutions have been proposed also in terms of latent features: [4] introduced a low dimensional representation shared across multiple related tasks.

In general, many multi-task methods have been published in machine learning, some in computer vision [170, 101, 133], natural language processing [3], and genomics [116]. Most
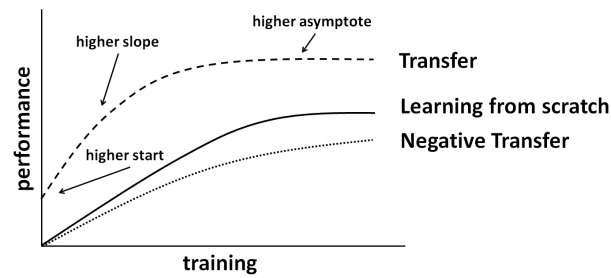
Figure 2.3 – Three ways in which transfer might improve learning and the effect of negative transfer [172].

of the works suppose multiple binary tasks and only few attempts have been done in the multiclass case without label correspondences [132, 125]. The effectiveness of multi-task has been shown in different practical situations. In finance and economics, predicting the value of many possibly related indicators simultaneously is often required [66]; in marketing modeling the preferences of many individuals simultaneously is common practice [2]; in bioinformatics, we may want to study tumor prediction from multiple micro–array data sets or analyze data from many related diseases.

2.2   Transfer Learning: Advantages and Challenges

As already mentioned, the goal of transfer learning is to improve the learning process in the target task by leveraging knowledge from the source task when the label set in source and target is different. It is possible to define three measures by which transfer might improve the effectiveness of learning, we list them below referring to Figure 2.3.

Higher start: the initial performance achievable in the target task is much better compared to that of an ignorant agent [172]. This is true even using only the source transferred knowledge, before any further learning on the specific problem.

Higher slope: this indicates a shorter amount of time needed to fully learn the target task, given the transferred knowledge, compared to that necessary in case of learning from scratch [172].

Higher asymptote: in the long run, the final performance level achievable over the target task can be higher compared to the final level without transfer [172].

To get all these advantages, is however necessary to properly answer to three questions: *what* to transfer, *how* to transfer and *when* to transfer. In particular, the last question is about whether transferring is worthwhile or not and, in case the transferring process is computational expensive, it would be also useful to consider when to stop it. In the more general framework of life long learning, knowledge transfer may encounter some specific issues in terms of

scalability with respect to the number of new target classes and possible number of sources. We dedicate the following sections to each of the described challenges with detailed references to the literature.

2.2.1 What to Transfer

What to transfer addresses which knowledge can be used to transfer across domains or tasks. Some knowledge is specific for individual domains, and some knowledge may be common between different domains such that it helps improving performance for the target task. Depending on which is the problem to solve, the transferred knowledge can be in the form of instances, feature representation, or model parameters [123].

The main idea at the basis of *instance transfer* approaches is that, although the source data cannot be reused directly, there are certain parts of them that can still be sampled and considered together with the few available labeled data in the target problem. In [41] Dai et al. proposed a boosting algorithm that uses both the source and the target samples to solve visual object classification problems. Lim et al. [98] have shown that it is possible to borrow and transform examples across different visual object classes, demonstrating a performance improvement in detection problems.

The second case can be referred to as *feature transfer* approach. The intuition behind it is to learn a good representation for the target domain encoding in it some useful knowledge extracted from the source. By exploiting the few available labeled target samples, together with the abundant set of source data, it is possible to apply feature learning as in the multi-task learning framework [4]. An alternative solution is to consider directly a metric learning approach [60] or more in general suitable kernels for the target data in SVM-based methods [142].

Finally the third case can be referred to as *parameter or model transfer* approach, which assumes that the source tasks and the target tasks share some parameters or prior distributions for the hyperparameters of the models. Fei-Fei et al. [59] proposed to transfer information via a Bayesian prior on object class models, using knowledge from known classes as a generic reference for newly learned models. Marx et al. [141] used a Bayesian transfer method for tasks solved by a logistic regression classifier. Stark et al. [152] defined a technique to transfer shape information across object classes.

2.2.2 How to Transfer

After discovering which knowledge can be transferred, learning algorithms need to be developed to properly pass information: this corresponds to the *how to transfer* issue. There is a big variety of methods in this sense: boosting approaches [41, 183], KNN [188], Markov logic [45], graphical models [39]. Most of the work has however been done in the generative probabilistic setting. Given the data, the target model makes predictions by combining them with the prior

source distribution to produce a posterior distribution. A strong prior can significantly affect these results. This serves as a natural way for Bayesian learning methods to incorporate prior knowledge in the case of transfer learning. Some discriminative (maximum margin) methods are presented in [60] by learning a distance metric, and in [8] where a template learned previously for some object categories is used to regularize the training of a new target category for detection.

2.2.3   When to Transfer

By evaluating *when* to transfer we question the relatedness of the source and the target task. Rosenstain et al. [29] empirically showed that if two tasks are dissimilar, brute force transfer hurts the performance producing the so called *negative transfer* (see Figure 2.3). Ideally, a transfer method would produce positive transfer between appropriately related tasks while avoiding negative transfer when the tasks are not a good match. In practice, these goals are difficult to achieve simultaneously. Approaches that have safeguards to avoid negative transfer often produce a smaller effect from positive transfer due to their caution. Conversely, approaches that transfer aggressively and produce large positive-transfer effects often have no protection against negative transfer.

In general we can identify two main strategies to decide when to transfer and to avoid any negative effect. One consists in rejecting bad information or at least making sure that its impact is minimized so that the transfer performance is at least no worse than what obtained by learning only on the target task. This means that it is always necessary to choose *how much* to transfer, rejecting some selected part of information and keep others, while in the extreme case, an agent might disregard the transferred knowledge completely. A different strategy can be applied in case there are more than one source task. In this condition the problem becomes choosing the best source and transfer methods without much protection against negative transfer may still be effective in this scenario, as long as the best source task is at least a decent match.

Taylor et al. [157] proposed a transfer hierarchy, ordering tasks by difficulty. Appropriate source tasks are usually less difficult than the target task. Given a task ordering, it may be possible to locate the position of the target task in the hierarchy and select the most useful source task. In [105] the authors used conditional Kolmogorov complexity to measure relatedness between tasks and transfer the right amount of information. Ruckert and Kramer [142] proposed to learn a meta-kernel that serves as a similarity function between tasks, together with a set of specific kernels for each source.

2.2.4   Life Long Learning

The fundamental motivation for knowledge transfer in the field of machine learning was discussed in the NIPS-95 workshop on *Learning to Learn*. Apart from the already discussed

questions, when thinking to an continuous learner able to exploit prior knowledge, there are two main problems to tackle: computational efficiency and novelty evaluation.

The computational efficiency issue arises both from the fact that the number of stored prior knowledge models may increase with time and also from the progressively growing number of samples with experience. Whenever a transfer learning approach is applied, the usefulness of the source models should be checked. On one side, the higher is the number of sources, the higher is the probability to find a reliable knowledge to leverage. On the other side the computational burden needed by the source evaluation progressively increases. Even if not directly facing the described issue in this perspective, some transfer learning techniques for visual object classification make use of a hierarchical ontology on the sources [191, 137, 145]. A practical solution is to exploit the higher levels of these hierarchies, going into more details only for the most useful nodes. In case of discriminative approaches a good regularizer that take care of a specific structure in the sources might also make the trick.

We have mentioned the asymptotic advantage that transfer learning should give in case of an increasing number of training samples. However, for an efficient learning agent, when a sufficiently good performance is reached on the target problem, it may be appropriate to stop the transferring process. Online transfer learning approaches have been proposed recently to take advantage of prior knowledge when considering a sequential input of new samples [189, 167]. Some active learning technique suggest instead a way to transfer online by estimating the confidence of the actual target model and possibly ask an oracle based on prior knowledge to label new samples [149, 84]. Both these solutions help in reducing the complexity with respect to batch transfer approaches and may progressively give less importance to prior knowledge. A particular mention is needed for all the literature on knowledge transfer for reinforcement learning based robot navigation problems [91, 151]. Here prior knowledge is properly used in a sequential control environment, to speed up the definition of a reliable policy function and balancing exploration and exploitation.

The basic intuition that explains the value of transfer learning in an open ended scenario is that, if a system has already learned N categories, learning the $(N+1)th$ should be easier, even from one or few training samples [159]. This implies a focus on approaching something new, while the problem reduces to more classical incremental learning otherwise. Most of the existing transfer approaches supposes that an external teacher provides the system with the novelty flag on the incoming samples, however this is not always feasible. An autonomous learning agent should be able to discern automatically if it is the case to assign or not one of the known labels. We are not aware of published works explicitly tackling this problem, but combining standard novelty detection approaches [108] with knowledge transfer methods could be a first step in this direction.

Finally, there is a last point that connects both the scalability problem in terms of increasing source models and the novelty detection. Once a new category has been learned even thanks to a knowledge transfer technique, it should be possible to automatically introduce it in the

source set such that it could help in a successive round of learning for a new target problem. Moreover it can happen that the target problem presents not only new categories but a random combination of old and new. As already mentioned, only few attempts of multiclass transfer learning have been published, but the prior and the new knowledge category sets are always kept disjoint, not considering the possibility in the test phase to distinguish a new class from one already known or eventually showing a dramatic drop in performance in this condition [138].

Part I
Leveraging over Models

3 Transfer Learning Across Categories

In this chapter we present our knowledge transfer approach for binary problems. We start by introducing the mathematical notation and then we define the formulation of the transfer learning algorithm. We specify the strategy used to evaluate when and how much to transfer and we discuss the main properties of the full method. The chapter ends presenting an extensive theoretical and experimental benchmark to other existing knowledge transfer techniques.

3.1 The Intuition

Consider the following situation. Suppose to be given the task to learn from few examples the class *motorbike* having already a model for the categories *bicycle, car, dog* and *cat*. On the basis of the visual similarity we can guess that the final model for motorbike will be close to that of car and bicycle. Thus in the learning process we would like to transfer information from these two categories. We would expect the model obtained in this way to produce better classification results with respect to (1) just considering car *or* bicycle as reference and (2) relying over all the prior knowledge in a flat way, as cat and dog might induce negative transfer.

This kind of reasoning motivate us to design a knowledge transfer algorithm able to find autonomously the best subset of known models from where to transfer and weight properly the relevant information.

3.2 Mathematical Framework

We introduce here the formal notation that will be used in this chapter. Specifically we present all the mathematical tools and concepts necessary to define our transfer learning approach. In the following we denote with small and capital bold letters respectively column vectors and matrices, e.g. $\boldsymbol{a} = [a_1, a_2, \ldots, a_N]^T \in \mathbb{R}^N$ and $\boldsymbol{A} \in \mathbb{R}^{M \times N}$ with A_{ji} corresponding to the (j, i) element.

Let us assume $\boldsymbol{x}_i \in \mathcal{X}$ to be an input vector to a learning system and $y_i \in \mathcal{Y}$ its associated

output. Given a set of data $D = \{\boldsymbol{x}_i, y_i\}_{i=1}^N$ drawn from an unknown probability distribution P, we want to find a function $f : \mathcal{X} \to \mathcal{Y}$ such that it determines the best corresponding y for any future sample $\boldsymbol{x}$. In general $\mathcal{X} \subseteq \mathbb{R}^d$, moreover if $\mathcal{Y} := \mathbb{R}$, then y_i is a real-valued random variable and we have a *regression* problem, while if y_i takes values from an unordered finite set we have pattern *classification*.

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

$$\Omega(f) + C \sum_{i=1}^N \ell(f(\boldsymbol{x}_i), y_i) \,, \tag{3.1}$$

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

By choosing the form of the loss function and the space $\mathcal{H}$ a number of different classification and regression schemes can be derived. Moreover it is possible to give a general probabilistic interpretation to the problem formulated in (3.1). Consider

$$P(f) \propto \exp\left\{-\frac{1}{2}\|f\|_{\mathcal{H}}^2\right\} \tag{3.2}$$

$$P(D|f) \propto \exp\left\{-C \sum_{i=1}^N \ell(f(\boldsymbol{x}_i), y_i)\right\} \tag{3.3}$$

here $P(f)$ denotes the prior probability and is obtained by supposing $\Omega(f) = \frac{1}{2}\|f\|_{\mathcal{H}}^2$: it indicates that the functions with small norm are more likely than the functions with a larger norm. $P(D|f)$ denotes the likelihood of the data. Thus the learning problem consists in finding the mode of the posterior $P(f|D) \propto P(D|f)P(f)$ and the solution can be obtained with a maximum a posteriori technique.

3.2.1 Adaptive Regularization

Let us consider a Reproducing Kernel Hilbert Space (RKHS) for $\mathcal{H}$ and constrain the function f to the form of linear models

$$f(\boldsymbol{x}) = \boldsymbol{w} \cdot \phi(\boldsymbol{x}) \,, \tag{3.4}$$

here $\phi(\boldsymbol{x})$ is a feature mapping that maps the samples into a high, possible infinite dimensional space, where the dot product is expressed with a functional form $K(\boldsymbol{x}, \boldsymbol{x}') = \phi(\boldsymbol{x}) \cdot \phi(\boldsymbol{x}')$ named kernel. With these assumptions, the regularizer in the learning problem (3.1) can be
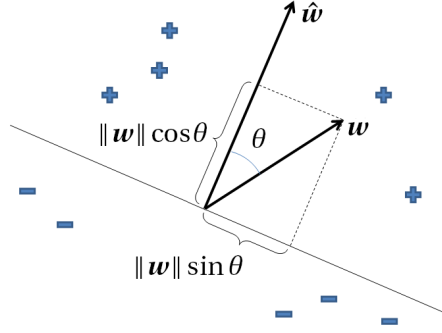
Figure 3.1 – Visualization of the projection of the vector $\boldsymbol{w}$ onto $\hat{\boldsymbol{w}}$ and onto the separating hyperplane orthogonal to $\hat{\boldsymbol{w}}$ (when using the hinge loss) [8].

written as $\Omega(f) = \frac{1}{2}\|\boldsymbol{w}\|^2$ and interpreted as choosing a zero mean Gaussian distribution on the parameter $\boldsymbol{w}$ as prior for the function f : $P(f) \propto \left\{\exp -\frac{1}{2}\|\boldsymbol{w}\|^2\right\}$. Whatever is the specific form of the loss function, we get the following optimization problem

$$\min_{\boldsymbol{w}} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N} \ell(\boldsymbol{w}\cdot\phi(\boldsymbol{x}_i), y_i) \,. \tag{3.5}$$

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

Let us suppose that the optimal $\hat{\boldsymbol{w}}$ has been already found by minimizing (3.5) for some source problem. When facing a new target task, we can use $\hat{\boldsymbol{w}}$ as mean in the Gaussian prior distribution of $\boldsymbol{w}$ i.e. $P(f) \propto \exp\left\{-\frac{1}{2}\|\boldsymbol{w} - \hat{\boldsymbol{w}}\|^2\right\}$, such that the learning problem results

$$\min_{\boldsymbol{w}} \frac{1}{2}\|\boldsymbol{w} - \hat{\boldsymbol{w}}\|^2 + C\sum_{i=1}^{N} \ell(\boldsymbol{w}\cdot\phi(\boldsymbol{x}_i), y_i) \,. \tag{3.6}$$

A detailed analysis of this formulation can be done by expanding the regularization term: $\|\boldsymbol{w} - \hat{\boldsymbol{w}}\|^2 = \|\boldsymbol{w}\|^2 + \|\hat{\boldsymbol{w}}\|^2 - 2\|\boldsymbol{w}\|\|\hat{\boldsymbol{w}}\|\cos\theta$, where θ is the angle between $\boldsymbol{w}$ and $\hat{\boldsymbol{w}}$ as shown in Figure 3.1. Thus apart from minimizing the original term $\|\boldsymbol{w}\|^2$, the optimization problem aims now at obtaining a vector $\boldsymbol{w}$ close to the source model $\hat{\boldsymbol{w}}$ by maximizing the projection of the first on the second. To properly scale the importance of this projection in the optimization problem, it is possible to add a weighting factor β such that the regularizer becomes $\|\boldsymbol{w} - \beta\hat{\boldsymbol{w}}\|^2$.

3.3   Learning to Transfer

All the transfer learning methods based on the adaptive regularization described above answer to the question *what to transfer* in terms of model parameters by passing the known $\hat{\boldsymbol{w}}$ to the new target problem.

Several generative transfer approaches in the literature are based on the probabilistic interpretation of the defined learning problem with various techniques to evaluate the relation between P and $\hat{P}$ [122, 40]. On the other hand, all the existent discriminative solutions do not pay too much attention on *when* and *how much* to transfer. The discussed weight factor β in the regularizer is usually considered equal to 1 with the hypothesis that the known models are useful and related to the target problem [181]. In other cases β is treated as a learning parameter and is chosen by cross validation supposing the availability of extra training samples [8]. Both these choices present some issues: the first case does not consider the danger of negative transfer when only unrelated prior information is available, while in the second, the existence of extra data for cross validation is incoherent with the small sample scenario of transfer learning.

Here we propose instead an adaptive learning approach that decides autonomously and in a principled way how much each prior knowledge model is reliable for the new target task. This is done by tuning automatically the value of the β weight by just using the few available target training data. In the following subsections we present the basic elements of this approach.

3.3.1   Adaptive Least-Square Support Vector Machine

The first step towards our goal consists in choosing the square loss

$$\ell^S(f(\boldsymbol{x}), y) = (f(\boldsymbol{x}) - y)^2 \; . \tag{3.7}$$

This cost function is commonly used in ridge regression, regularized least square classification and gaussian process regression [17]. The optimization problem reads now like this [161]:

$$\min_{\boldsymbol{w}, b} \frac{1}{2} \| \boldsymbol{w} - \beta \hat{\boldsymbol{w}} \|^2 + \frac{C}{2} \sum_{i=1}^{N} \xi_i^2$$

$$\text{subject to} \quad y_i = \boldsymbol{w} \cdot \phi(\boldsymbol{x}_i) + b + \xi_i \quad \text{for} \quad i = 1, \dots, N \tag{3.8}$$

where we have also generalized the linear model adding a bias term $f(\boldsymbol{x}) = \boldsymbol{w} \cdot \phi(\boldsymbol{x}) + b$, and we have introduced the slack variables ξ_i which measure the degree of misclassification on the data $\boldsymbol{x}_i$. Due to the similarity with the soft margin version of the classical Support Vector Machine formulation [37], we use here the name Adaptive Least-Square Support Vector Machine (following the Least-Square SVM in [156]). The objective function in 3.8 seeks to reduce the distance between $\boldsymbol{w}$ and $\hat{\boldsymbol{w}}$ while minimizing the error measured by the square loss.

The corresponding Lagrangian problem is:

$$\mathcal{L} = \frac{1}{2} \| \boldsymbol{w} - \beta \hat{\boldsymbol{w}} \|^2 + \frac{C}{2} \sum_{i=1}^{N} \xi_i^2 - \sum_{i=1}^{N} a_i \{ \boldsymbol{w} \cdot \phi(\boldsymbol{x}_i) + b + \xi_i - y_i \} \,, \tag{3.9}$$

where $\boldsymbol{a} \in \mathbb{R}^N$ is the vector of Lagrange multipliers. The optimality conditions can be expressed as:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}} = \boldsymbol{0} \implies \boldsymbol{w} = \beta \hat{\boldsymbol{w}} + \sum_{i=1}^{N} a_i \phi(\boldsymbol{x}_i) \,, \tag{3.10}$$

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \implies \sum_{i=1}^{N} a_i = 0 \,, \tag{3.11}$$

$$\frac{\partial \mathcal{L}}{\partial \xi_i} = 0 \implies a_i = C \xi_i \,, \tag{3.12}$$

$$\frac{\partial \mathcal{L}}{\partial a_i} = 0 \implies \boldsymbol{w} \cdot \phi(\boldsymbol{x}_i) + b + \xi_i - y_i = 0 \,. \tag{3.13}$$

From (3.10) it is clear that the adapted model is given by the sum of the pre-trained source model $\hat{\boldsymbol{w}}$ (weighted by β) and a linear combination of the target samples. Note that when β is 0 we recover the original LS-SVM formulation without any adaptation to previous knowledge. By using (3.10) and (3.12) to eliminate $\boldsymbol{w}$ and $\boldsymbol{\xi}$ from (3.13) we find that:

$$\sum_{j=1}^{N} a_j \, \phi(\boldsymbol{x}_j) \cdot \phi(\boldsymbol{x}_i) + b + \frac{a_i}{C} = y_i - \beta \hat{\boldsymbol{w}} \cdot \phi(\boldsymbol{x}_i) \,. \tag{3.14}$$

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

$$\begin{bmatrix} \boldsymbol{K} + \frac{1}{C}\boldsymbol{I} & \boldsymbol{1} \\ \boldsymbol{1}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{a} \\ b \end{bmatrix} = \begin{bmatrix} \boldsymbol{y} - \beta \hat{\boldsymbol{y}} \\ 0 \end{bmatrix} \,, \tag{3.15}$$

where $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$ are the vectors containing respectively the label samples and the prediction of the previous model i.e. $\boldsymbol{y} = [y_1, \ldots, y_N]^T$, $\hat{\boldsymbol{y}} = [\hat{\boldsymbol{w}} \cdot \phi(\boldsymbol{x}_1), \ldots, \hat{\boldsymbol{w}} \cdot \phi(\boldsymbol{x}_N)]^T$. Thus the model parameters can be calculated simply by matrix inversion:

$$\begin{bmatrix} \boldsymbol{a} \\ b \end{bmatrix} = \boldsymbol{P} \begin{bmatrix} \boldsymbol{y} - \beta \hat{\boldsymbol{y}} \\ 0 \end{bmatrix} \,, \tag{3.16}$$

where $\boldsymbol{P} = \boldsymbol{M}^{-1}$ and $\boldsymbol{M}$ is the first matrix on the left in (3.15). We underline that the pre-trained model $\hat{\boldsymbol{w}}$ can be obtained by any training algorithm, as far as it can be expressed as a weighted sum of kernel functions, the framework is therefore very general.

3.3.2 Leave-One-Out Predictions

Let us denote by $\tilde{y}_i$, $i = 1, \ldots, N$, the prediction on sample i when it is removed from the training set. LS-SVM in its original formulation makes it possible to write these leave-one-out predictions in closed form and with a negligible additional computational cost [29]. We show below that the same property extends to the modified problem in (3.8).

Proposition 1. *Let* $[\boldsymbol{a}'^T, b']^T = \boldsymbol{P}[\boldsymbol{y}^T, 0]^T$ *and* $[\boldsymbol{a}''^T, b'']^T = \boldsymbol{P}[\hat{\boldsymbol{y}}^T, 0]^T$ *with* $\boldsymbol{a} = \boldsymbol{a}' - \beta \boldsymbol{a}''$. *The prediction* $\tilde{y}_i$, *obtained on sample i when it is removed from the training set, is equal to*

$$y_i - \frac{a_i'}{P_{ii}} + \beta \frac{a_i''}{P_{ii}} . \tag{3.17}$$

Proof. We start from

$$\boldsymbol{M} \begin{bmatrix} \boldsymbol{a} \\ b \end{bmatrix} = \begin{bmatrix} \boldsymbol{y} - \beta \hat{\boldsymbol{y}} \\ 0 \end{bmatrix} . \tag{3.18}$$

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

$$\boldsymbol{M} = \begin{bmatrix} \mathbf{K} + \frac{1}{C}\mathbf{I} & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix} = \begin{bmatrix} m_{11} & \mathbf{m}_1^T \\ \mathbf{m}_1 & \mathbf{M}_{(-1)} \end{bmatrix} . \tag{3.19}$$

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

$$\begin{bmatrix} \boldsymbol{a}_{(-1)} \\ b_{(-1)} \end{bmatrix} = \mathbf{P}_{(-1)}(\boldsymbol{y}_{(-1)} - \beta \hat{\boldsymbol{y}}_{(-1)}) , \tag{3.20}$$

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

$$\begin{aligned} \tilde{y}_1 &= \mathbf{m}_1^T \begin{bmatrix} \boldsymbol{a}_{(-1)} \\ b_{(-1)} \end{bmatrix} + \beta \hat{\boldsymbol{w}} \cdot \phi(\boldsymbol{x}_1) \\ &= \mathbf{m}_1^T \mathbf{P}_{(-1)}(\boldsymbol{y}_{(-1)} - \beta \hat{\boldsymbol{y}}_{(-1)}) + \beta \hat{\boldsymbol{w}} \cdot \phi(\boldsymbol{x}_1) . \end{aligned} \tag{3.21}$$

Considering the last N equations in the system in (3.18), it is clear that $[\mathbf{m}_1 \ \mathbf{M}_{(-1)}][\boldsymbol{a}^T, b]^T = (\boldsymbol{y}_{(-1)} - \beta \hat{\boldsymbol{y}}_{(-1)})$, and so

$$\begin{aligned} \tilde{y}_1 &= \mathbf{m}_1^T \mathbf{P}_{(-1)}[\mathbf{m}_1 \mathbf{M}_{(-1)}][a_1, \ldots, a_N, b]^T + \beta \hat{\boldsymbol{w}} \cdot \phi(\boldsymbol{x}_1) \\ &= \mathbf{m}_1^T \mathbf{P}_{(-1)} \mathbf{m}_1 a_1 + \mathbf{m}_1^T [a_2, \ldots, a_N, b]^T + \beta \hat{\boldsymbol{w}} \cdot \phi(\boldsymbol{x}_1) . \end{aligned} \tag{3.22}$$

Noting from the first equation in the system in (3.18) that $y_1 - \beta \hat{\boldsymbol{w}} \cdot \phi(\boldsymbol{x}_1) = m_{11} a_1 + \mathbf{m}_1^T [a_2, \ldots, a_N, b]^T$,

we have

$$\tilde{y}_1 = y_1 - a_1(m_{11} - \mathbf{m}_1^T \mathbf{P}_{(-1)} \mathbf{m}_1) \ . \tag{3.23}$$

Finally, by using $\mathbf{P} = \mathbf{M}^{-1}$ and applying the block matrix inversion lemma we get,

$$\mathbf{P} = \begin{bmatrix} \mu^{-1} & -\mu^{-1}\mathbf{m}_1\mathbf{P}_{(-1)} \\ \mathbf{P}_{(-1)} + \mu^{-1}\mathbf{P}_{(-1)}\mathbf{m}_1^T\mathbf{m}_1\mathbf{P}_{(-1)} & -\mu^{-1}\mathbf{P}_{(-1)}\mathbf{m}_1^T \end{bmatrix} \ ,$$

where $\mu = m_{11} - \mathbf{m}_1^T \mathbf{P}_{(-1)} \mathbf{m}_1$. The system of linear equations (3.18) is insensitive to permutations of the ordering of the equations and of the unknowns, thus we have

$$\tilde{y}_i = y_i - \frac{a_i}{P_{ii}} \ . \tag{3.24}$$

By considering $\boldsymbol{a} = \boldsymbol{a}' - \beta \boldsymbol{a}''$, $[\boldsymbol{a}'^T, b']^T = \boldsymbol{P}[\boldsymbol{y}^T, 0]^T$ and $[\boldsymbol{a}''^T, b'']^T = \boldsymbol{P}[\hat{\boldsymbol{y}}^T, 0]^T$, from the equation above we get :

$$\tilde{y}_i = y_i - \frac{a_i'}{P_{ii}} + \beta \frac{a_i''}{P_{ii}} \ . \tag{3.25}$$

$\square$

Notice that $\boldsymbol{a}$ depends linearly on β , thus it is straightforward to define the learning model once β has been chosen.

3.3.3 Multiple Sources

The introduction of the transfer learning method up to here has been done supposing the presence of a unique known model $\hat{\boldsymbol{w}}$ as source of prior information. However, as discussed in section 2.2.3, relying on more than one source may be very helpful to avoid negative transfer. To this goal, we present below how to enlarge the learning method considering a linear combination of $j = 1, \dots, J$ known models [165]:

$$\min_{\boldsymbol{w}, b} \frac{1}{2} \left\| \boldsymbol{w} - \sum_{j=1}^{J} \beta_j \hat{\boldsymbol{w}}_j \right\|^2 + \frac{C}{2} \sum_{i=1}^{N} \xi_i^2$$

$$\text{subject to} \quad y_i = \boldsymbol{w} \cdot \phi(\boldsymbol{x}_i) + b + \xi_i \quad \text{for} \quad i = 1, \dots, N \ . \tag{3.26}$$

The original single coefficient β has been substituted with a vector $\boldsymbol{\beta}$ containing as many elements as the number of prior models, J . For this formulation the optimal solution is:

$$\boldsymbol{w} = \sum_{j=1}^{J} \beta_j \hat{\boldsymbol{w}}_j + \sum_{i=1}^{N} a_i \phi(\boldsymbol{x}_i) \ . \tag{3.27}$$

Here $\boldsymbol{w}$ 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. The leave-one-out prediction of

each sample i can again be written in closed form, similarly to (3.25), as

$$\tilde{y}_i = y_i - \frac{a'_i}{P_{ii}} + \sum_{j=1}^{J} \beta_j \frac{a''_{ij}}{P_{ii}}, \tag{3.28}$$

where $[\boldsymbol{a}''^T_j, b''_j]^T = \boldsymbol{P}[\hat{\boldsymbol{y}}^T_j, 0]^T$ and $\hat{\boldsymbol{y}}_j$ is the vector which contains the predictions of the j-th previous model $[\hat{\boldsymbol{w}}_j \cdot \phi(\boldsymbol{x}_1), \ldots, \hat{\boldsymbol{w}}_j \cdot \phi(\boldsymbol{x}_N)]$.

3.3.4 When and How Much to Transfer

Evaluating the elements of the weight vector $\boldsymbol{\beta}$ corresponds to ranking the prior knowledge sources and decide from where and how much to transfer. We can exploit the availability of a closed form for the leave-one-out predictions and derive the leave-one-out error which is an almost unbiased estimator of the generalization error [29]. By multiplying the correct label y_i to (3.28) we get

$$y_i \tilde{y}_i = 1 - y_i \left(\frac{a'_i}{P_{ii}} - \sum_{j=1}^{J} \beta_j \frac{a''_{ij}}{P_{ii}} \right), \tag{3.29}$$

thus the best values for β_j are those producing positive values for $y_i \tilde{y}_i$, for each i. However focusing only on the sign of those quantities would result in a non-convex formulation with many local minima. We propose instead the following loss function [165]:

$$\begin{aligned} \ell(\tilde{y}_i, y_i) &= \max\{0, 1 - y_i \tilde{y}_i\} \\ &= \max \left\{ 0, y_i \left(\frac{a'_i}{P_{ii}} - \sum_{j=1}^{J} \beta_j \frac{a''_{ij}}{P_{ii}} \right) \right\}. \end{aligned} \tag{3.30}$$

This loss function is similar to the hinge loss

$$\ell^H(f(\boldsymbol{x}), y) = \max\{0, 1 - yf(\boldsymbol{x})\}. \tag{3.31}$$

It is a convex upper bound to the leave-one-out misclassification loss and it favors solutions in which $\tilde{y}_i$ has a value of 1, beside having the same sign of y_i. Finally, the objective function is:

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{N} \ell(y_i, \tilde{y}_i) \quad \text{subject to} \quad \|\boldsymbol{\beta}\| \le 1, \; \beta_j \ge 0. \tag{3.32}$$

The constraint of having $\boldsymbol{\beta}$ in the intersection of the unitary ball and the positive semi-plane, can be seen as a form of regularization. It is necessary to avoid overfitting problems which can happen when the number of known models J is large compared to the number of training samples N. Notice that this formulation is equivalent to the more common optimization problem $(1/2)\|\boldsymbol{\beta}\|^2 + CJ$ for a proper choice of C [37]. By solving (3.32) we can find the best values of β_j for $j = 1, \ldots, J$ to weight the known prior models in transfer learning.

The optimization process can be implemented by using a simple projected sub-gradient descent algorithm, where at each iteration $\boldsymbol{\beta}$ is projected onto the L_2-sphere $\|\boldsymbol{\beta}\| \leq 1$, and then on the positive semi-plane. The pseudo-code is in Appendix in Algorithm 4.

3.3.5 Sample Unbalance

When focusing on binary classification, e.g. in the case of object-vs-background, the problem of scarcity in labeled samples is generally related to the positive (object) class. This means that the training data present a strong unbalance with few positive and many negative samples while it is possible that the test set has a more balanced distribution among the two classes. To take care of this in the learning process, we can reweight each sample by using

$$\zeta_i = \begin{cases} \frac{N}{2N^+} & \text{if } y_i = +1 \\ \frac{N}{2N^-} & \text{if } y_i = -1 \,. \end{cases} \tag{3.33}$$

Here N^+ and N^- represent the number of positive and negative examples respectively. We can introduce the weights in (3.26) obtaining

$$\min_{\boldsymbol{w},b} \frac{1}{2} \left\| \boldsymbol{w} - \sum_{j=1}^{J} \beta_j \hat{\boldsymbol{w}}_j \right\|^2 + \frac{C}{2} \sum_{i=1}^{N} \zeta_i \xi_i^2$$

$$\text{subject to} \quad y_i = \boldsymbol{w} \cdot \phi(\boldsymbol{x}_i) + b + \xi_i \qquad \text{for} \quad i = 1, \ldots, N \,. \tag{3.34}$$

In this way the weighting factors ζ_i help to balance the contribution of the sets of positive and negative examples to the data misfit term [161, 165]. This takes also into account that the proportion of positive and negative examples in the training data are known to be not representative of the operational class frequencies. The corresponding matricial form now is

$$\begin{bmatrix} \mathbf{K} + \frac{1}{C}\mathbf{Z} & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{a} \\ b \end{bmatrix} = \begin{bmatrix} \boldsymbol{y} - \beta\hat{\boldsymbol{y}} \\ 0 \end{bmatrix}, \tag{3.35}$$

where $\mathbf{Z} = diag\{\zeta_1^{-1}, \zeta_2^{-1}, \ldots, \zeta_N^{-1}\}$ replaces the identity matrix $\mathbf{I}$. Still the procedure to solve the optimization problem is analogous to the one already described in section 3.3.1.

The reweighting procedure can also be beneficial when evaluating the relevance of each prior knowledge model to the target task [161, 165]. Hence we may also modify the function proposed in (3.30) as

$$\ell(\tilde{y}_i, y_i) = \zeta_i \max\{0, 1 - y_i\tilde{y}_i\} \,. \tag{3.36}$$

3.4 Properties

The combination of the adaptive learning process defined according to (3.34) and the choice of prior knowledge weights on the basis of (3.32) and (3.36) define our Knowledge Transfer

(KT) approach. In this section we discuss both theoretically and empirically some of its properties. With this aim we ran proof of concept experiments on a subset of the Caltech-256 dataset [67] which has a specific *clutter* category that can be used as negative class in object-vs-background problems. Moreover, this dataset is associated to a hierarchical graph that describes the ontology over the 256 covered object classes. This makes it easy to select related and unrelated categories.

3.4.1 Computational Complexity

From a computational point of view the runtime of the KT algorithm is $\mathcal{O}(N^3 + JN^2)$, with N the number of training samples, and J the number of prior knowledge models. The first term is related to the evaluation of the matrix $\boldsymbol{P}$, which must anyway occur while training, while the second term is the computational complexity of (3.28) which results negligible, if compared to the complexity of training. Thus we match the complexity of a plain SVM, which in the worst case is known to be $\mathcal{O}(N^3)$ [69], 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 to optimize (3.30) is $\mathcal{O}(JN)$ and results extremely fast (the MATLAB implementation takes just half a second with $N = 12$ and $J = 3$ on current hardware).

3.4.2 Setting the Constraints

The optimization problem defined to choose from where and how much to transfer considers a constraint on the Euclidean norm of the weight vector $\boldsymbol{\beta}$ (see equation (3.32)). However it is even possible to use other regularization conditions that give a different flavour to the final choice over the prior knowledge models.

Let's define with

$$\|\boldsymbol{x}\|_p := \left(\sum_{i=1}^{d} |x_i|^p \right)^{1/p} \tag{3.37}$$

the *p-norm* of a vector $\boldsymbol{x} \in \mathbb{R}^d$. For different values of p we get

L$_2$ norm when $p = 2$. This is the well known Euclidean norm indicated by $\|\cdot\|_2$ or simply $\|\cdot\|$. A regularization based on it generally induces numerical stability and might produce a balancing effect with the vector elements forced to be more similar to each other.

L$_1$ norm when $p = 1$. This is simply the sum of the absolute values of the vector elements. By minimizing it for regularization we get a sparsity condition i.e. only some vector elements remain different from zero. Applied on prior knowledge regularization, the condition $\|\boldsymbol{\beta}\|_1 \leq 1$ can be easily implemented on the basis of the algorithm proposed in [52], and gives rise to an automatic source selection technique.
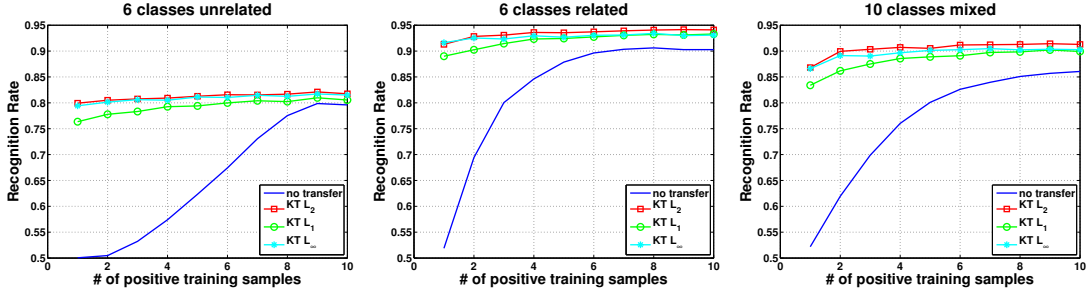
Figure 3.2 – Performance of the proposed Knowledge Transfer (KT) method with various settings for the constraint on the prior knowledge weights. The results correspond to average recognition rate over the categories, considering each class-out experiment repeated ten times. From left to right the title of each plot indicates the different level of relatedness between the source models and the target task.

L_∞ **norm** is a particular case defined as

$$\|\boldsymbol{x}\|_\infty := \max\{|x_1|, \ldots, |x_d|\} . \tag{3.38}$$

In practice, by using $\|\boldsymbol{\beta}\|_\infty \le 1$ as regularizer we are imposing that all the vector elements assume separately an absolute value not bigger than one. This can be simply obtained by truncation.

We compare the results obtained by KT with these different conditions for $\boldsymbol{\beta}$, over three groups of data that differ in the level of relatedness among source and target knowledge. Specifically we extracted 6 unrelated classes (harp, microwave, fire-truck, cowboy-hat, snake, bonsai), 6 related classes (all vehicles: bulldozer, fire-truck, motorbikes, school-bus, snowmobile, car-side) and 10 mixed classes (motorbikes, dog, cactus, helicopter, fighter-jet, car-side, dolphin, zebra, horse, goose) from Caltech-256. We refer to a class as the combination of 80 object and 80 background images. We perform the experiments in a leave-one-class-out approach, considering in turn each class as target and all the others as sources. When a specific class is used as target, we extract randomly from its set 20 training and 100 testing samples with half positive and half negative data. We also consider different steps in training by adding one positive sample at the time while keeping fixed the number of negative samples to 10. The random extraction is repeated 10 times, for an equal number of experimental runs. We suppose that each prior knowledge model is built with classical LS-SVM, and we use the Gaussian kernel for all the experiments $K(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\gamma\|\boldsymbol{x} - \boldsymbol{x}'\|^2)$.

We used the pre-computed features of [61] which the authors made available on their website [1]. Specifically, we selected four different image descriptors: PHOG Shape Descriptors [20], SIFT Appearance Descriptors [103], Region Covariance [174] and Local Binary Patterns [117]. They were all computed in a spatial pyramid [92], we considered just the first level (i.e. information extracted from the whole image) and combined the features using the average kernel. We also

---

1. http://www.vision.ee.ethz.ch/~pgehler/projects/iccv09/

benchmark all the results against *no transfer*: this corresponds to learning from scratch using weighted-LS-SVM, i.e. solving the optimization problem in equation (3.34) with $\boldsymbol{\beta} = \mathbf{0}$.

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

The results in Figure 3.2 show the clear gain obtained by using KT with respect to learning from scratch. The advantage is maximum in case of related classes (the difference between KT L_2 and no transfer is 39% in recognition rate for 1 positive sample), it is just a little bit smaller for mixed classes (34%) and drops more in case of sources unrelated to the target task (29%). However, regardless of the level of relatedness, the choice of the constraint for the prior knowledge weight $\boldsymbol{\beta}$ does not produce significantly different results [2], apart for a slightly lower performance of the L_1 case with respect to the others. In the following we will keep the L_2 norm constraint.

As a final remark we comment on the second condition that limits the weights for the prior knowledge models to be positive: $\beta_j \geq 0$. All the considered source and target sets have the background category as common negative class, thus it is reasonable to expect that the angle between $\boldsymbol{w}$ and any $\hat{\boldsymbol{w}}_j$ is always acute. The projection of the first vector on the second is already a positive value, by using a positive weight we indicate how much the source is relevant for the target problem. We exclude the condition of $\boldsymbol{w}$ closer to $-\hat{\boldsymbol{w}}_j$ than to $\hat{\boldsymbol{w}}_j$ by imposing $\beta_j = 0$ for $j = 1, \ldots, J$ in this case.

3.4.3 Setting Prior Knowledge

The proposed KT method is based on the hypothesis of pushing the target model $\boldsymbol{w}$ close to a linear combination of prior known sources $\sum_{j=1}^{J} \beta_j \hat{\boldsymbol{w}}_j$. However, to impose this closeness, all the vectors should live in a single space, this means that the kernel used in learning over all the sources and on the new target must be the same. This is quite a strict condition because it does not give the freedom to build prior knowledge over different feature descriptors and imposes a unique metric to evaluate the sample similarity.

We show here that this limit may be easily overcome by enlarging the space in which we seek

---

2. We used the sign test [62] to evaluate the statistical significance of the results for all the experiments in this chapter.
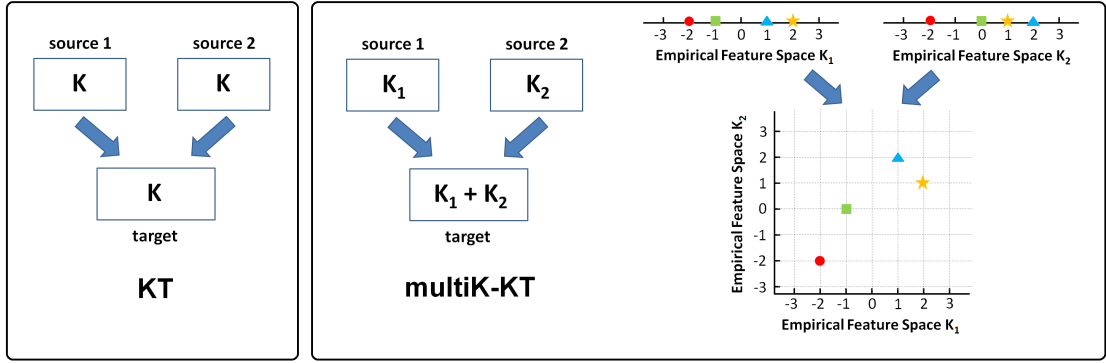
Figure 3.3 – This scheme shows the difference between KT and multiK-KT. For the first the source and target models must live in the same space identified by the kernel K. For MultiK-KT all the sources can be defined independently in their own space and the target solution lives in the space obtained by orthogonal combination. We show also a geometrical interpretation of the kernel combination.

the final learning function on the target, by a multi-kernel approach. We call this variant multiK-KT.

Let us assume to have $j = 1, \ldots, J$ mappings, each to a different space, where the image of a vector $\boldsymbol{x}$ is $\phi_j(\boldsymbol{x})$. We can always compose all of them orthogonally (see Figure 3.3) obtaining the mapping to the final space by concatenation: $\phi'(\boldsymbol{x}) = [\phi_1(\boldsymbol{x}), \phi_2(\boldsymbol{x}), \ldots, \phi_J(\boldsymbol{x})]^T$. The dot product in this new space is expressed by the kernel K'

$$
\begin{aligned}
\phi'(\boldsymbol{x}) \cdot \phi'(\boldsymbol{z}) &= \sum_{j=1}^{J} \phi_j(\boldsymbol{x}) \cdot \phi_j(\boldsymbol{z}) \\
&= \sum_{j=1}^{J} K_j(\boldsymbol{x}, \boldsymbol{z}) = K'(\boldsymbol{x}, \boldsymbol{z}) ,
\end{aligned}
\tag{3.39}
$$

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

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

$$
\hat{\boldsymbol{w}}_j = \sum_{i=1}^{\hat{N}_j} \hat{a}_{ij} \phi_j(\boldsymbol{x}_i)
\tag{3.40}
$$

and can always be mapped in the composed new space using zero padding. Since $\phi_j(\boldsymbol{x}) \rightarrow$

$\phi'_j(\boldsymbol{x}) = [0, \dots, \phi_j(\boldsymbol{x}), \dots, 0]^T$, we have

$$\hat{\boldsymbol{w}}_j \to \hat{\boldsymbol{w}}'_j = [0, \dots, \hat{\boldsymbol{w}}_j, \dots, 0]^T$$

$$= [0, \dots, \sum_{i=1}^{\hat{N}_j} \hat{a}_{ij} \phi_j(\boldsymbol{x}_i), \dots, 0]^T . \tag{3.41}$$

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

$$\sum_{j=1}^{J} \beta_j \hat{\boldsymbol{w}}'_j = [\beta_1 \hat{\boldsymbol{w}}_1, \dots, \beta_J \hat{\boldsymbol{w}}_J]^T$$

$$= [\beta_1 \sum_{i=1}^{\hat{N}_1} \hat{a}_{i1} \phi_1(\boldsymbol{x}_i), \dots, \beta_J \sum_{i=1}^{\hat{N}_J} \hat{a}_{iJ} \phi_J(\boldsymbol{x}_i)]^T . \tag{3.42}$$

By supposing that the target problem lives in the new composed space, we can apply the KT algorithm there. Hence the original optimization problem in equation (3.34) becomes:

$$\min_{\boldsymbol{w}', b} \frac{1}{2} \left\| \boldsymbol{w}' - \sum_{j=1}^{J} \beta_j \hat{\boldsymbol{w}}'_j \right\|^2 + \frac{C}{2} \sum_{i=1}^{N} \zeta_i (y_i - \boldsymbol{w}' \cdot \phi'(\boldsymbol{x}_i) - b)^2 . \tag{3.43}$$

The solving procedure is the same described in section 3.3.1 and the optimal solution is:

$$\boldsymbol{w}' = \sum_{j=1}^{J} \beta_j \hat{\boldsymbol{w}}'_j + \sum_{i=1}^{N} a_i \phi'(\boldsymbol{x}_i) . \tag{3.44}$$

When we use it for classification we get

$$\boldsymbol{w}' \cdot \phi'(\boldsymbol{z}) = \sum_{j=1}^{J} \beta_j \hat{\boldsymbol{w}}'_j \cdot \phi'(\boldsymbol{z}) + \sum_{i=1}^{N} a_i \phi'(\boldsymbol{x}_i) \cdot \phi'(\boldsymbol{z}) \tag{3.45}$$

$$= \sum_{j=1}^{J} \beta_j \hat{\boldsymbol{w}}_j \cdot \phi_j(\boldsymbol{z}) + \sum_{i=1}^{N} a_i \left(\sum_{j=1}^{J} \phi_j(\boldsymbol{x}_i) \cdot \phi_j(\boldsymbol{z}) \right) , \tag{3.46}$$

that is exactly the same that would be obtained from (3.27) supposing to use $K'(\boldsymbol{x}, \boldsymbol{z})$ as kernel.

In practice we could rewrite the transfer learning problem using the classical multi-kernel learning formulation [9]:

$$\min_{\boldsymbol{w}_j, b} \frac{1}{2} \left(\sum_{j=1}^{J} \| \boldsymbol{w}_j - \beta_j \hat{\boldsymbol{w}}_j \| \right)^2 + \frac{C}{2} \sum_{i=1}^{N} \zeta_i \left(y_i - \sum_{j=1}^{J} \boldsymbol{w}_j \cdot \phi_j(\boldsymbol{x}_i) - b \right)^2 . \tag{3.47}$$

This is equivalent to (3.43) since in our setting all the $\boldsymbol{w}_j$ (and $\hat{\boldsymbol{w}}_j$) for $j = 1, \dots, J$ are mutually orthogonal.
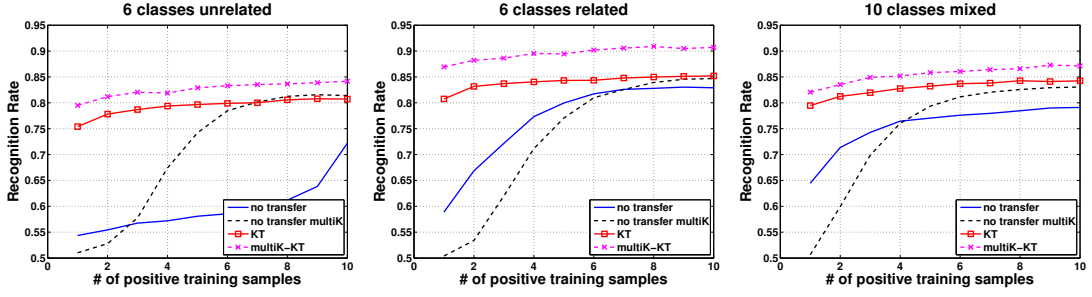
Figure 3.4 – Performance of the multiK-KT method in comparison with the single kernel KT formulation. The curves identified by *no transfer* and *no transfer multiK* corresponds respectively to learning from scratch by using only the Gaussian kernel or the combination of generalized Gaussian kernels. The results correspond to average recognition rate over the categories, considering each class-out experiment repeated ten times. From left to right the title of each plot describes the different level of relatedness between the source models and the target task.
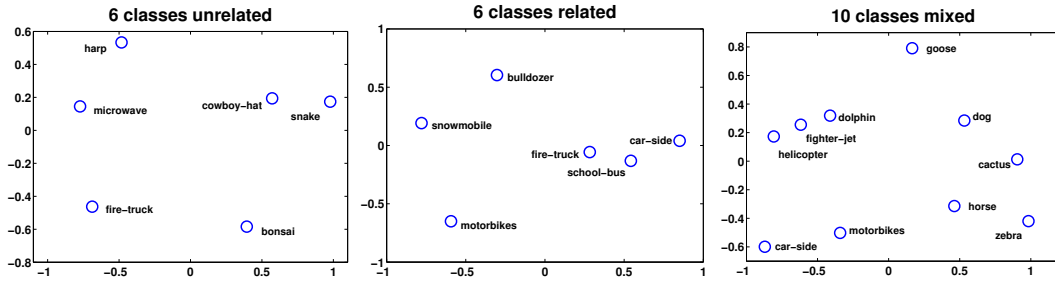
Figure 3.5 – Output of the bidimensional scaling applied on the $\boldsymbol{\beta}$ vector values. The plots indicate that the transfer learning weights are actually related to the semantic similarity among the objects.

Even the procedure to define the best $\boldsymbol{\beta}$ can be easily enlarged to the case of linearly combined orthogonal spaces. The vector $\hat{\boldsymbol{y}}'_j$ containing the predictions of the $j-$th known model is:

$$
\begin{aligned}
\hat{\boldsymbol{y}}'_j &= [\hat{\boldsymbol{w}}'_j \cdot \boldsymbol{\phi}'(\boldsymbol{x}_1), \dots, \hat{\boldsymbol{w}}'_j \cdot \boldsymbol{\phi}'(\boldsymbol{x}_N))] \\
&= [\hat{\boldsymbol{w}}_j \cdot \boldsymbol{\phi}_j(\boldsymbol{x}_1), \dots, \hat{\boldsymbol{w}}_j \cdot \boldsymbol{\phi}_j(\boldsymbol{x}_N))] \\
&= \hat{\boldsymbol{y}}_j,
\end{aligned}
\tag{3.48}
$$

indeed there is no real changing and it is possible to choose from where and how much to transfer proceeding exactly as in the original formulation once chosen the kernel as in (3.39).

The presented space enlarging trick not only allows us to overcome the problem raised by the existence of a variability in the prior knowledge sources, but also, by exploiting this higher level of freedom, produces better performance than the single space case. We show it with experiments on the same data sets used in the previous section. Here we considered SIFT as

unique descriptor together with the generalized Gaussian kernel: $K(\boldsymbol{x}, \boldsymbol{z}) = \exp(-\gamma \|\boldsymbol{x}^\rho - \boldsymbol{z}^\rho\|^\delta)$. Each source knowledge is defined considering the best set $\{\gamma, \rho, \delta\}$ obtained by cross validation on the corresponding class, while we learn on the target task considering the sum over the kernels. We name *no transfer multiK* the baseline corresponding to learning from scratch in this combined space. Figure 3.4 presents the obtained results in comparison with the case of using a single standard Gaussian kernel with fixed γ for sources and target tasks (*no* transfer and *KT* curves in the plot). We can state that *multiK-KT* performs significantly better than *KT* ($p \leq 0.002$) regardless of the level of relatedness between the source and the target classes.

3.4.4 Transfer Weights and Semantic Similarity

The presented KT algorithm defines automatically the relevance of each source model to the current target task on the basis of the descriptors extracted from the images. We want to analyze the $\boldsymbol{\beta}$ vector obtained as a byproduct of the transfer process to verify if its elements have a correspondence with the real visual and semantic relation among the tasks.

We start from the results obtained in section 3.4.2 with the L_2 norm constraint and we consider the intermediate training step with 5 positive samples. We average the $\boldsymbol{\beta}$ vectors obtained over the 10 runs defining a matrix of weights with one row for each class used as target. By simple algebra we can transform it to a fully symmetric matrix containing measures of class dissimilarities evaluated as $(1 - \beta_j)$ and apply multidimensional scaling on it [34]. To have an immediate visualization we considered only two dimensions and we obtain plots where each point represents a class, and the distance among the points is directly proportional to the input dissimilarities.

Figure 3.5 shows the obtained results. It can be seen that in the case of unrelated classes the corresponding points tend to be far from each other. On the other hand, among the related classes extracted from the general category *motorized-ground-vehicles* the *four wheels* vehicles (fire-truck, school-bus and car-side) form a cluster, leaving aside motorbikes (two wheels), snowmobile (skis) and bulldozer (tracks). Finally among the mixed classes, helicopter and fighter-jet results close to each other and to dolphin. Probably this is due to the shape appearance of these object classes and to the common uniformity of the sky and water background. Moreover all the four legged animals (zebra, horse and dog) appear on the right side of the plot while the vehicles (car-side and motorbikes) are in the left bottom corner.

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

3.5 Comparison and Evaluation

In this section we analyze the connections of our KT algorithm to related work. This, on one side, indicates what are the common elements that define a successful transfer learning

method, and on the other, allows us to show the differences in the theoretical assumptions and problem setting of KT with respect to other existing approaches.

3.5.1 Ensemble Learning

We start with a remark regarding the comparison of our KT algorithm with classical machine learning approaches adopted for ensemble learning. The formulation of our KT method brings to the solution

$$\boldsymbol{w} = \sum_{j=1}^{J} \beta_j \hat{\boldsymbol{w}}_j + \sum_{i=1}^{N} a_i \phi(\boldsymbol{x}_i) \,, \tag{3.49}$$

which indicates that the final decision function has the same form of an ensemble method that combines $J+1$ classifiers. However, this is different from a genuine ensemble technique that puts together multiple classifiers trained independently because the Lagrange multipliers a_i are estimated under the influence of $\hat{\boldsymbol{w}}_j$ for $j = 1,\dots,J$ and their values are not the same as those estimated uniquely from the target training samples. Moreover, this solution is also different from that obtained by learning on all the source and target samples mixed together: this would require to keep always all the source data and to estimate the corresponding multipliers $\hat{a}_{ij}$ for all the samples of each source $i = 1,\dots,\hat{N}_j$, which are instead constant in our KT method.

3.5.2 Single Source Transfer

We describe in the following different transfer learning techniques that supposes the existence of a single source knowledge. The first two are based on transferring model parameters as our KT, while the last one is an instance transfer approach and exploits directly the prior knowledge samples.

Adaptive SVM (A-SVM).   This method has been originally presented in [180] and is based on substituting the usual regularizer of the SVM formulation with the adaptive version

$$\min_{\boldsymbol{w}} \| \boldsymbol{w} - \beta \hat{\boldsymbol{w}} \|^2 + C \sum_{i=1}^{N} \ell^H (\boldsymbol{w} \cdot \phi(\boldsymbol{x}_i), y_i) \,. \tag{3.50}$$

Although this approach is strictly related to our KT, it uses the hinge loss and does not get an automatic way to estimate the best β value. The focus of [180] was on domain adaptation with source and target task containing the same categories but with shifted distributions. Hence the weight given to prior knowledge was always fixed to $\beta = 1$ and a selective sampling strategy was applied to identify the complementary information necessary to adapt the source to the target task. The same problem formulation was used in [8] for cross category transfer in detection problems and the weight β applied to the prior knowledge model was obtained via

cross validation exploiting extra available target samples.

Projective Model Transfer SVM (PMT-SVM).   As already mentioned in section 3.2.1 the adaptive regularizer can always be expanded as $\|\boldsymbol{w} - \beta\hat{\boldsymbol{w}}\|^2 = \|\boldsymbol{w}\|^2 + \beta^2\|\hat{\boldsymbol{w}}\|^2 - 2\beta\|\boldsymbol{w}\|\|\hat{\boldsymbol{w}}\|\cos\theta$. To minimize the term $-2\|\boldsymbol{w}\|\|\hat{\boldsymbol{w}}\|\cos\theta$ we induce the transfer by maximizing $\cos\theta$, but at the same time $\|\boldsymbol{w}\|$ is encouraged to be large. This means that β, which should define the amount of transfer regularization, is actually a trade-off parameter between margin maximization and knowledge transfer. A different formulation for the same problem can be defined by considering the projection of $\boldsymbol{w}$ onto the prior knowledge separating hyperplane (see Figure 3.1) and minimizing consequently a term proportional to $\beta\|\boldsymbol{w}\|\sin\theta$. In this way augmenting the amount of transferring with a higher β would not penalize margin maximization. A complete formulation for this alternative approach has been presented in [8] and is named *Projective Model Transfer SVM* (PMT-SVM). Its objective function is

$$
\min_{\boldsymbol{w}} \ \|\boldsymbol{w}\|^2 + \beta\|\mathbf{R}\boldsymbol{w}\|^2 + C\sum_{i=1}^{N}\ell^H(\boldsymbol{w}\cdot\phi(\boldsymbol{x}_i), y_i)
$$
$$
\text{subject to} \quad \boldsymbol{w}^T\hat{\boldsymbol{w}} \geq 0 . \tag{3.51}
$$

where $\mathbf{R}$ is the projection matrix

$$
\mathbf{R} = \mathbf{I} - \frac{\hat{\boldsymbol{w}}\hat{\boldsymbol{w}}^T}{\hat{\boldsymbol{w}}^T\hat{\boldsymbol{w}}} , \tag{3.52}
$$

β controls the amount of transfer regularization and C controls the relative importance of the hinge loss with respect to the regularizers. $\|\mathbf{R}\boldsymbol{w}\|^2 = \|\boldsymbol{w}\|^2\sin^2\theta$ is the squared norm of the projection of the vector $\boldsymbol{w}$ onto the source hyperplane $\hat{\boldsymbol{w}}$, and $\boldsymbol{w}^T\hat{\boldsymbol{w}} \geq 0$ constrains $\boldsymbol{w}$ to the positive halfspace defined by $\hat{\boldsymbol{w}}$. Although the formulation is convex and can be minimized using quadratic optimization [8], this approach does not provide a proper way to define the value of the weight β which is chosen by cross validation, again relying on extra available target samples.

TrAdaBoost: boosting for Transfer Learning.   An alternative way to exploit prior knowledge is to keep all the source samples and use them when learning on the target task. It is possible to do it by extending the AdaBoost learning framework which aims to boost the accuracy of a weak learner by carefully adjusting the weight for each training instance. A linear combination of weak classifiers is then used to build the final decision function. Specifically *TrAdaBoost* [41] considers a mix of source and target data in training and is based on a mechanism which decreases the weights of the source instances in order to weaken their impact. In each iteration round, a source training instance which is mistakenly predicted may likely conflict with the target data. Thus its training weight is reduced such that in the next round it will affect the learning process less than the current round. Finally the instances with large training weights help the learning algorithm to train better classifiers. For a formal description of the method,
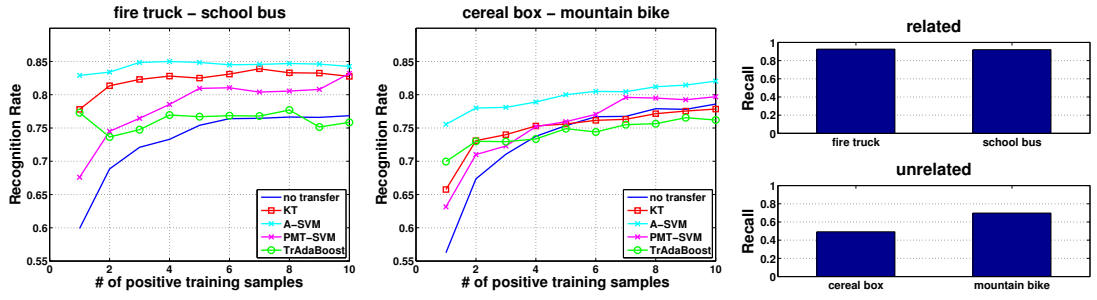
Figure 3.6 – Left and middle: recognition rate as a function of the number of positive training samples. Each experiment is defined by considering in turn one of the classes as target and the other as source and extracting randomly the training samples ten times. The final results are obtained as average over all these runs. The title of each plot indicate which classes have been used. Right: the histogram bars represent the recall produced by the model of the source class (indicated on the x-axis) when used to classify directly on the other class considered as target.

we refer to the Algorithm 1 in the Appendix (considering $J = 1$).

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

We used the MATLAB code of A-SVM and PMT-SVM provided by their authors[3] slightly modifying it to introduce the weights ζ_i for $i = 1, \ldots, N$ in the corresponding loss function, so to have a fair comparison with our KT. The original implementation considered the linear kernel, thus we chose $K(\boldsymbol{x}, \boldsymbol{z}) = \boldsymbol{x} \cdot \boldsymbol{z}$ for all the experiments together with the SIFT feature descriptors. For TrAdaBoost we set the number of boosting iterations to $M = 20$ [4]. Since A-SVM and PMT-SVM do not provide a specific technique to define automatically the β value when no extra validation target samples are available, we decided to simply tune it on the test set, showing the best result that can be obtained[5].

The results are shown in Figure 3.6. In the related (left plot) case all the transfer learning methods show better performance than learning from scratch with different extent.
The results of our KT are significantly better than those of no transfer and PMT-SVM ($p \leq 0.01$). Only for 10 positive training samples PMT-SVM and KT produce comparable results. KT also outperforms TrAdaBoost for all the training steps ($p \leq 0.01$) except the first one, where they are statistically equivalent. Finally, the difference between KT and A-SVM is not significant: since

---

3. http://www.robots.ox.ac.uk/~vgg/software/tabularasa/
4. We tried $M = \{10, 20, 50\}$ iterations and we report here only the best results.
5. We varied β in the set $\{0.01, 0.1, 0.2, 0.4, 0.6, 0.8, 1, 10, 100, 1000\}$ giving to the methods the maximum freedom also allowing values bigger than 1.

the β parameter for A-SVM is tuned on the test set, this indicates that KT is autonomously able to identify the optimal weight to assign to prior knowledge. The bias of A-SVM towards the best possible recognition rate is evident in the case of unrelated classes (middle plot) where it is the only method to outperform no transfer along all the steps. The other knowledge transfer approaches show better results than no transfer only for less than three positive training samples ($p \leq 0.05$), becoming then statistically equivalent to learning from scratch.

The histogram bars on the right in Figure 3.6 show the recall produced by each source model when used directly to classify on the target task. This indicates the prior knowledge capability in recognizing the new object without adaptation and it is clearly lower for unrelated than for related classes.

3.5.3 Multiple Sources Transfer.

When more than one source knowledge is available, there are three main strategies that a transfer learning method can consider. Two extreme solutions consist in either selecting only one source, evaluated as the best for the target problem, or averaging over all of them supposing that they are all equally useful. The third strategy considers the intermediate case where only some of the source knowledge are helpful for the target task and consists in selecting them by assigning to each a proper weight. In the following we present existing approaches that adopt one of the first two strategies: up to our knowledge, only our KT method is based on the third selective technique.

MultiSourceTrAdaBoost: boosting by transferring samples. An extension to the TrAdaBoost approach in the case of multiple available sources has been presented in [183]. The method *MultiSourceTrAdaBoost* considers one source set at the time, combining it with the target set and defining a candidate weak classifier. The final classifier is then chosen as the one producing the smallest training target classification error by automatically selecting the corresponding best source. Here the weighting update of the source training instances is the same as TrAdaBoost, and the weighting update of the target training instances is the same as in the original AdaBoost. Finally the method reduces to TrAdaBoost in case of a single source set. A formal description of this framework is given in the Appendix in Algorithm 1.

TaskTrAdaBoost: boosting by transferring models. The previous method corresponds to an instance transfer approach and its authors proposed also a parameter transfer variant. *TaskTrAdaBoost* [183] consists of two steps. Phase I deploys traditional AdaBoost separately on each source task to get a collection of candidate weak classifiers. Only the most discriminative are stored by asking that the weight assigned by the boosting process to each classifier is greater than a certain threshold τ : this guarantees to avoid overfitting. Phase II is again an AdaBoost loop over the target training data where at each iteration the weak classifier is extracted from the set produced in the previous phase. The choice is done on the basis of

the minimal classification error produced on the target training set. Finally the update of the target training instances drives the search of the next most useful source knowledge to transfer. The two phases are formally described in the Appendix respectively in Algorithm 2 and 3.

Single KT. Our KT algorithm chooses the best set of weights for all the prior knowledge models at once on the basis of the loss function defined in (3.30) and weighted according to (3.36). An alternative approach can be defined adopting a logistic loss function [161]:

$$\ell(\tilde{y}_i, y_i) = \zeta_i \frac{1}{1 + \exp\{-10(\tilde{y}_i - y_i)\}} \ . \tag{3.53}$$

If we consider one single source knowledge j at the time, the corresponding loss $\ell_j(\tilde{y}_i, y_i)$ will depend on the difference $(\tilde{y}_i - y_i) = \left(\frac{a'_i}{P_{ii}} - \beta_j \frac{a''_{ij}}{P_{ii}} \right)$ for all $i = 1, \ldots, N$. Although this formulation results in a non convex objective function with respect to β_j, it is always possible to evaluate (3.53) for a finite set $\mathscr{S}$ of weights [6]. We can store for each source the value $\min_{\mathscr{S}}\{\sum_i \ell_j(\tilde{y}_i, y_i)\}$, and then compare all the results to identify the best prior knowledge model and the best weight value to assign. We call this variant of our method *Single-KT*.

Average Prior Knowledge. The first knowledge transfer approach able to perform one-shot learning on computer vision problems was presented in [59]. The proposed method is defined in the Bayesian setting by extracting *general knowledge* from previously learned categories and representing it in the form of a prior probability density function in the space of model parameters. Given a new target training set, no matter how small, the source knowledge is updated and the produced posterior density is then used for classification. This approach does not make any assumption on the reliability of the prior knowledge, which is always considered as an average over all the known classes. The algorithm structure is strictly related to the part-based model descriptors and neither the code nor the feature used for the experiments in [59] have ever been publicly released. However, following the proposed main idea, any transfer learning method that originally considers the existence of a single source task can be extended to the case of multiple sources by relying on the average of all the prior known models.

Experiments. Here we show a benchmark evaluation of our KT algorithm against its Single-KT version, MultiSourceTrAdaBoost and TaskTrAdaBoost. Following the basic idea of [59] we also use A-SVM as baseline supposing to consider the average of all the prior models as source knowledge, thus $\hat{w} = \frac{1}{J} \sum_{j=1}^{J} \hat{w}_j$ and $\beta = 1$.

We used the same setting adopted for the experiments with a single source, considering the linear kernel, SIFT features and two randomly extracted sets of 10 and 20 classes from Caltech-256. In particular the second set is obtained by adding an extra group of 10 classes to the

---

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

Figure 3.7 – Left and middle: recognition rate as a function of the number of positive training samples. Each experiment is defined by considering in turn one of the classes as target and the others as sources and extracting randomly the training samples ten times. The final results are obtained as average over all these runs. The title of each plot indicate which classes have been used. Right: average norm of the difference between two $\boldsymbol{\beta}$ vectors obtained for a pair of subsequent training steps.

first one [7]. For the boosting approaches we fixed the number of iterations to $M = 20$ and in Phase I of TaskTrAdaBoost we chose the threshold $\tau = 0.6$ [8]. The results are shown in Figure 3.7. In both the experiments our KT approach clearly outperforms Single-KT and the two boosting methods ($p \leq 0.01$), besides producing better results than learning from scratch ($p \leq 0.01$). Moreover, for very few samples, properly weighting each prior knowledge source with KT is better ($p \leq 0.05$) than averaging over all the known models as done by A-SVM: the two approaches are equivalent only after five positive training samples with 10 classes and respectively three positive training samples for 20 classes.

For any method that chooses only one source model in transferring, each time there is a change in the selected source, the behavior of the algorithm might change. This indicates low stability. A recent work has shown that the more stable is an algorithm, the better is its generalization ability [22]. The plot on the right in Figure 3.7 shows the comparison of KT with its Single-KT version in terms of stability. The best β_j value chosen by Single-KT can be considered as an element of the full $\boldsymbol{\beta}$ vector where all the remaining elements are zero. For each pair of subsequent steps in time, corresponding to a new added positive training sample, we calculate the difference between the obtained $\boldsymbol{\beta}$ both for KT and Single-KT. From the average norm of these differences it is evident that choosing a combination of the prior known models for transfer learning is more stable than relying on just a single source (lower average variation in the vector $\boldsymbol{\beta}$).

---

7. First group of classes: kayak, hot-air-balloon, blimp, golf-ball, fighter-jet, rotary-phone, mandolin, computer-monitor, microwave, yarmulke. The second group is: rainbow, horse, radio-telescope, cormorant, boom-box, cereal-box, fire-hydrant, toaster, fern, starfish.

8. We run experiments with $\tau = \{0.4, 0.6, 0.8\}$. We report here the best obtained results.
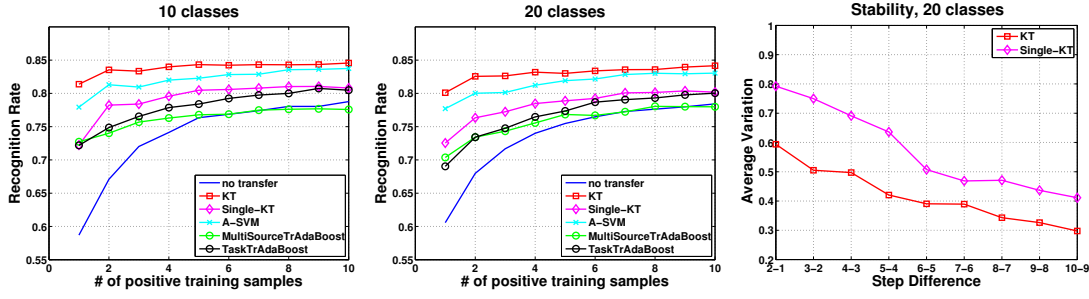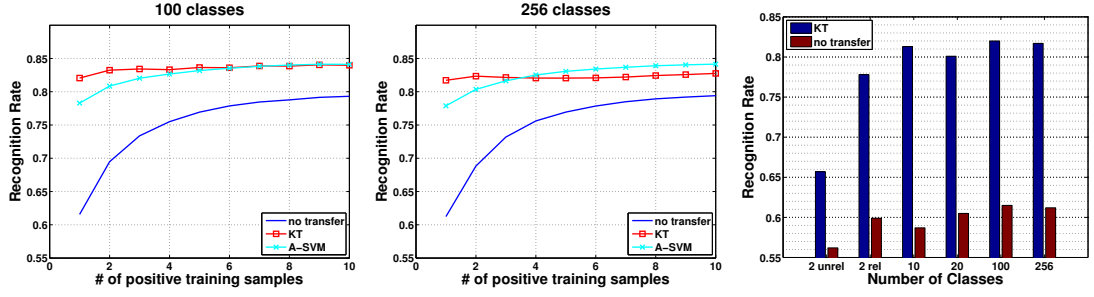
Figure 3.8 – Left and middle: recognition rate as a function of the number of positive training samples. Each experiment is defined by considering in turn one of the classes as target and the others as sources and extracting randomly the training samples ten times. The final results are obtained as average over all these runs. The title of each plot indicate which classes have been used. Right: one shot learning performance of KT and no transfer when varying the number of prior known object categories.

3.5.4 Increasing Prior Knowledge

For any open-ended learning agent the number of known object categories is expected to grow in time. This on one side is an advantage because a large variety in prior knowledge helps when learning something new. On the other side, a high number of sources gives rise to a scalability problem for any transfer learning method due to the necessity of checking each of them to evaluate the reliability for the new task. Specifically, for 10^2 source sets the boosting methods described in the previous section becomes extremely expensive in computational terms, considering the multiple iterations to run for each source [9].

We performed experiments with 100 and 256 object classes [10] considering the full Caltech-256 dataset reporting the result of KT, no transfer and A-SVM with average prior knowledge in Figure 3.8. In both cases, properly choosing the weights to assign to each source pays off with respect to average over all the sources only for very few available positive training samples. In particular for 100 classes, after an initial phase where KT is better than A-SVM ($p \leq 0.05$), they perform equally with more than five positive samples. Instead, using the average knowledge is the best choice for 256 classes with more than three positive samples ($p \leq 0.05$). We can conclude that with enough training samples, and a rich prior knowledge set, the best choice is to not neglect any source information.

---

9. Indeed the paper which presented these methods considered a maximum of 5 sources [183].

10. To get the set of 100 classes we added another group of 80 random selected categories to the 20 used in the previous experiment: house-fly, computer-mouse, snail, snake, bonsai, pci-card, roulette-wheel, palm-tree, bowling-ball, cowboy-hat, spoon, scorpion, people, tower-pisa, lathe, dice, mattress, eiffel-tower, covered-wagon, bear, human-skeleton, basketball-hoop, toad, vcr, frog, tomato, teddy-bear, buddha, hourglass, conch, windmill, hot-dog, frisbee, tennis-shoes, faces-easy, harmonica, fireworks, duck, football-helmet, breadmaker, lightning, self-propelled-lawn-mower, sextant, ladder, mailbox, camel, hamburger, bulldozer, bathtub, dumb-bell, ipod, unicorn, chess-board, traffic-light, galaxy, menorah, screwdriver, spider, tweezer, head-phones, car-tire, goose, praying-mantis, necktie, car-side, sushi, calculator, umbrella, american-flag, mars, kangaroo, golden-gate-bridge, iris, horseshoe-crab, soda-can, helicopter, bowling-pin, watermelon, soccer-ball, backpack.

We can expect that with a growing prior knowledge set, also the probability to find a useful source for the target task increases. To verify this behavior we focus on the KT results obtained with a single positive image. The one-shot performance obtained in the previous experiments for 2 unrelated classes, 2 related classes, random sets of 10, 20, 100 classes plus the final full set of 256 objects are summarized in Figure 3.8 (right). Although some small oscillation due to the specific group of classes considered, it is clear that by increasing the number of available sources of one order of magnitude the one-shot recognition rate obtained with KT grows. After an evident gain obtained by passing from 10^0 to 10^1 classes, the difference becomes less evident from 10^1 to 10^2 classes.

3.5.5 Heterogeneous Sources

Any parameter transfer learning method supposes that, when the learning process on the target task starts, the corresponding one on the sources has already ended with the definition of a model for each of them. Hence unless the learning space has been fixed from the beginning, there is no direct control on the source learning approach. This means that the sources may be heterogeneous in feature descriptors and models, adding an extra challenge for a knowledge transfer approach. Among the methods that we considered in the previous sections, the only one that allows the use of heterogeneous sources is TaskTrAdaBoost. In practice, each source weak classifier could have been learned with a different method since in the transfer process only the prediction on the target training samples is used. The multiK-KT version of our transfer learning method presented in section 3.4.3 can also tackle this heterogeneity. We benchmark here their performance.

We consider the random set of ten classes already used in the previous section and we run one-class-out experiments with each time one of the classes as target and the remaining nine as sources. For each source we suppose to have already learned an SVM model with SIFT descriptors and Gaussian kernel where the γ parameter is set to the mean of the pairwise distances among the samples. This means that each source model lives in its own specific feature space. TaskTrAdaboost in each boosting iteration simply chooses one of the source models, while multiK-KT learns the target task in the composed space defined by all the sources and obtained on the basis of the sum kernel. Figure 3.9 shows that multiK-KT outperforms TaskTrAdaBoost ($p \leq 0.01$) besides obtaining better results than learning from scratch.

3.5.6 Increasing Number of Samples

Transfer learning has its maximum effectiveness in the small sample scenario in comparison to learning from scratch. However, it is also interesting to evaluate the performance of a knowledge transfer approach when the number of available training instances increases, thus checking its asymptotic behavior (see Figure 2.3).

We repeated the experiments on the full Caltech-256 dataset considering the four features

10 classes, Heterogeneous sources
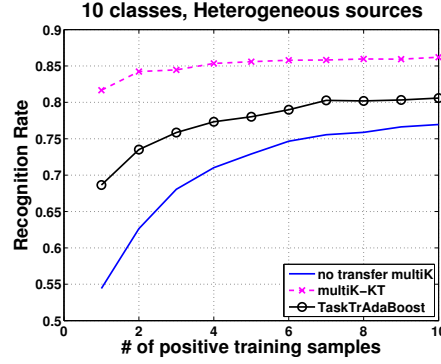
Figure 3.9 – Recognition rate as a function of the number of positive training samples. Each source model is defined by using a Gaussian kernel with a different γ parameter. In turn every class is considered as target and the remaining nine as sources. The presented results are the average over all these runs repeated ten times with random extraction of the training samples.

already used in section 3.4.2 and $\{1,5,10,30,50\}$ positive training samples with a fixed set of 50 negative training samples and 60 (30 positive and 30 negative) test instances. We also run analogous experiments on the Animals with Attributes (AwA) [89] and IRMA [163] dataset with several descriptors. For all the experiments we used the average Gaussian kernel. This setting allows us to show the effectiveness of our KT method independently of the dataset used, while considering different features.

The AwA dataset contains 50 animal classes and has been released with several pre-extracted feature representations for each image [11]. From the full set of categories we extracted the six sea mammals (killer whale, blue whale, humpback whale, seal, walrus and dolphin) and used them to define the background class. We used three of the precomputed descriptors for our experiments: color histogram, PHOG and SIFT. We considered sets of $\{1,5,10,30,50,60\}$ positive training samples with a fixed set of 60 negative training samples and 60 (30 positive and 30 negative) test instances.

The IRMA database [12] is a collection of x-ray images presenting a large number of rich classes defined according to a four-axis hierarchical code [94]. We decided to work on the 2008 IRMA database version [48], just considering the third axis of the code: it describes the anatomy, namely which part of the body is depicted, independently to the used acquisition technique or direction. A total of 23 classes with more than 100 images were selected from various sub-levels of the third axis, 3 of them were used to define the background class [13]. We considered set of

---

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

12. http://phobos.imib.rwth-aachen.de/irma/datasets_en.php

13. 213-nose area (242 images), 230-neuro area (365 images), 310-cervical spine (508 images), 320-thoracic spine (279 images), 330-lumbar spine (540 images), 411-hand finger (325 images), 414- left hand (541 images), 415-right hand (176 images), 421-left carpal joint (124 images), 441-left elbow (114 images), 442-right elbow (105 images), 463-right humero-scapular joint (146 images), 610-right breast (144 images), 620-left breast (155 images), 914-left foot(146 images), 915-right foot (139 images), 921-left ankle joint (192 images), 922-right ankle joint (229 images), 942-left knee (231 images), 943-right knee (222 images). Three classes used for background: 700-abdomen (219

Figure 3.10 – Recognition rate as a function of the number of positive training samples. Each experiment is defined by considering in turn one of the classes as target and the others as sources and extracting randomly the training samples ten times. The final results are obtained as average over all these runs. The title of each plot indicate which dataset have been used.
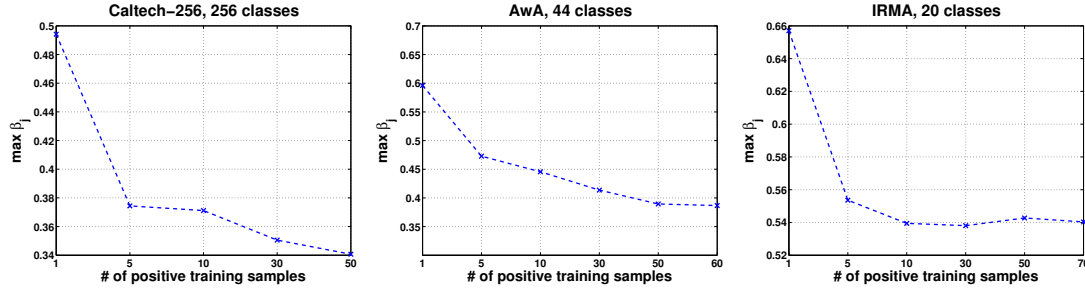
Figure 3.11 – Maximum value over the elements of the $\boldsymbol{\beta}$ vector averaged over the classes and the splits correspondent to the experiments reported in Figure 3.10.

$\{1, 5, 10, 30, 50, 70\}$ positive training samples with a fixed set of 70 negative training samples and 60 (30 positive and 30 negative) test instances. As features we used the global pixel-based and local SIFT-based descriptors following the experimental setup in [164].

The results for all these experiments are reported in Figure 3.10. Although it is clear the gain of KT with respect to learning from scratch for limited available data, in general this advantage disappears when the number of positive training samples reaches 50. Figure 3.11 indicates that the weights associated to prior knolwdege decrease, but there is always at least one β_j value different from zero. The absence of the asymptotic advantage was to be expected for KT and can be justified in theoretical terms: when the number of training samples increases, the adaptive regularization loses its relevance and the problem reduces to learning from scratch.

---

images), 800-pelvis (263 images), 500-chest (4611 images).

3.6 Discussion

A learning system able to exploit prior knowledge when learning something new should rely only on the available target information for choosing from where and how much to transfer. To be autonomous it should not need an external teacher providing either information on which is the best source to use, or extra training samples.

In this chapter we presented our KT method: it is a LS-SVM-based approach with a principled technique to weight source models, selecting a subset of them as the most useful when facing a novel target task. We discussed the properties of this approach: we showed its flexibility in terms of the specific form of the known sources and we noticed its stability with respect to approaches able to rely only on a single source. Moreover the weights assigned to the prior knowledge set proved to be meaningful in terms of the semantic relation among the considered classes. The results of extensive experiments demonstrated the effectiveness of KT for object categorization problems with respect to other existing transfer learning methods. Finally, we analyzed the behaviour of KT both when the number of source sets and available training samples increases: the obtained performance is always better or equal than learning from scratch independently to the relatedness among the classes, the features and the dataset used.

If we think of scaling up the learning problem, there are three issues that may arise.

When increasing the number of prior known models, the linear dependence of the KT computational complexity with respect to the dimensionality of the source set can be eventually reduced. One possible solution is to organize the sources in a hierarchy such that the transfer process can focus only on few general nodes at the beginning, investigating the specific classes inside each of them only in a second stage.

KT is defined as a batch approach and the learning process restarts every time a new sample is added to the training set. The corresponding scalability problem that arises due to the increasing number of training samples can be overcome by casting KT in an online learning framework (see chapter 6).

Finally KT has been specifically defined for binary problems. Increasing the number of classes to be distinguished means passing to the multiclass setting. A possible extension for multiclass problems for domain adaptation is presented in the next chapter.

4 Extension to Domain Adaptation

In real learning scenarios it is common to encounter regression and multiclass classification problems where the training and the test set differ due to a domain shift. This chapter presents how to enlarge the KT algorithm for domain adaptation purposes and shows the effectiveness of the obtained approach on two practical applications. One focuses on biomedical signals recorded on different subjects for the classification of hand movements and the prediction of the applied force in grasping. The second considers visual object recognition for images downloaded from the web and acquired with different cameras, where the main causes of domain shift are image resolution, lighting conditions, background and viewpoint.

4.1 Domain Adaptation Problems

It can happen that the effort in learning how to solve a task results vain when facing the same task during deployment. The reason in most of the cases is the existence of a domain shift between the source problem considered in training and the actual target test. For example, learning morphological, syntactic and semantic information on text resources mostly based on newspapers results in a poor performance when annotating biomedical texts or transcription of conversations. Similarly, a robot trained to move in outdoor settings may fail to reach its target in indoor conditions.

As already discussed in section 2.1.1, domain adaptation aims at adjusting a classifier or a regression model trained on a source domain for use in a target domain. Despite the difference with transfer learning where the source and the target do not share the same label set, it is still necessary to evaluate which part of the previous knowledge should be kept and which should be updated.

The adaptive method presented in the previous chapter was used to transfer information across visual objects with a specific binary focus for object-vs-background problems. Nevertheless KT is not constrained to this specific scenario and can be used easily for domain adaptation in semi-supervised settings. Generally more than two classes are involved in re-

alistic annotation and classification tasks presenting a domain shift. For this reason in this chapter we present an extension of our KT algorithm to multiclass problems and regression modeling.

4.2 KT Algorithm Extensions

We use here the mathematical framework already introduced in the previous chapter: we denote with $\boldsymbol{x}_i$ an input vector and y_i its associated output. For simplicity we report below the KT binary learning problem from which we start:

$$\min_{\boldsymbol{w},b} \frac{1}{2} \left\| \boldsymbol{w} - \sum_{j=1}^{J} \beta_j \hat{\boldsymbol{w}}_j \right\|^2 + \frac{C}{2} \sum_{i=1}^{N} \xi_i^2$$

$$\text{subject to} \quad y_i = \boldsymbol{w} \cdot \phi(\boldsymbol{x}_i) + b + \xi_i \quad \text{for} \quad i = 1,\dots,N \,, \tag{4.1}$$

considering N target training samples and $j = 1,\dots,J$ source models. As before, small and capital bold letters indicate respectively column vectors and matrices. Moreover, we use here a subscript to indicate a specific column of a matrix: e.g. $\boldsymbol{A}_i$ is the i-th column of the matrix $\boldsymbol{A}$.

4.2.1 Multiclass Classification

The KT algorithm can be easily generalized to multiclass classification problems in the hypothesis of a set of $g = 1,\dots,G$ classes, fixed for both sources and target task. Consider the model $(\boldsymbol{w}_g, b_g)$ that discriminates class g considered as positive, from all the others considered negative, and repeat it for each class (1-vs-All). The predicted class for sample i is then obtained by $\text{argmax}_g \{\boldsymbol{w}_g \cdot \phi(\boldsymbol{x}_i) + b_g\}$.

We define the matrix $\boldsymbol{Y} \in \mathbb{R}^{G \times N}$ composed by the columns $\boldsymbol{Y}_i$, where for each sample i the vector $\boldsymbol{Y}_i$ has all the components equal to -1 except for the y_i-th that is equal to 1. In the same way, the matrix $\hat{\boldsymbol{Y}}^j$ is composed by the columns $\hat{\boldsymbol{Y}}_i^j$ that contain the predictions generated by a known multiclass source j on the sample i. For each sample i we also obtain a vector of G leave-one-out predictions, we indicate it with $\tilde{\boldsymbol{Y}}_i$ and, on the basis of (3.28), it is easy to show that

$$\tilde{\boldsymbol{Y}}_i = \boldsymbol{Y}_i - \frac{\boldsymbol{A}_i'}{P_{ii}} + \beta_j \frac{\boldsymbol{A}_i''^j}{P_{ii}}, \tag{4.2}$$

where

$$[\boldsymbol{A}', \boldsymbol{b}'] = [\boldsymbol{Y}, \boldsymbol{0}] \boldsymbol{P}^T \,, \tag{4.3}$$

$$[\boldsymbol{A}''^j, \boldsymbol{b}''^j] = [\hat{\boldsymbol{Y}}^j, \boldsymbol{0}] \boldsymbol{P}^T \,. \tag{4.4}$$
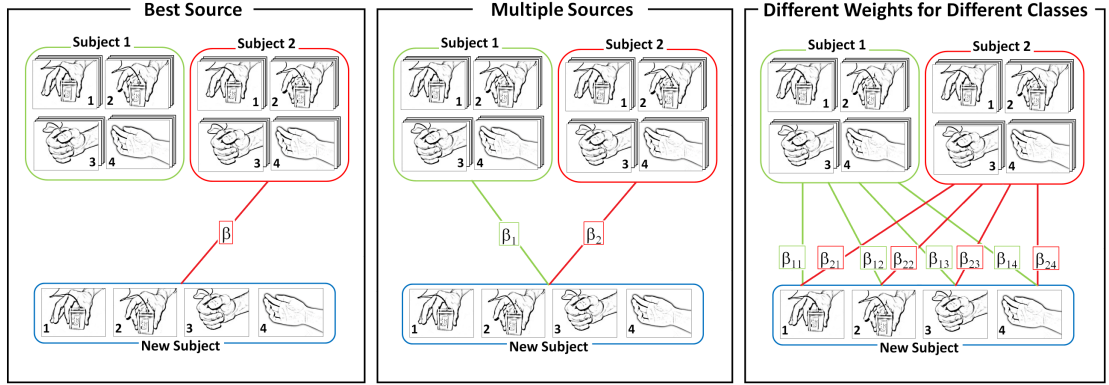
Figure 4.1 – Three versions of the KT multiclass extension for domain adaptation problems. Left: choose only the best known source and use its reweighted models as starting point for learning. Center: consider a linear combination of all the known sources with equal weight for all the classes of each source. Right: consider again a linear combination of all the sources but assign a different weight for each class. The figure presents the specific application to surface electromyography signals recorded from multiple subjects while performing three grasp movements and in the rest condition [166]. See section 4.3 for details on this application.

Here A', $A''^j \in \mathbb{R}^{G \times N}$ and b, $0 \in \mathbb{R}^G$. In case of multiple sources combined linearly, we get

$$\tilde{Y}_i = Y_i - \frac{A'_i}{P_{ii}} + \sum_{j=1}^{J} \beta_j \frac{A''^j_i}{P_{ii}} \; . \tag{4.5}$$

There are three possible solutions to weight prior knowledge multiclass models and rely on them for the new learning problem. We describe them below and report a general scheme in Figure 4.1.

Best Source. Following the strategy used for Single-KT described in section 3.5.3, a first solution could be to consider the logistic loss

$$\ell(Y_i, \tilde{Y}_i) = \frac{1}{1 + \exp(-10(\max_{g \neq y_i}\{\tilde{Y}_{gi}\} - \tilde{Y}_{y_i i}))} \; , \tag{4.6}$$

and to evaluate it separately for each of the $j = 1, \ldots, J$ pre-trained models on the basis of (4.2), varying β_j with small steps in $[0,1]$. The minimal result identifies both the best known source for adaptation and the corresponding weight. Still, we already know that this approach is non-convex thus reaching the global optimum is not computationally efficient. This technique is schematically depicted in Figure 4.1 (left) and we name the final corresponding domain adaptation approach *Best-Adapt*.

Multiple Sources. To consider multiple prior knowledge sources we propose to use (4.5) in

the convex multiclass loss [36]:

$$\ell(\boldsymbol{Y}_i, \tilde{\boldsymbol{Y}}_i) = \max \left\{ 0, 1 - \tilde{Y}_{y_i i} + \max_{g \neq y_i} \{ \tilde{Y}_{g i} \} \right\} , \qquad (4.7)$$

with the final objective function

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{N} \ell(\boldsymbol{Y}_i, \tilde{\boldsymbol{Y}}_i) \quad \text{subject to} \quad \|\boldsymbol{\beta}\|_2 \leq 1, \quad \beta_j \geq 0 . \qquad (4.8)$$

The optimization process can be implemented simply by using a projected sub-gradient descendent algorithm (similarly to what presented in section 3.3.4) , where at each iteration $\boldsymbol{\beta}$ is first projected onto the L_2-sphere, $\|\boldsymbol{\beta}\|_2 \leq 1$ and then onto the positive semi-plane [166]. The pseudo-code is in the Appendix in Algorithm 5. Figure 4.1 (center) describes this solution and the corresponding domain adaptation approach is named *Multi-Adapt.*

Different Weights for Different Classes. Until now we considered techniques which assign a unique weight to each known source. This means that, the whole set of 1-vs-All pre-trained models of a source are equally weighted. However, in case of two sources, when learning the model for one target class, it may be useful to give more weight in adaptation to the first source than to the second, while it could be the opposite when learning the model for a different class. Hence, to have one more degree of freedom and decide the adaptation specifically for each class, we enlarge the set of weight parameters introducing the matrix $\mathbf{B} \in \mathbb{R}^{J \times G}$ where each row j contains the vector $\boldsymbol{\beta}_j^T$ with G elements, one for each class [166]. This approach is described in Figure 4.1 (right) and we name the corresponding domain adaptation method *Multi-perclass-Adapt.*

The optimization problem is analogous to the one described in (4.8), with a change in the constraints. Each class problem is now considered separately, so we have G conditions, one for each of the columns $\mathbf{B}_g$ of the $\mathbf{B}$ matrix, we impose $\|\mathbf{B}_g\|_2 \leq 1$ and $B_{ji} \geq 0$.

4.2.2 Regression

The KT method, by exploiting the square loss function, aims at minimizing the mean square error between the predicted output for $\boldsymbol{x}_i$ and the real y_i . Thus the square loss, apart from providing the possibility to estimate easily the relevance of pre-existent source knowledge, makes KT directly suitable for transfer learning and domain adaptation regression problems.

Best Source. By using (3.28) we can always evaluate the square difference between the leave-one-out prediction produced by relying on the source j and the correct y_i of each sample:

$$(y_i - \tilde{y}_i)^2 = \left(\frac{a'_i}{P_{ii}} + \beta_j \frac{a''_{ij}}{P_{ii}} \right)^2 .$$

By summing over all the samples, we get a quadratic function in β_j and the minimum is

obtained for:

$$\beta_j = \frac{\sum_{i=1}^{N} \frac{a_i'}{P_{ii}} \frac{a_{ij}''}{P_{ii}}}{\sum_{i=1}^{N} \left(\frac{a_{ij}''}{P_{ii}}\right)^2} \; .$$

(4.9)

We can impose the constraint $\beta_j \geq 0$ by just enforcing $\beta_j = 0$ whenever it is negative. Hence in this case we do not need any optimization procedure, the best weight to assign to each prior knowledge is obtained by a closed formula. Once computed the β_j for $j = 1, \ldots, J$, by comparing all of them we can identify the best known source j^* to use for adaptation and impose $\beta_{\{j \neq j^*\}} = 0$ when learning the regression model on the target.

Multiple Sources. To take advantage from all the available pre-trained models at once, we can consider their linear combination. The square difference between the leave-one-out prediction $\tilde{y}_i$ and the correct y_i is

$$(y_i - \tilde{y}_i)^2 = \left(\frac{a_i'}{P_{ii}} + \sum_{j=1}^{J} \beta_j \frac{a_{ij}''}{P_{ii}}\right)^2 \; .$$

(4.10)

Adding also the condition $\|\boldsymbol{\beta}\|_2 \leq 1$, we can find the best $\boldsymbol{\beta}$ vector which minimizes (4.10) with a standard Quadratically Constrained Quadratic Program (QCQP) solver.

Following the same naming policy used in the previous section, we indicate the described regression approaches respectively as *Best-Adapt* and *Multi-Adapt* [166].

4.3 Application to Biological Signals for Hand Prosthetics

Domain adaptation problems are often related to personalization of pre-existent standard tools. A typical example is that of spam filters that might be different for two different users [73]. In this section we focus on the practical problem of hand prosthetics with the final aim to exploit the experience gained on several known subjects to pre-train a robotic hand prosthesis before shipping it to a patient.

In the prosthetics/rehabilitation robotics community it is generally understood nowadays [186, 113, 127] that advanced hand prostheses are in dire need of accurate and reliable control schema to make them easy to use. Together with excessive weight and low reliability, *lack of control* is the main reason why 30% to 50% of upper-limb amputees do not use their prosthesis regularly [6].

The prosthesis is generally controlled by using two surface electromyography (sEMG) electrodes and complex sequences of muscle contraction impulses [21, 32, 130]. The patient must get acquainted and proficient with this "language" if (s)he wants to achieve a minimum control over the prosthesis. In the last years, the use of more electrodes (typically more than five) and the application of machine learning techniques on the recorded signals have shown

promising results in detecting what the patient *wants* to do and to enforce it in a more natural way [126, 113, 127, 110, 111, 112, 102]. The word "natural" here is still quite a misnomer, as it refers to the choice among a finite number of predefined hand configurations; but this kind of control is still much more natural than before, as each posture is achieved by configuring one's muscle remnants as they would be if the missing limb were still there. Recent results on amputees indicate that even long-term patients can generate rather precise residual activity: there is essentially no statistically significant difference in the performance attained by learning on signals recorded from trans-radial amputees and intact subjects [27, 158].

Researchers have mainly concentrated so far on increasing the accuracy of sEMG classification and/or regression, but in general, a finer control implies a longer training period. A desirable characteristic would be to shorten the training time.

Anatomical similarity among humans intuitively suggests that good statistical models built in the past might be proficiently reused when training a prosthesis for a new patient. This idea cannot be naïvely enforced with standard learning techniques, as shown at least in [26], where cross-subject analysis (i.e., using a model trained on a subject to do prediction on a new subject) shows poor performance. Hence, more refined adaptive methods are necessary. One possible approach consists in combining the target samples with the source samples properly reweighted on the basis of the domain relatedness. This solution is adopted in [155] and the sensibility of the method to the weighting parameter is evaluated empirically, but how to choose it is left as an open problem.

4.3.1 Experiments

We run several set of experiments to evaluate the proposed model adaptation technique for regression and classification. We present below the experimental setting, followed by a description of the two databases used and the obtained results on each of them.

Our working assumption is to have $J-1$ pre-trained models stored in memory and trained off-line on data acquired on $J-1$ different subjects. When the prosthetic hand starts to be used by subject J the system begins to acquire new data. Given the differences among the subject's arms and as well in the placement of the electrodes, these new data will belong to a new probability distribution, in general different from the $J-1$ previously modeled and stored. Still all the subjects perform the same grasp types, thus it is reasonable to expect that the new distribution will be *close* to at least one of those already modeled.

To simulate this scenario, we applied a leave-one-subject out strategy: in turn one of the subjects for which we have data recordings is used as target, while the model learned on the others are used as sources. This procedure is repeated J times. We consider three baseline approaches as reference

No-Adapt: it is plain LS-SVM using only the new data for training, as it would be in the standard scenario without adaption.
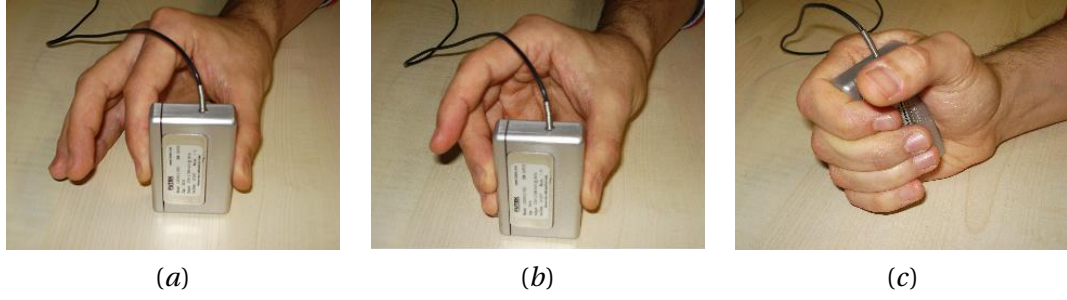
Figure 4.2 – The three different grasp types recorded in the hand posture and force signal dataset [26]: (*a*) index precision grip; (*b*) other fingers precision grip; (*c*) power grasp. Reproduced from [26].

Prior Average: consists in using only the pre-trained models without updating them with the new training data. We evaluate their average performance.

Prior Start: this corresponds to the performance of the best model chosen by Best-Adapt at the first training step.

Prior Test: this is the result that can be obtained *a posteriori* comparing all the prior knowledge models on the test set and choosing the best one.

As a measure of performance, for classification we use the standard classification rate. For regression, the performance index is the correlation coefficient evaluated between the predicted force signal and the real one. Although we minimized the mean square error in the regression learning process, the choice of the correlation coefficient is suggested by a practical consideration. When driving a prosthesis, or even a non-prosthetic mechanical hand, we are not interested in the absolute force values desired by the subject. Mechanical hands usually cannot apply as much force as human hands do (for obvious safety reasons), or they could be able to apply *much more* force than a human hand can (e.g. in teleoperation scenarios). As already done, e.g. in [28, 27, 26], we are rather concerned with getting a signal which is *strongly correlated* with the subject's will. The significance of the comparisons between the methods is evaluated through the sign test.

To build the pre-trained models we used the standard SVM algorithm. All the parameters to be set during training (C and γ of the Gaussian kernel) were chosen by cross-validation. Specifically when the subject j^* is the new target problem, this is excluded form the dataset and the parameters are chosen over the remaining source set $\mathcal{J} = \{1, \ldots, J \setminus j^*\}$ looking for the values that produce on average the best recognition rate or correlation coefficient by learning on each subject j in $\mathcal{J}$ and testing on $\mathcal{J} \setminus \{j^*, j\}$.

Hand posture and force signals [26]

This database of sEMG hand posture and force signals was presented and used in [26]. The signals are collected from 10 intact subjects (2 women, 8 men) using 7 sEMG electrodes (Aurion
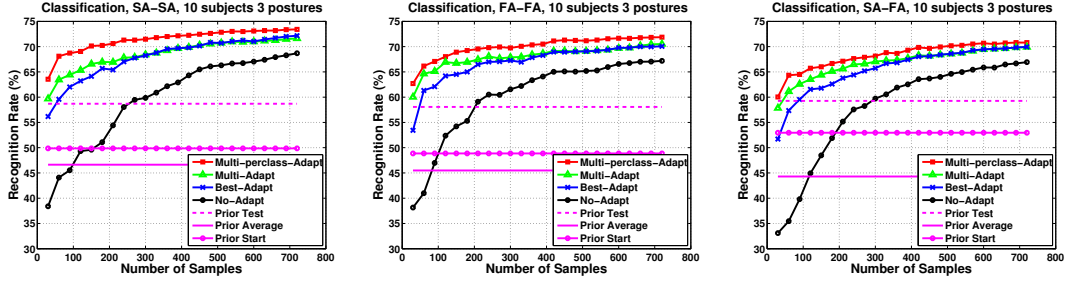
Figure 4.3 – Hand posture and force signals dataset [26]. Classification rate obtained averaging over all the subjects as a function of the number of samples in the training set. The title of each figure specifies if the data used as source and target are registered in Still-Arm (SA) or Free-Arm (FA) setting.
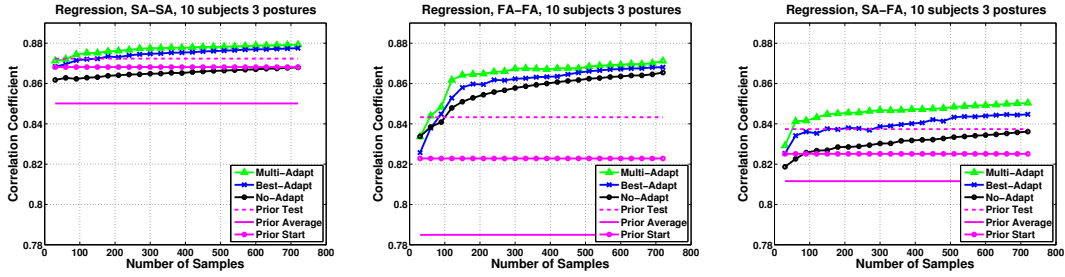
Figure 4.4 – Hand posture and force signals dataset [26]. Correlation coefficient obtained averaging over all the subjects as a function of the number of samples in the training set. The title of each figure specifies if the data used as source and target are registered in Still-Arm (SA) or Free-Arm (FA) setting.

ZeroWire wireless) placed on the dominant forearm according to the medical literature [79]. A FUTEK LMD500 force sensor is used to measure the force applied by the subject's hand during the recording. Data are originally sampled at 2 kHz. Each subject starts from a rest condition (sEMG baseline activity) then repeatedly grasps the force sensor using in turn three different grips, visible in Figure 4.2. The subject either remains seated and relaxed while performing the grasps, or is free to move (walk around, sit down stand up, etc.). These phases are referred to as *Still-Arm (SA)* and *Free-Arm (FA)* respectively. Each grasping action is repeated along 100 seconds of activity. The whole procedure is repeated twice. The root mean square of the signals along 1 second (for classification) and 0.2 seconds (for regression) is evaluated; subsampling at 25 Hz follows. Samples for which the applied force is lower than 20% of the average force value obtained for each subject are labeled as "rest" class. After this pre-processing we got around 15000 samples per subject, each sample consists of a 7 elements sEMG signal vector and one force value.

For our experiments on this dataset, the training sequences were random subsets from the entire dataset of the new subject, i.e. they are taken without considering the order in which they were acquired. We considered 24 successive learning steps, for each of them the number of
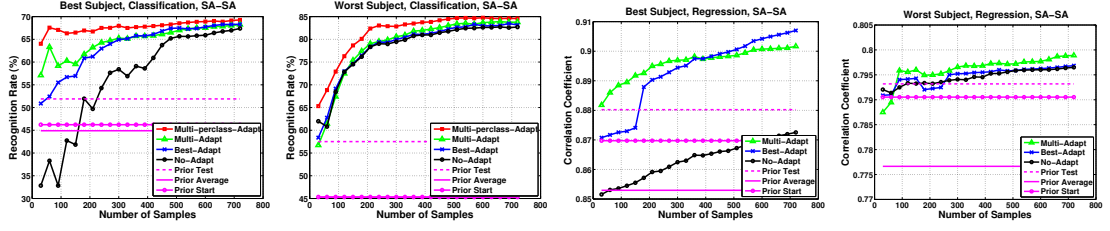
Figure 4.5 – Hand posture and force signals dataset [26]. Classification and Regression in the SA-SA setting for the best and worst subjects. With best and worst we mean the subjects for which the difference in performance between learning with adaption and learning from scratch is respectively maximum and minimum.
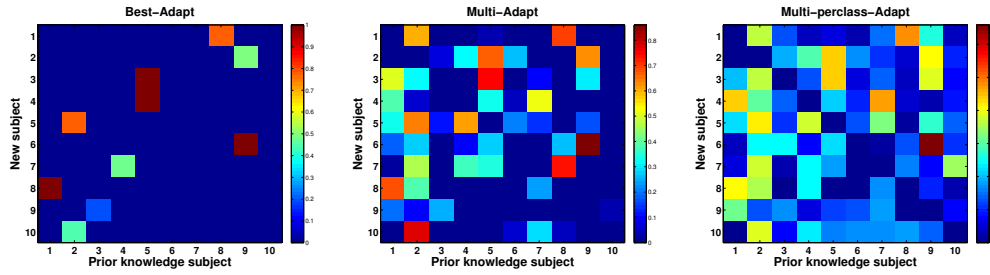
Figure 4.6 – Hand posture and force signals dataset [26]. Maps of the beta values for the three adaptive methods in classification, SA-SA, obtained for 210 training samples. The title of each figure indicates the adaptive method that produced the corresponding beta weights, in particular for Multi-perclass-Adapt we are showing the average values over the four classes (3 grasp postures plus rest). The rows 1 and 9 in all the matrices correspond respectively to the best and worst subject in classification considered in Figure 4.5, first and second plots from the left.

available training samples increases by 30 elements reaching a maximum of 720 samples. The test runs over all the remaining samples. We conducted three sets of experiments considering different prior knowledge-new problem pairs: SA-SA, FA-FA and SA-FA. In the first two cases we have consistent recording conditions among the source and the new target problem. The last case reproduces the more realistic scenario where the prior knowledge is built on data recorded on subjects in laboratory controlled conditions while the new subject moves freely. We both classify the grasp type and predict the force measured by the force sensor.

Figure 4.3 (left) reports the obtained classification rate at each step when using SA-SA data. The plot shows that Multi-perclass-Adapt outperforms both the baselines No-Adapt, Priors, and all the other adaptive learning methods. The difference between Multi-perclass-Adapt and Best-Adapt shows an average advantage in recognition rate of around 2% ($p < 0.03$). The gain obtained by Multi-perclass-Adapt with respect to No-Adapt ($p < 0.003$) stabilizes around 5% for 500-720 training samples.

Analogous results are obtained when considering FA-FA data: Figure 4.3 (center) reports the classification rate results in this setting. Multi-perclass-Adapt shows again the best perfor-

mance, but now the advantage with respect to Best-Adapt is significant ($p < 0.03$) only for less than 100 training samples. Multi-perclass-Adapt outperforms No-Adapt ($p < 0.03$) with a gain of 4% in recognition rate for 500-720 samples.

Finally, Figure 4.3 (right) shows the SA-FA results. Here the statistical comparison among Multi-perclass-Adapt, Best-Adapt and No-Adapt is the same as in the FA-FA case.

Analyzing Figure 4.3 as a whole, we can state that all the proposed adaptive methods outperform learning from scratch with the best results obtained when exploiting a linear combination of pre-trained models with a different weight for each known subject and each class (Multi-perclass-Adapt). Moreover, we notice that learning with adaption with 30 training samples performs almost as No-Adapt with around 300 samples. Considering the acquisition time, this means that the adaptive methods are almost ten time faster than learning from scratch. Using the prior knowledge by itself appears as a good choice if only very few training samples are available but looses its advantage when the dimension of the training set increases. Passing from SA-SA and FA-FA to SA-FA we can also notice that the results for Prior Average show a small drop (46.3%, 45.5%, 44.3%) related to the change in domain between the data used for pre-trained model and the one used for the new subject. The increasing difficulty of the task can be also evaluated by the progressive decrease in performance of Multi-perclass-Adapt at the very first step in the three cases: SA-SA 63.6%, FA-FA 62.7%, SA-FA 60.0%.

The corresponding regression results are reported in Figure 4.4. From the plot on the left we can notice that, in the SA-SA case, both the adaptive learning methods outperform No-Adapt ($p < 0.03$). However here Multi-Adapt and Best-Adapt performs almost equally (no statistical significant difference).

Figure 4.4 (center) shows that Best-Adapt is slightly worse than Multi-Adapt when passing to the FA-FA setting. Still the two methods are statistically equivalent and they show a significant gain with respect to No-Adapt only for more than 200 training samples ($p < 0.03$).

The problem becomes even harder in the SA-FA case (Figure 4.4 right), here Multi-Adapt outperforms No-Adapt only for more than 500 training samples ($p < 0.03$).

Globally the increasing difficulty of the three regression task passing from left to right in Figure 4.4 is demonstrated by the general drop in performance. Although we decided to show the correlation coefficient results, the corresponding mean square error would lead to the same conclusions.

Ninapro [7]

This database was presented in [7] and already used in [88]. It contains kinematic and sEMG data from the upper limbs of 27 intact subjects (7 women, 20 men) while performing 12 finger, 9 wrist, 23 grasping and functional movements, plus 8 isometric, isotonic hand configurations. Data are collected using 10 surface sEMG electrodes (double-differential OttoBock MyoBock
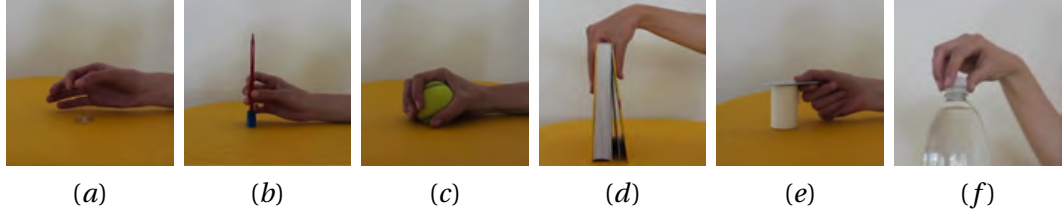
Figure 4.7 – The six different grasp types extracted from the Ninapro dataset [7]: (*a*) tip pinch grasp; (*b*) prismatic four fingers grasp; (*c*) power grasp; (*d*) parallel extension grasp; (*e*) lateral grasp; (*f*) open a bottle with a tripod grasp. Reproduced from [7].
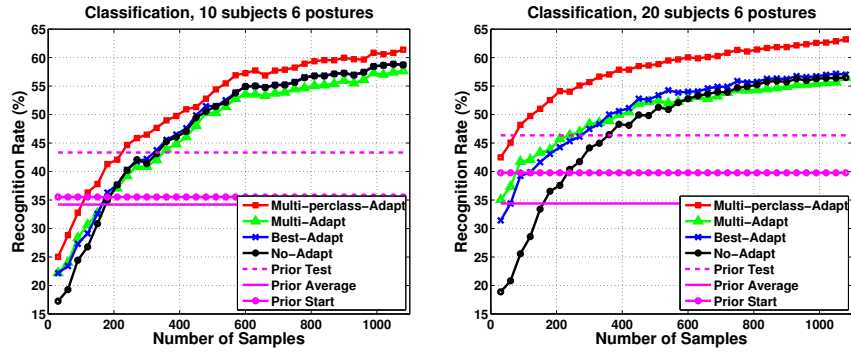
Figure 4.8 – Ninapro dataset [7]. Classification rate obtained averaging over all the subjects as a function of the number of samples in the training set. The title of each figure indicates the the number of subjects and hand postures considered.

13E200), 8 placed just beneath the elbow at fixed distance from the radio-humeral joint, while 2 are on the flexor and extensor muscles. Each subject sits comfortably on an adjustable chair in front of a table and is instructed to perform ten repetitions of each movement by imitating a video, alternated with a rest phase. The sEMG electrodes are connected to a standard DAQ card sampling the signals at 100 Hz and provide an RMS rectified version of the raw sEMG signal. We focused only on the grasp and functional movements extracting 6 actions: tip pinch, prismatic four fingers, power, parallel extension, lateral and open a bottle with a tripod grasp (see Figure 4.7). Each of them belongs to a different branch of a hierarchy containing all the dataset hand postures and the first three grasps are the most similar to the ones considered in [26]. We randomly extracted two sets of 10 and 20 subjects from the dataset and performed classification experiments on the described 7 class (6 grasps plus rest) problem considering the Mean Absolute Value (MAV) of the sEMG signal as time domain features [88]. We repeated the preprocessing and data split procedure described in [88] with an extra subsampling of the "rest" data to get a class-balanced setting.

For the experiments on this dataset, we shuffled randomly the training set and we considered 36 learning steps each with 30 samples reaching a maximum of 1080 training data.

Figure 4.8 (left) reports the obtained classification rate at each step when considering 10 subjects for the 6 grasp postures plus rest. The plot shows that all the adaptive methods performs almost as No-Adapt, in particular for less than 200 samples there is no statistical difference among learning from scratch, learning with adaptation or using directly the prior knowledge (the fair comparison is with Prior Average and Prior Start). It is important to remark that the "few sample" range grows together with the number of considered classes: the samples are selected randomly and it is necessary a minimum amount of samples per class to get meaningful classification results. Only Multi-perclass-Adapt outperforms No-Adapt ($p < 0.05$) with an average advantage of 2.5% in recognition rate for more than 200 samples.

Figure 4.8 (right) shows the corresponding results in case of 20 subjects. On average No-Adapt and Prior Average perform almost equally to the previous case (with 10 subjects), showing that the average learning capability per subject is almost stable in a fixed range. On the other hand Prior Test and Prior Start present an increase in performance: the higher is the number of available prior models, the higher is the probability to find useful information for the new problem. Moreover, here Multi-perclass-Adapt outperforms both Best-Adapt and No-Adapt ($p < 0.001$) with an average gain of 6% with respect to learning from scratch.

4.3.2 Conclusion

On the basis of the presented results we can state that the three proposed adaptive methods (Multi-perclass-adapt, Multi-Adapt and Best-Adapt) are able to properly leverage over source knowledge across different subjects.

In the considered setting the prior knowledge models by themselves are only partially helpful on a new target subject. Prior Test performance shows that, even supposing to know which is the best source, by using it directly we get an advantage that becomes negligible in case of many available training samples. On the other hand, the Prior Average line corresponds to an attempt to use a flat combination of all the pre-trained models on a new subject: the obtained results indicates that this is not a good solution.

In comparison with all the defined baselines, combining source and target knowledge with our domain adaptation approaches improve the learning performance to different extents if prior knowledge contains useful information and never harm if any good match between source and target is found.

Figure 4.5 shows the classification and regression results on SA-SA data respectively for the subject that have the maximum (best) and the minimum (worst) difference in recognition and regression performance with adaptation compared to No-Adapt. The worse-case subject represents the paradigmatic case of no previous models matching the current distribution; as a consequence the parameter β ($\boldsymbol{\beta}$) is set automatically to a small value (to a vector of small norm). In this case there is essentially no transfer of prior knowledge.

More insight on this point is given by Figure 4.6. Here we are mapping the beta values for each

adaptive model. Best-Adapt chooses only one prior model as reference, while Multi-Adapt can rely on more than one known subject. The results are consistent to each other: e.g. for subject 1 (1st row in the matrices), all the adaptive methods choose subject 8 as the most relevant, Multi-Adapt gives credit also to subject 2 and the same happens for Multi-perclass-Adapt which has more freedom in weighting each class and finds also subject 9 a bit useful.

Finally, if confirmed on data acquired from amputees, the current results could pave the way to a significantly higher acceptance of myoprostheses in the clinical setting.

4.4 Application to Visual Categories

In object recognition problems, many factors such as pose, illumination or image quality can produce a significant mismatch between two domains: images of the same object category may appear dramatically different and any standard classification model would degrade significantly when passing from one set to the other. Considering the vast amount of visual information available online nowadays, not being able to use it in situated environments (e.g. object recognition for images captured with a mobile phone camera) due to a possible domain shift constitutes a frustratingly strict limit.

To overcome this issue, some recent work focused on adapting visual category models to new domains. In [143, 87] the authors propose to learn a regularized non-linear transformation that compensate for the domain-induced changes. This method relies on the possibility to define similarity and dissimilarity constraints among the samples of two domains. Instead of assuming the availability of explicit information on the domain shift, [65] introduces a data-driven unsupervised approach. Here the source and the target domains are considered as points on a Grassmann manifold and intermediate subspace representations are obtained by sampling along the geodesic connecting them. Such subspaces are then combined and used to learn a discriminative classifier to predict on the target. This approach has been translated into an efficient kernel-based method in [64].

In the following we evaluate the performance of our Multi-Adapt and Multi-perclass-Adapt on visual object categorization, using the described state-of-the art methods as baselines.

4.4.1 Experiments

We considered the dataset presented in [143] which contains 31 different object categories collected under three domain settings: *amazon*, *dslr* and *webcam* (see Figure 4.9). The first consists of images from the web downloaded from the online merchant Amazon. The second corresponds to images captured with a digital SLR camera in realistic environments with natural lighting conditions. Finally the third consists of low resolution images recorded with a simple webcam. The amazon set has an average of 90 instances for each category, whereas dslr and webcam have roughly around 30 instances per category.
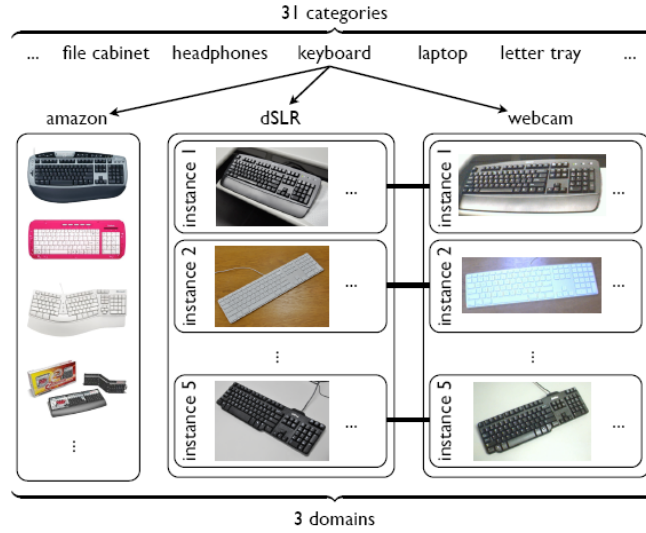
Figure 4.9 – Image dataset for investigating domain shift in visual category recognition tasks (reproduced from [143]).

We followed the experimental protocol proposed in previous work [143, 65, 64]. Briefly, we used SURF features encoded in a bag of words histogram, with an 800 elements codebook generated by k-means clustering on a random subset of amazon [1]. In the target, 3 samples per class are used as training and all the remaining images define the test set. In the source domain, 8 samples per class are used for webcam and dslr, while 20 for amazon; the multiclass models are learned with LS-SVM. For each set considered as target, we repeated the experiments on 10 random trials. For all the experiments we used the Gaussian kernel and we fixed $\gamma = 1$ as in [143], while $C = 100$ was chosen on the source sets and kept also when learning on the target. We benchmark our domain adaptation approaches with

Metric: the metric learning method proposed in [143].
SGF: the method based on subsampling the geodesic flow between two domains proposed in [65].
GFK: the geodesic flow kernel method proposed in [64].
No-Adapt: it is plain LS-SVM using only the new data for training, as it would be in the standard scenario without adaptation.

The authors of [64] re-implemented the methods Metric and SGF and reported better performance for these two approaches with respect to that presented in the original papers. Where possible we use the results in [64] for comparison.

The upper part of Table 4.1 presents the results in case of a single source set. By comparing them we can state that both Multi-Adapt and Multi-perclass-adapt outperform Metric and

---

1. We downloaded these features directly from http://www1.icsi.berkeley.edu/~saenko/projects.html

s	t	Metric [64]	SGF [64]	GFK [64]	No Adapt	Multi-Adapt	Multi-perclass-Adapt
W	D	48.1 ± 0.6	61.0 ± 0.5	66.3 ± 0.4	50.5 ± 2.6	59.8 ± 3.5	57.6 ± 4.1
D	W	36.9 ± 0.8	55.2 ± 0.6	61.3 ± 0.4	49.7 ± 1.4	58.7 ± 1.7	54.9 ± 1.4
A	W	34.5 ± 0.7	37.4 ± 0.5	46.4 ± 0.5	49.8 ± 2.5	52.9 ± 1.6	50.9 ± 2.5

s	t	SGF [65]	No Adapt	Multi-Adapt	Multi-perclass-Adapt
A, D	W	52 ± 2.5	48.1 ± 1.7	59.1 ± 1.4	54.1 ± 1.6
A, W	D	39 ± 1.1	49.6 ± 3.2	57.3 ± 2.9	55.5 ± 2.6
D, W	A	28 ± 0.8	21.1 ± 1.1	22.8 ± 1.2	22.2 ± 1.3

Table 4.1 – Recognition rate results (%) on the target domain with semi-supervised adaptation. In the first row of the table, s and t indicate respectively source and target, while in the first two columns W, D and A are used to indicate the domains webcam, dslr and amazon. All the baseline results are obtained with 1-nearest-neighbor.

are better or equal than SGF. With respect to GFK, Multi-Adapt produces analogous or better recognition rate results when the target is webcam, while the results are worse for dslr as target. Multi-perclass-adapt is instead not as good as GFK in two cases out of three. Considered the known relation in terms of similarity across the three domains, we can state that both our domain adaptation approaches appear more able than GFK to manage the case of weakly related domains.

The bottom part of Table 4.1 considers instead the case of two source domains. This setting was used previously only in [65] where the information of the two sources was averaged before proceeding with domain adaptation. Both Multi-Adapt and Multi-perclass-Adapt outperform SGF in two cases out of three. When amazon is used as target, our domain adaptation methods do not find any useful information in prior knowledge and perform as No Adapt.

By comparing Multi-Adapt and Multi-perclass-Adapt, we can see that the first is always slightly better than the second, although the difference is not statistically significant considering the standard deviation. This may be due to the different regularization imposed on the weights assigned to the class models for the two methods. Multi-perclass-Adapt has separate regularizing conditions for each class and this can be not the right solution in case of few (one or two) available sources.

4.4.2 Conclusion

The presented results show that our domain adaptation approaches are comparable with the state of the art methods on visual object categorization problems. The existing techniques are mostly based on deriving a new feature representation and usually consider the availability of a single source set. Starting from KT, we perform instead domain adaptation by transferring the model parameters. This on one side gives us the advantage of not requiring the storage of source samples and on the other allows for a proper weighting of multiple source sets.

4.5   Discussion

In this chapter we introduced several ways to extend out KT approach to domain adaptation problems showing that the obtained algorithms are able to perform well on two realistic tasks. This, apart from providing a further demonstration of the flexibility of KT, gives a new point of view on domain adaptation that is generally tackled by defining ad hoc feature representations.

Pointing out the possible limitations, the presented adaptive approaches have the same general constraints of the original KT. This means that both the source and the target problem should share the same learning space in terms of feature descriptors (unless starting form multiK-KT). Moreover, our domain adaptation methods need a minimum amount of labeled target samples, which means that they are feasible in semi-supervised settings but not in unsupervised ones where, on the other hand, feature learning techniques can perform.

The next chapter introduces a way to translates our model transfer approach into a feature transfer method.

Part II
Leveraging over Features

5 Transfer Learning From Unconstrained Sources

In this chapter we present how to move from our original KT model transfer method to a feature augmentation approach. By changing the form of the prior knowledge we free the target learning process from the constraint of using the same learning system chosen for the sources. Moreover, we can define a principled multiclass knowledge transfer method. We formulate the problem by casting it into the multi-kernel learning framework and we discuss the relation of the obtained approach to other existent state-of-the-art learning methods. Finally we assess the performance of the defined technique with binary and multiclass experiments both in the domain adaptation and knowledge transfer setting.

5.1 From Model Transfer to Feature Augmentation

When defining a transfer learning approach, the specific form of the knowledge to transfer may affect the choice of the learning algorithm. A model transfer method assumes that both the source and the target task are faced with the same learning approach, such that the source model can be used as reference for the target. However this condition may be too strict preventing the use of relevant information when it is coded in a way not directly accessible by a new learning problem.

We start here from the KT method presented in chapter 3 for binary tasks and we show that it is possible to reformulate it in a different and much more flexible way. In practice, instead of relying directly on the source models we can use their prediction as extra features for the target samples.

Let us start from

$$\min_{\boldsymbol{w},b} \frac{1}{2} \| \boldsymbol{w} - \beta \hat{\boldsymbol{w}} \|^2 + \frac{C}{2} \sum_{i=1}^{N} \xi_i^2$$

$$\text{subject to} \quad y_i = \boldsymbol{w} \cdot \phi(\boldsymbol{x}_i) + b + \xi_i \quad \text{for} \quad i = 1, \dots, N \,, \tag{5.1}$$

we already know that one of the optimality conditions on the corresponding Lagrangian is

$w = \beta \hat{w} + \mathbf{v}$, where $\mathbf{v} = \sum_{i=1}^{N} a_i \phi(\boldsymbol{x}_i)$. By substituting it back into (5.1) we get

$$\min_{\mathbf{v},b} \frac{1}{2} \|\mathbf{v}\|^2 + \frac{C}{2} \sum_{i=1}^{N} \xi_i^2$$

subject to $\quad y_i = (\beta \hat{w} + \mathbf{v}) \cdot \phi(\boldsymbol{x}_i) + b + \xi_i \qquad \text{for} \quad i = 1,\dots,N$. $\qquad\qquad$ (5.2)

We can now apply a change of variables, augmenting the dimensionality of the vectors in the following way

$$\bar{\mathbf{v}} = \left[\begin{array}{c} \mathbf{v} \\ \beta \end{array} \right] , \qquad\qquad (5.3)$$

$$\bar{\phi}(\boldsymbol{x}_i) = \left[\begin{array}{c} \phi(\boldsymbol{x}_i) \\ \hat{w} \cdot \phi(\boldsymbol{x}_i) \end{array} \right] . \qquad\qquad (5.4)$$

With this trick the learning problem can be rewritten as

$$\min_{\bar{\mathbf{v}},b} \frac{1}{2} \|\bar{\mathbf{v}}\|^2 + \frac{C}{2} \sum_{i=1}^{N} \xi_i^2$$

subject to $\quad y_i = \bar{\mathbf{v}} \cdot \bar{\phi}(\boldsymbol{x}_i) + b + \xi_i \qquad \text{for} \quad i = 1,\dots,N$, $\qquad\qquad$ (5.5)

which is exactly the original LS-SVM problem.

The described steps can be repeated independently from the number of considered sources. This demonstrates that our model transfer learning approach KT can be easily translated into a feature transfer method. In more general terms, from what shown above we can conclude that it is always possible to integrate some source knowledge by feature augmentation when learning on a new target task. We extend this idea supposing to neglect the specific form of the sources: regardless of the learning system used, we can always consider each source as a classifier and transfer its confidence output on a new sample as an extra feature descriptor. This frees the target learning process from the constraint of using the same source learning approach. At the same time, it allows the use of any loss function with the possibility to obtain a principled multiclass formulation.

5.2 Mathematical Framework

This section gives the formal definition of a transfer learning approach based on feature augmentation, starting from the intuition presented above. We introduce here the notation used in this chapter: matrices and vectors are again represented with small and capital bold letters. We use the bar accent to indicate the vector formed by the concatenation of K vectors, hence $\bar{\boldsymbol{w}} = [\boldsymbol{w}^1, \boldsymbol{w}^2, \cdots, \boldsymbol{w}^K]^T$.

5.2.1   Prior Knowledge and Transfer Setting

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

$$f(\boldsymbol{x}) = \underset{z \in \mathcal{Z}}{\mathrm{argmax}} \ s(\boldsymbol{x}, z)$$

where $s(\boldsymbol{x}, z)$ 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 source classifier is about assigning the label z to the instance $\boldsymbol{x}$. For binary classification, the function can be further simplified as $f(\boldsymbol{x}) = \mathrm{sign}\,(s(\boldsymbol{x}))$. In the following, we will describe the multiclass case, as its modification to the binary condition is straightforward.

We are interested in the task of learning a classifier for F' target categories, different from the F source categories already known. Given the new training set $\{\boldsymbol{x}_i, y_i\}_{i=1}^{N}$, we also gather for each sample the source score $s_{\mathrm{s}}(\boldsymbol{x}_i, z)$, $z = 1, \ldots, F$ predicted by the prior models, and we propose to use them as auxiliary features. To this purpose we introduce the joint feature mapping function $\phi^{(\cdot)}(\cdot, \cdot) : \mathcal{X} \times \mathcal{Y} \rightarrow \mathcal{H}$ [173], which maps the samples into some high, possibly infinite dimensional space. We define $\phi^{(0)}(\boldsymbol{x}, y)$ for the original input feature $\boldsymbol{x}$, and $\phi^{(y,z)}(s_{\mathrm{s}}(\boldsymbol{x}, z), y)$ as the mapping of the prior score of class z to the new class y where $y = 1, \ldots, F'$ and $z = 1, \ldots, F$.

We focus on the standard linear model, thus when learning on the target task, by concatenating all the mappings in $\bar{\phi}(\boldsymbol{x}, y)$ the score function results

$$s(\boldsymbol{x}, y) = \bar{\boldsymbol{w}} \cdot \bar{\phi}(\boldsymbol{x}, y) \tag{5.6}$$

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

here $\boldsymbol{w}^{(\cdot)}$ is a hyperplane, and $\bar{\boldsymbol{w}}$ contains all the parameters of the learning model. Finally, the predicted label for the target task is the class achieving the highest score:

$$f(\boldsymbol{x}) = \underset{y \in \mathcal{Y}}{\mathrm{argmax}} \ s(\boldsymbol{x}, y)\,,$$

where $\mathcal{Y} = \{1, \ldots, F'\}$.

To have an intuitive understanding of the learning problem, let's consider a multiclass (F classes) source containing among the others the model of *bicycle* and *dog*, together with a new multiclass (F' classes) target task where one of the classes is *motorbike*. On the basis of the visual similarities, we can suppose that the scores produced by the prior knowledge on a motorbike sample $\boldsymbol{x}$ are $s_{\mathrm{s}}(\boldsymbol{x}, \mathrm{bicycle}) > s_{\mathrm{s}}(\boldsymbol{x}, \mathrm{dog})$. This suggests that the information coming from the class bicycle is more relevant to recognize a motorbike with respect to that
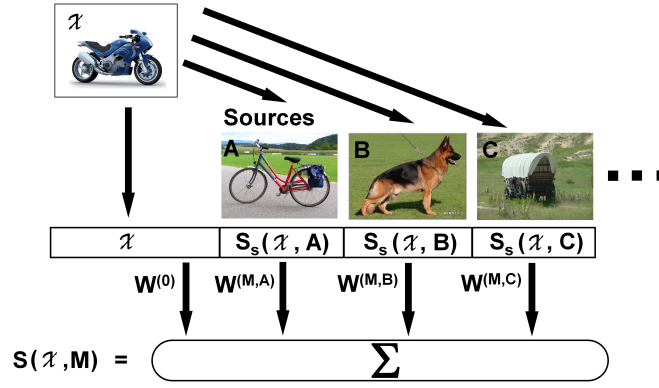
Figure 5.1 – A graphical representation of how to use the outputs from the prior models as auxiliary features when computing the score of a new class. Here M stands for motorbike, while {A,B, C, ...} indicates the source classes bicycle, dog, covered wagon, etc.

coming from the class dog. Thus in defining the motorbike target model we would expect to give more weight to the first than to the second (see Figure 5.1).

5.2.2   The Learning Problem

Solving the learning problem described above consists in finding the best $\bar{\boldsymbol{w}}$ able to generalize correctly on a new test sample. This can be defined by minimizing the structural risk on N training samples. We formalize the problem starting from the generic objective function already presented in chapter 3:

$$\min_{\bar{\boldsymbol{w}}} \ \Omega(\bar{\boldsymbol{w}}) + C \sum_{i=1}^{N} \ell\left(\bar{\boldsymbol{w}}, \boldsymbol{x}_i, y_i\right) , \tag{5.7}$$

where $\Omega(\bar{\boldsymbol{w}})$ is a regularizer which avoids overfitting, C is the coefficient that controls the bias-variance trade-off, and ℓ is some convex, non negative loss function.

In defining the specific form of the regularizer, we must remember that each component $\boldsymbol{w}^{(y,z)}$ of $\bar{\boldsymbol{w}}$ corresponds to the weight given to one source knowledge and evaluates from where and how much to transfer. We would like to impose sparsity on the prior models for two reasons: (1) 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; (2) among the F prior models, we expect only few to be relevant with respect to each specific new class, while the rest can even add noise producing negative transfer. Thus a good choice for the regularizer consists in the squared $(2, p)$ group norm [184]:

$$\begin{aligned} \Omega(\bar{\boldsymbol{w}}) &= \frac{1}{2} \|\bar{\boldsymbol{w}}\|_{2,p}^2 \\ &= \frac{1}{2} \left\| \left[\|\boldsymbol{w}^{(0)}\|_2, \|\boldsymbol{w}^{(1,1)}\|_2, \cdots, \|\boldsymbol{w}^{(F',F)}\|_2 \right] \right\|_p^2 , \end{aligned} \tag{5.8}$$

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 us to tune the level of sparsity on the norms, increasing it if p is close to 1.

As loss we can use any convex Lipschitz function. For the binary case, we consider the most popular hinge loss:

$$\ell^{\mathrm{H}}\left(\bar{\boldsymbol{w}}, \boldsymbol{x}, y\right) = \max\{0, 1 - y\bar{\boldsymbol{w}} \cdot \bar{\phi}(\boldsymbol{x})\}, \tag{5.9}$$

while for the multiclass case, we choose the convex multiclass loss [36, 173]:

$$\ell^{\mathrm{MC}}\left(\bar{\boldsymbol{w}}, \boldsymbol{x}, y\right) = \max_{y' \neq y}\{0, 1 - \bar{\boldsymbol{w}} \cdot (\bar{\phi}(\boldsymbol{x}, y) - \bar{\phi}(\boldsymbol{x}, y'))\}. \tag{5.10}$$

5.3 Multiple Kernel Learning

In the domain of Support Vector Machines, the problem of integrating multiple information in learning corresponds to using multiple kernels and choosing how to properly weight them. The Multiple Kernel Learning (MKL) algorithm, proposed for the first time in [9], is based on a principled strategy to jointly solve the learning problem and finding the optimal kernel combination.

The mathematical formulation for this algorithm assumes the existence of a combined model parameter vector $\bar{\boldsymbol{w}} = [\boldsymbol{w}^1, \boldsymbol{w}^2, \cdots, \boldsymbol{w}^K]^T$, where K is the number of kernels. The original MKL uses a L_1 norm regularization which has been recently extended to the L_p norm version [83, 120]: this permits tuning the level of sparsity over the kernels and leads to better performance when all the combined information are relevant. Hence, by using a generic group norm and a generic convex function, the MKL optimization problem can be written as:

$$\min_{\bar{\boldsymbol{w}}} \quad \frac{1}{2}\|\bar{\boldsymbol{w}}\|_{2,p}^2 + C\sum_{i=1}^{N} \ell\left(\bar{\boldsymbol{w}}, \boldsymbol{x}_i, y_i\right). \tag{5.11}$$

When $p = 1$, we recover the formulation proposed in [9] and the optimization problem is very difficult to solve due to the non-smooth nature of the L_1 norm. It has been shown that when p is larger than 1, the optimization problem (5.11) becomes much easier [120], at the same time when p tends to 1, the solution still gets extremely close to the sparse solution of $p = 1$.

5.4 Multiple Kernel Transfer Learning

The transfer learning problem proposed in section 5.2.2, can be easily cast in the multi-kernel learning framework. In particular, the L_p norm MKL algorithm can be used to solve it with any off-the-shelf implementation. We name the final method *Multiple Kernel Transfer Learning* (MKTL, [104]), and we give in the following all the details necessary to set it.

First of all the full definition of the $\bar{\boldsymbol{w}}$ and $\bar{\phi}(\boldsymbol{x}, y)$ is

$$\bar{\boldsymbol{w}} = [\, \boldsymbol{w}^{(0)}, \, \boldsymbol{w}^{(1,1)}, \, \cdots, \, \boldsymbol{w}^{(y,z)}, \, \cdots, \, \boldsymbol{w}^{(F',F)} \,]^T \,,$$

$$\bar{\phi}(\boldsymbol{x}, y) = [\, \phi^{(0)}(\boldsymbol{x}, y), \, \phi^{(1,1)}(s_{\mathrm{s}}(\boldsymbol{x}, 1), 1), \, \cdots, \, \phi^{(y,z)}(s_{\mathrm{s}}(\boldsymbol{x}, z), y), \, \cdots, \, \phi^{(F',F)}(s_{\mathrm{s}}(\boldsymbol{x}, F), F') \,]^T \,.$$

Therefore, in total, we have $(F \times F' + 1)$ feature mapping functions $\phi^{(\cdot)}(\cdot, \cdot)$, and the same number of kernels $K^k((\boldsymbol{x}, y), (\boldsymbol{x}', y')) = \phi^k(\boldsymbol{x}, y) \cdot \phi^k(\boldsymbol{x}', y')$.

For a multiclass target problem, we suppose to have F' different hyperplanes, one for each new class. Thus $\boldsymbol{w}^{(0)}$ and $\boldsymbol{w}^{(y,z)}$ are composed by F' blocks as well as $\phi^{(0)}$ and $\phi^{(y,z)}$. We define the joint mapping function for the original feature as

$$\phi^{(0)}(\boldsymbol{x}, y) = [\mathbf{0}, \cdots, \mathbf{0}, \underbrace{\psi^{(0)}(\boldsymbol{x})}_{y}, \mathbf{0}, \cdots, \mathbf{0}]^T \,, \tag{5.12}$$

where $\psi^{(0)}(\cdot)$ is a transformation that depends only on the data and occupies the $y-th$ position in the vector. The feature mapping function for the z-th prior model output is defined as:

$$\phi^{(y',z)}(\boldsymbol{x}, y) = \begin{cases} [\mathbf{0}, \cdots, \underbrace{\psi(s_{\mathrm{s}}(\boldsymbol{x}, z))}_{y}, \cdots, \mathbf{0}]^T \,, & \text{if } y = y' \\ \\ \mathbf{0} & , \quad \text{otherwise} \,, \end{cases}$$

where $y, y' \in \mathcal{Y} = \{1, \ldots, F'\}$. 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'.

5.4.1 MKL Solver and Efficient Implementations

We solve the MKTL problem using the online-batch strongly convex multi-kernel learning (OBSCURE) approach [120]. OBSCURE [1] 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 increases, which somehow mitigates the problem that the number of kernels grows linearly with the number of prior knowledge models. Moreover, this approach 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).

Here, we only consider a linear mapping function $\psi(s_{\mathrm{s}}(\boldsymbol{x}, z)) = s_{\mathrm{s}}(\boldsymbol{x}, z)$ (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 can also

---

1. We used the implementation available in the DOGMA toolbox [119].

store $\boldsymbol{w}^{(y,z)}$ directly in its primal representation. Hence, compared to the original supervised learning problem without prior knowledge, the algorithm uses $\mathcal{O}(F \times F')$ extra memory space, and the additional computational complexity at each iteration is also $\mathcal{O}(F \times F')$.

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 [120], it is also possible to set $p^* = \frac{2 \log K}{2 \log K - 1}$ to get a convergence rate that depends logarithmically on the total number of kernels, which is denoted by K. With this setup we have only one free parameter C.

5.5 Comparison with Existing methods

In this section we discuss the relation of our MKTL to other standard learning and knowledge transfer methods. We also consider the comparison with a domain adaptation technique.

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 became available on the web. Several papers demonstrated that the output of visual attributes [56, 89] and semantic visual concepts [171, 177] can be used to define a good feature representation and to improve recognition performance. In particular Object Bank [96] uses the output of semantic part detectors (e.g., sky, tree) as features. Since the information is extracted at different spatial pyramid levels and is associated to a localized representation, this solution is particularly suited to recognize cluttered images composed of many objects such as natural scenes.

Our transfer learning approach follows this general line of thought. 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. By using the MKL machinery we can group freely information from various sources, including the new training data, into different kernels. We consider object categorization problems where features extracted from the whole image are the standard, however the MKTL algorithm is general and can handle various representations, e.g., the same used by Object Bank.

Binary KT. We have already briefly illustrated the relation of our KT binary approach and the feature augmentation solution in section 5.1. Indeed, if we consider F binary sources, the KT score function for a given sample $\boldsymbol{x}$ can be written as:

$$s(\boldsymbol{x}) = \mathbf{v} \cdot \phi(\boldsymbol{x}) + \sum_{z=1}^{F} \beta_z \hat{\boldsymbol{w}}_z \cdot \phi(\boldsymbol{x}) \,.$$

This is similar to the binary version of the MKTL score function defined in (5.6). However, our original KT is solved on the basis of two separate optimization problems: the first step identifies the best weights β_z while the second gives the optimal $\mathbf{v}$. Instead MKTL finds

both the best hyperplane parameters and the weights to be assigned to each prior knowledge model in a single joint optimization process.

Moreover KT requires that the prior knowledge models are obtained with a kernel learning approach i.e. by using the same type of classifier and hyperparameters of the new model. MKTL does not have this constraint and it is capable of *heterogeneous transfer from unconstrained priors*: we can freely combine different learning methods and different features to boost performance.

Finally, KT can be used in multiclass settings on the basis of the 1-vs-All approach, as we have seen in the previous chapter, but it cannot be extended to principled multiclass formulation. On the other hand, this is possible for MKTL by using the loss function in (5.10).

Feature Replication. A very simple strategy for domain adaptation based on increasing the feature representation was introduced in [43]. The proposed idea consists in augmenting the feature vectors of both the target (t) and the source (s) samples by replication as follows:

$$\phi^s(\boldsymbol{x}) = [\phi(\boldsymbol{x}), \phi(\boldsymbol{x}), \mathbf{0}]^T \tag{5.13}$$
$$\phi^t(\boldsymbol{x}) = [\phi(\boldsymbol{x}), \mathbf{0}, \phi(\boldsymbol{x})]^T .$$

In practice each feature is repeated three times: the first block is related to a general representation and it is common for source and target, the second is instead specific for the source and finally the third is specific for the target. If we indicate with K the kernel associated to the feature representation $\phi(\cdot)$, we know that $K(\boldsymbol{x}, \boldsymbol{x}') = \phi(\boldsymbol{x}) \cdot \phi(\boldsymbol{x}')$. In the expanded space defined through (5.13) the kernel function is

$$\tilde{K}(\boldsymbol{x}, \boldsymbol{x}') = \begin{cases} 2K(\boldsymbol{x}, \boldsymbol{x}'), & \text{if } \boldsymbol{x}, \boldsymbol{x}' \in s \text{ or } \boldsymbol{x}, \boldsymbol{x}' \in t \\ K(\boldsymbol{x}, \boldsymbol{x}'), & \text{otherwise} . \end{cases} \tag{5.14}$$

Intuitively, if we consider the kernel as a measure of data similarity, the chosen representation is such that data points from the same domain are by default twice as similar as those from different domains [43].

This strategy, although in some way related to our MKTL, is different in several aspects. First of all the described feature replication method supposes that when training on the new target task all the source samples are available: this corresponds to an instance transfer method. MKTL instead does not need to store the source samples: the pre-existing prior models are directly used to predict on the target and the obtained output is considered as source information. Moreover, MKTL automatically assigns different weights to each source knowledge. This leads to a proper combination of kernel functions, differently from the described case in (5.14) which considers fixed weights equals for source and target.

5.6 Experiments

We present here several experiments designed to study the behavior of MKTL in different settings, each specified in one of the following subsections. We benchmark MKTL against the following baselines:

no transfer or *No-Adapt:* they correspond to standard supervised learning without considering prior knowledge. The two names are used respectively in transfer learning and domain adaptation experiments and correspond to train an SVM classifiers using the 1-vs-All scheme or LS-SVM as in section 4.3.1.

prior-features: The output of all the prior models are used as feature descriptors, we concatenated them into a vector representation and applied a linear SVM classifier. This baseline follows the "classemes" idea [171] and it is useful to understand the role of the prior models in the final performance of MKTL. For example if the obtained recognition rate on the test set is low in comparison to no transfer, we might expect to see a very small improvement of MKTL over the performance of 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 knowledge alone could lead already to high accuracy.

KT or *Multi-perclass-Adapt:* We also compared against our KT transfer learning algorithm or the corresponding multiclass domain adaptation version presented in section 4.2.1. These methods assume that both the prior models and the new model use the same feature descriptor and the same type of learning method (SVM classifier).

average-TL: MKTL learns the weights to combine the outputs of each prior model with the new knowledge representation. Thus, a natural baseline is to consider the information coming from the sources and the target as equally relevant. This can be done by training an SVM classifier using the average over all the available kernels. This method often performs as good as many MKL algorithms [61].

feature-replication: This corresponds to the domain adaptation method presented [43].

For all of the transfer learning experiments the regularization parameter C of the considered SVM-based methods is selected in $\{0.1, 1, 10, 100, 1000\}$, and the parameter p for MKTL is chosen in $\{1.01, 1.05, 1.10, 1.15, 1.20, 1.25, 1.30, 1.40, 1.50\}$. We report the results obtained in two settings. In one case we let no transfer choosing the best C on the target problem and we fix that for all the other methods. In the other case we follow the strategy proposed in [50] and let all the approaches free to select their parameter in the defined sets, showing then the best result for each method. For p we considered both the best and the automatic value p^* as described in section 5.4. For no transfer, average-TL (in relation to the mapping $\phi^{(0)}$) and KT, we use the Gaussian kernel: γ is equal to the mean value of the pairwise distances among the samples for the first two methods and is defined by cross validation on the source knowledge

for KT.

For the domain adaptation experiments we reproduced the setup presented in section 4.3.1.

5.6.1 Binary Transfer Learning

In binary problems 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). For these experiments we reproduced the setup defined in chapter 3.

We selected 30 classes from the Caltech-256 dataset[2] and we considered the same four image descriptors already used in section 3.4.2, combining here the features through concatenation. In turns, one object class is used as target and all the 29 remaining classes are used as sources. For each source we have two models both learned with LS-SVM and the Gaussian kernel: one is based on a fixed pair (γ, C) optimized over all the sources, while the other considers specific pair of parameters different from source to source and evaluated as the best for each. The first is used for KT while the second for prior-features and MKTL.

Each class contains 80 positive and 80 negative samples. When used as source, all the samples of a class are considered together, while 50 positive and 50 negative images are randomly selected to define the target test set. The target training set is defined by the remaining 30 background samples with an increasing number of positive object images from 1 to 30. This means that in all the training steps except the last one we have an unbalanced problem and to tackle it we modified MKTL giving different importance weights to the positive and negative training examples, with a strategy analogous to that adopted for KT (see section 3.3.5). Here the weights are defined as $\zeta^+ = N^-/N^+$ and $\zeta^- = 1$, where N^+ and N^- are the number of positive and negative samples. Both the plain version of *MKTL* ($\zeta^+ = \zeta^- = 1$) and the weighted one, indicated as *MKTLw*, are considered in the experiments.

The average results over all the 30 categories and 10 trials, as well as the average results for each class, are shown in Figure 5.2. It can be observed that all the transfer learning methods outperform the no transfer approach ($p \leq 0.01$, using the sign test). The weighted version of MKTL achieves in general equal or better performance than KT ($p \leq 0.02$), while the plain version is worse only at the beginning for less than 10 positive training samples. Since both the source and the target problem are binary and consist of distinguishing different objects from a common background class, we expect the prior-features to achieve high accuracy. This is actually visible in the experimental results where prior-features is always better than no transfer ($p \leq 0.02$). MKTL combines the source knowledge in prior-features with the

---

2. In particular we considered three classes from ten macro-categories following the ontology of the dataset. "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.
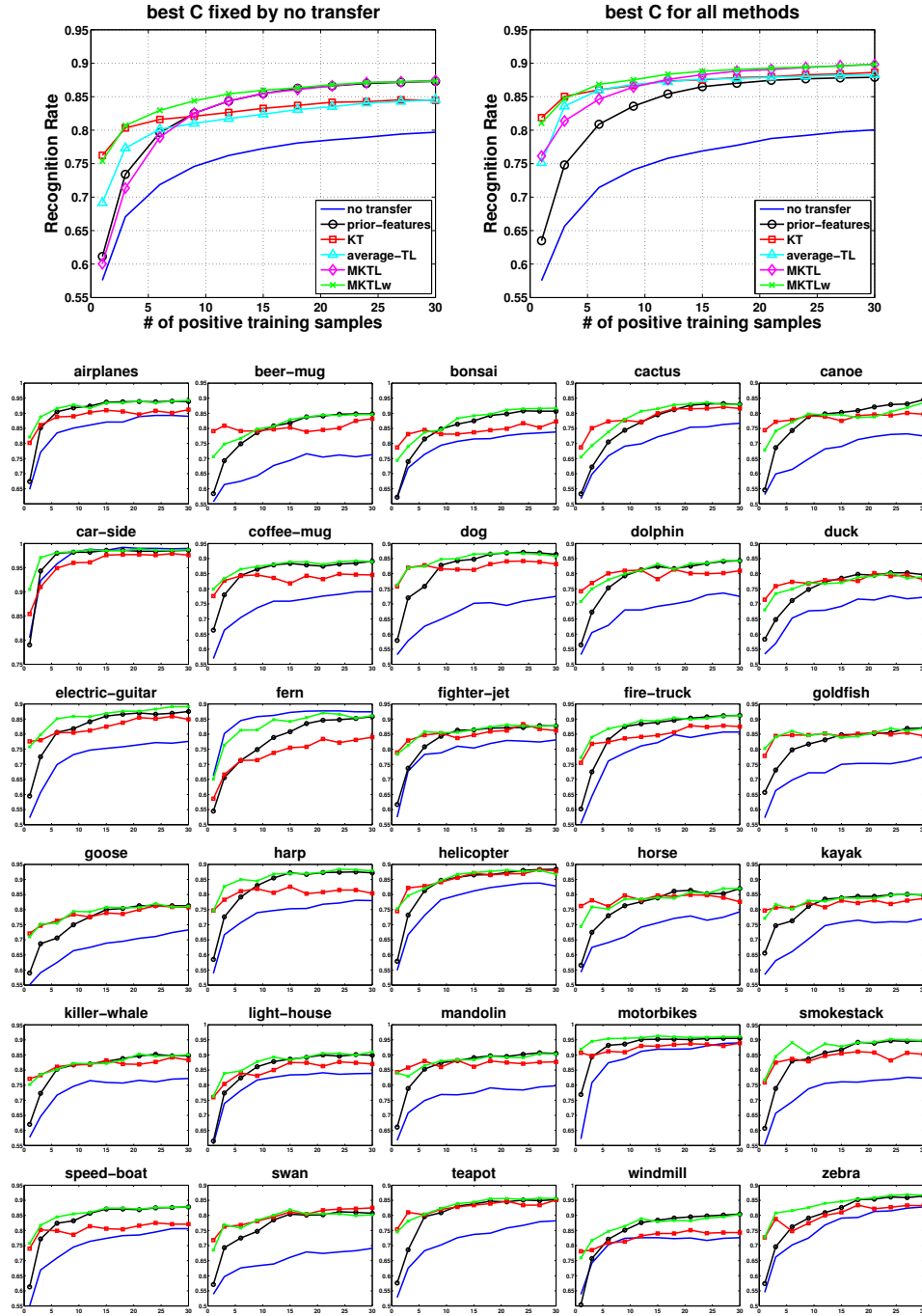
Figure 5.2 – Results obtained for binary transfer learning, when one out of 30 object categories is used as target while all the remaining 29 classes are sources. The classification performance is shown as a function of the number of object training images. For each class, we repeat the experiment 10 times using different random permutations. Two different setting for the parameter C are considered and indicated in the figure title, while always the best p is used. From the left plot we also extracted the results class by class. For the sake of clarity, we only report the results of no transfer, prior-feature, KT and MKTLw on these last figures.
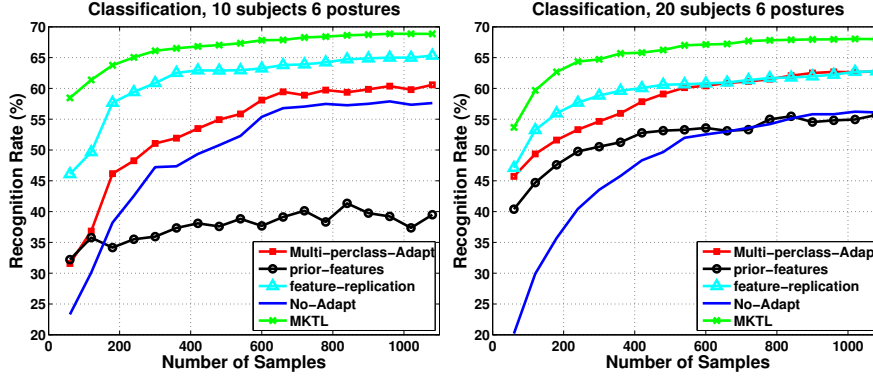
Figure 5.3 – Results obtained in the domain adaptation scenario on the Ninapro dataset [7]. Classification rate obtained averaging over all the subjects as a function of the number of samples in the training set. The title of each figure indicates the the number of subjects and hand postures considered.

new target information and it guarantees a performance as least as good as what has been transferred. Finally average-TL performs almost as KT but it is worst for less than five positive samples ($p \leq 0.02$). Compared to MKTL, the results of average-TL indicates that it is important to properly weight the prior knowledge.

It is also interesting to look into the results obtained for each single class. In most cases transferring gives a big advantage and this is particularly visible for the animals e.g. *dog* and *swan*. For *car-side* it seems instead that the prior knowledge is not really relevant and all the methods perform almost as no transfer. Finally *fern* is the only case in which prior features and KT clearly fail to avoid negative transfer, while MKTL performs almost as no-transfer. One possible explanation of this behavior may be that the clutter class of Caltech-256 used as background contains many images of grass, lawn, leaves and trees which are easily confused with fern.

5.6.2 Domain Adaptation

Our MKTL algorithm can be used directly on domain adaptation problems without any particular modification. To evaluate its behavior in this framework we re-ran the experiments on the Ninapro data (see section 4.3.1). The prior knowledge models used as reference for Multi-perclass-Adapt are now exploited as expert that predict on each target sample. The outputs are then used to augment the original feature representation. We consider an increasing target training set with 18 steps of 60 samples each, and for MKTL we chose the same (γ, C) parameters already used in the previous experiments, maintaining in this way a fair comparison with No-Adapt and Multi-perclass-Adapt. We also fixed the sparsity parameter according to the optimal automatic value p^* (see section 5.4.1).

Figure 5.3 reports the results both for the case of 10 and 20 subjects performing the 6 hand

movements plus the rest condition. The gain obtained by MKTL over the baselines is evident: the sign test confirms that MKTL significantly outperform both Multi-perclass-Adapt ($p \leq 0.01$) and feature-replication ($p \leq 0.05$). When 9 subjects are used as sources prior-features appear to be not really informative. This changes when the number of sources increase to 19. Still the weighted combination of prior-features and new knowledge show very high recognition rate results, allowing to state the effectiveness of MKTL for domain adaptation problems.

5.6.3 Multiclass Transfer Learning

We consider here the case of multiclass transfer learning where the source and the target task have different label sets. With respect to the domain adaptation setting where the correspondence among the tasks could possibly guide a 1-vs-All approach (as for Multi-perclass-Adapt), now there is no pre-defined rule in transferring. This is not a problem for a feature transfer method as MKTL, but it is not clear how to perform model transfer for KT. In the experiments we suppose to collect all the 1-vs-All models learned for each source set and provide them as prior knowledge to KT that can consider a linear combination of all of them for each new 1-vs-All model on the target. We run the experiments on two different datasets.

Subset of Caltech-256.   We selected nine classes as target: bonsai, sunflower, mushroom, horse, skunk, gorilla, motorbike, snowmobile, segway. For each class we extracted a maximum of 30 training samples and 50 testing samples. Twenty-three classes were chosen to form four multiclass source sets with each class containing 80 samples: *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).

We used different feature descriptors for each source: a combination of SIFT and LBP for both plants and mix, SIFT for vehicles and the combination of REGCOV [174], SIFT and V1S+ [131] for animals [3]. For plants and vehicles, we concatenated the feature descriptors together, and we used Multiclass AdaBoost [147] as learning method. For animals and mix, we computed the Gaussian kernel for each feature descriptor, and trained SVM using the average kernel with the 1-vs-All multiclass extension. On target samples we used the PHOG features together with the Gaussian kernel.

For the experiments on this setting our choice of feature descriptors and source learning methods is arbitrary as we want to show that the prior knowledge could be defined using various cues and learning algorithms in an unconstrained way. We suppose to use the sources as black boxes that take as input the target samples and give as output a confidence in classification, used to augment their feature representation. Moreover, we also analyze here what happens when the available source information increases starting with *plants*, *animals* and vehicles and adding *mix* only in a second step.

---

3. We used the precalculated descriptors available from http://www.vision.ee.ethz.ch/~pgehler/projects/iccv09/
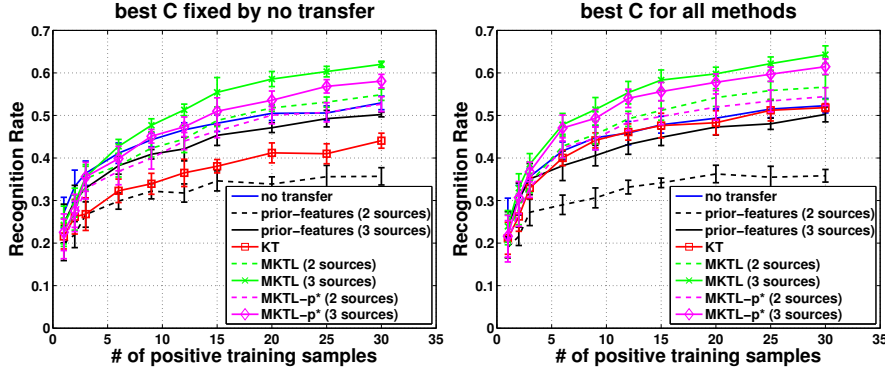
Figure 5.4 – Results obtained in the multiclass object categorization scenario on a subset of Caltech-256. Classification performance is shown as a function of the number of object training images. Each experiment is repeated for 10 times, and the average results are reported with the corresponding standard deviation. (2 sources) indicates that we used *plants* and *animals* as prior knowledge sets, while (3 sources) considers also *mix*.
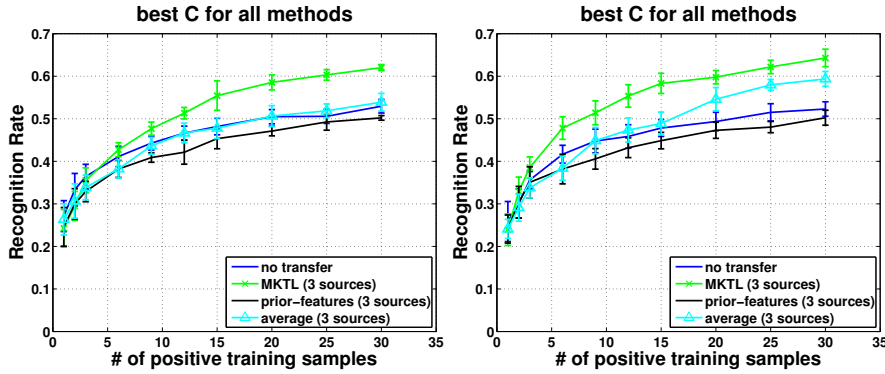
Figure 5.5 – Results obtained in the multiclass object categorization scenario on a subset of Caltech-256. Same settings of Figure 5.4, here we show the comparison with the average transfer solution.

The results are reported in Figure 5.4. They clearly show that MKTL has a gain in performance with respect to no transfer and the other baseline methods, especially when the number of training samples grows and the source knowledge increases. Here the expected *higher start* effect (see section 2.2) 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.

The results for MKTL using the p^* parameter is comparable to the results we obtained when choosing the best possible p. This suggests a way to eliminate one free parameter in
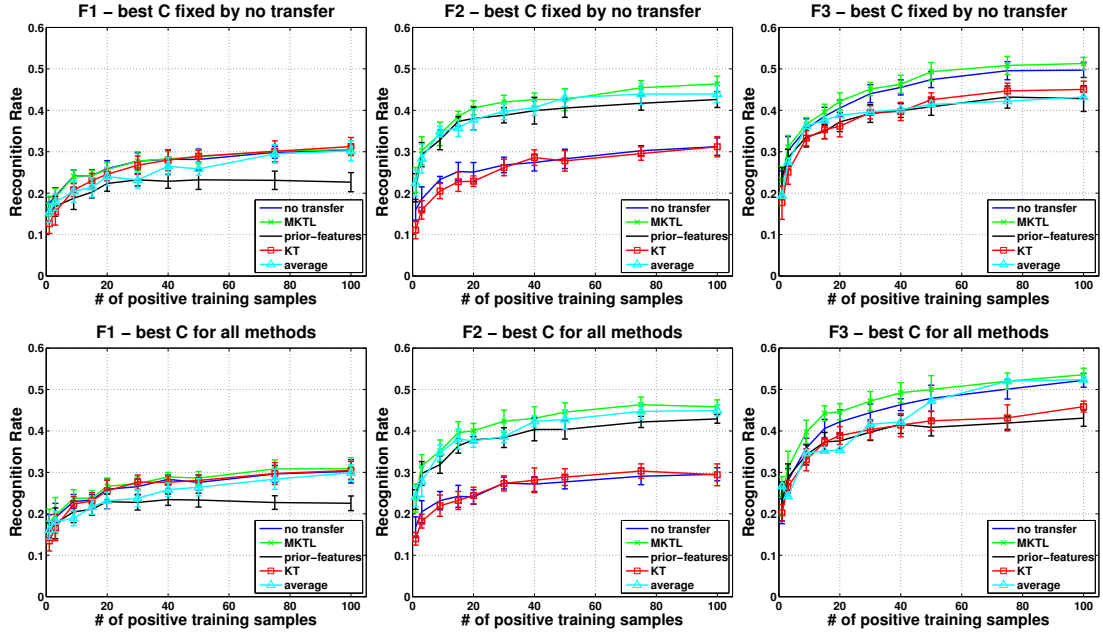
Figure 5.6 – Results obtained in the multiclass object categorization scenario on the Animals with Attributed dataset. Classification performance is shown as a function of the number of object training images. Each experiment is repeated for 10 times, and the average results are reported with the corresponding standard deviation. The title indicates which setting has been chosen for the features and the parameter C.

practice. Regarding KT, it is evident that it does not improve over the no transfer baseline and appears even worse when using the C best value chosen by no transfer (Figure 5.4 left). This behavior might be expected considering the possible confusion among the classes created when combining the source 1-vs-All results over the target.

By focusing on the case with all the three source sets (plants, vehicles and mix), Figure 5.5 shows the comparison of MKTL with respect to an *average* transfer approach. This is obtained simply combining with equal weight the linear kernel on the prior-features and the Gaussian kernel on the PHOG features extracted from the training samples (no transfer) and running 1-vs-All SVM. In both the two possible settings of the parameter C it is clear that MKTL outperforms the average solution.

Animals with Attributes. We performed similar experiments on the AwA dataset. We consider the same 10 test classes in [89] as new classes to learn [4], randomly extracting a maximum of 100 samples from each class for training and 50 samples for test. The remaining 40 classes define a multiclass knowledge source.

We run this experiments with a specific focus on the feature choice, while keeping all the learning methods fixed and equal (SVM for source and target). We assume that for the full dataset

---

4. chimpanzee, giant panda, hippopotamus, humpback whale,leopard, persian cat, pig, raccoon, rat,seal.

three feature descriptors are available: color histograms, SURF and PHOG [5]. We indicate with F1 the case in which both source and target use only PHOG. F2 is instead the case where the prior knowledge is built with a combination of color histogram and SURF, but on the target we consider only PHOG, again using the sources as black boxes to which the image samples are given in input. Finally we indicate with F3 the hypothesis of all the samples in source and target described by all the three features. Note that only for KT the source knowledge should live in the same learning space of the target, thus for this method we built the prior models by using the same descriptor chosen for the new target set.

The obtained results are reported in Figure 5.6. Regardless of the specific choice for the learning parameter C, when the sources and the target are based on the same descriptor, MKTL performs equally or only marginally better than no transfer. This is visible in the F1 and F3 cases. Here the prior-features do not result very useful, by learning on them we obtain a lower performance than learning from scratch. On the other hand, in case of heterogeneous features across source and target (F2) the advantage obtained by MKTL over no transfer is clear and it is significant even over prior-features. Here the improvement stays consistent even after receiving 100 training samples per class. This demonstrates the *higher asymptote* advantage for knowledge transfer (see section 2.2). In this particular case such an advantage is also theoretically guaranteed by the fact that the knowledge transfer problem is solved in a higher dimensional feature space than that used for no transfer. The same performance cannot be expected for KT as already discussed in section 3.5.6. In all the plots MKTL outperforms the average transfer even if the advantage is not always significative: this behavior is different with respect to what observed in the previous set of experiments on the Caltech dataset, and it is relevant to underline that the AwA dataset contains only animals.

Finally it is also worth mentioning that our learning algorithm, by relying on the OBSCURE implementation, results very efficient and takes less than 1 minute to finish on the AwA dataset with 100 training sample per categories and a 40 class source knowledge.

5.6.4 Mixing Old and New Classes

Knowledge transfer methods are usually evaluated considering disjoint source and target problems. The source knowledge is used as reference when learning on the target, but the final method is not challenged to distinguish the source *from* the target. Only few attempts has been made in this sense, showing that if we want to exploit the similarity between old and new knowledge, then it becomes difficult to discriminate among them [138].

Here we show that MKTL does not suffer from this drawback. We re-ran the experiment presented in the previous section with the F2 setting and we added to the target task 5 and 10 classes randomly extracted from the source set of 40 categories. This means that in the target task we have respectively 15 and 20 classes. Of course the samples used in the target where not considered previously in the source.

---

5. We used the precalculated descriptors available from http://attributes.kyb.tuebingen.mpg.de/

The results shown in Figure 5.7 are coherent with that already discussed in the previous section: MKTL still performs significantly better than no transfer and prior-features. Moreover, the per-class results show that there is always a clear advantage in the recognition rate of the classes already known from the source, while the new classes benefit in different extent from the transferred knowledge.

5.7 Discussion

Despite the great variety of existing transfer learning methods in the computer vision and machine learning literature, they all generally assume a strong control over the prior knowledge, whether in the form of constraining how the models are built [59, 165], or in the way of preserving the priors training samples [39, 43], or in the form of imposing the same feature representation for all priors and for the new target class [43, 165]. Moreover, the vast majority of knowledge transfer approaches are designed for binary problems discarding multiclass tasks. However the use of source information might be beneficial especially in this setting when the number of categories grows and it becomes harder to get enough annotated data for training standard learning methods.

The MKTL algorithm proposed in this chapter covers these two issues: it is a multiclass transfer learning algorithm based on 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 by solving an optimization problem which considers both from where and how much to transfer using a principled multiclass formulation. We modeled our learning algorithm using the structural risk minimization principle, with a group norm regularization term which allows us to tune the level of sparsity in the domain of the prior models. We showed that it is possible to cast the problem within the multi-kernel learning framework, and to solve it efficiently with off-the-shelf MKL solvers.

We showed experimentally the effectiveness of MKTL in the binary, domain adaptation and multiclass transfer framework, even in case of partial superposition between the source and the target task. The obtained results indicate that the best setting to use MKTL on multiclass problems is when at least 5 training samples per class are available, in case of many prior knowledge sources and if the source and the target features are heterogeneous.
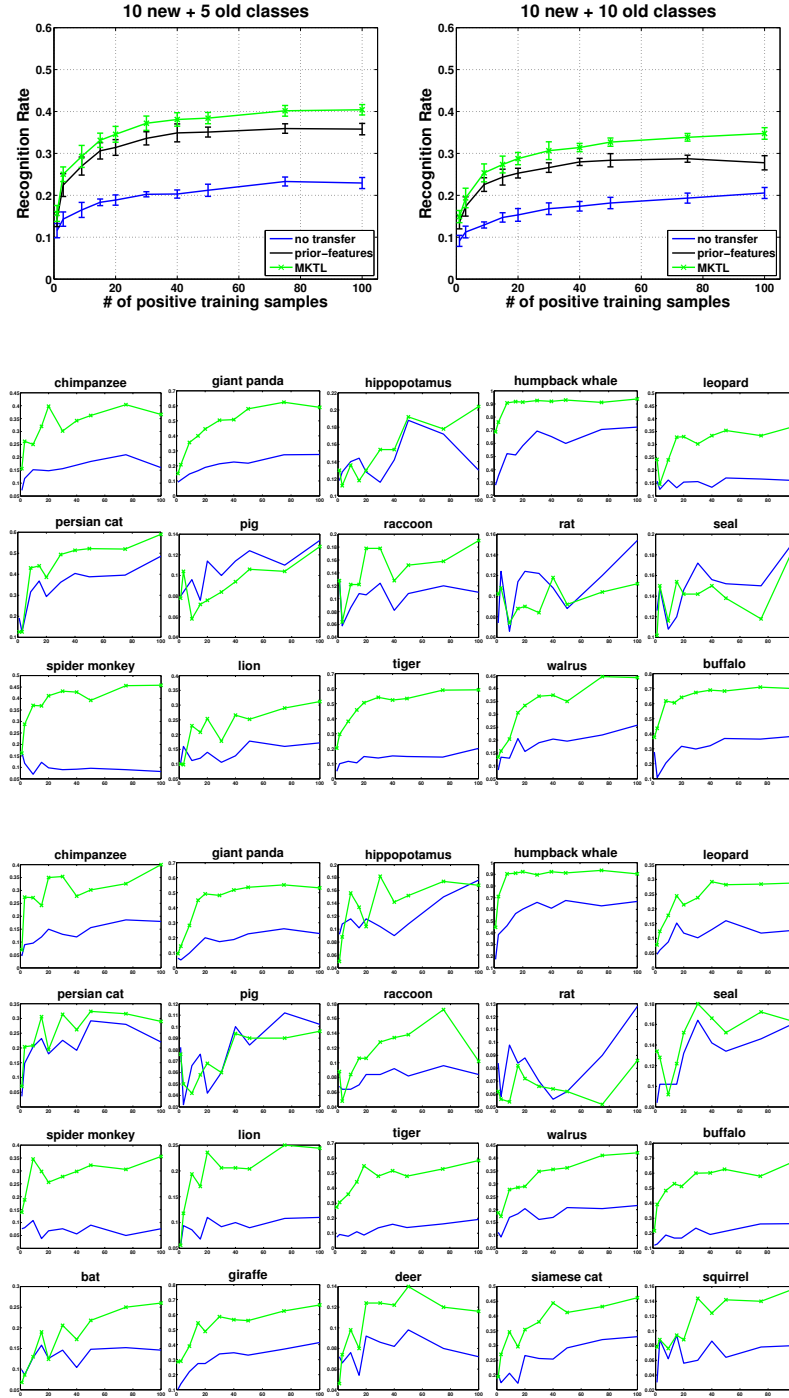
Figure 5.7 – Results on the Animals with Attributed dataset in case of a superposition between the source and the target label set. Here we added samples of the source categories spider monkey, lion, tiger, walrus, buffalo, bat, giraffe, deer, siamese cat and squirrel to the target in two groups of five classes. We show first the average recognition rate over all the classes and then the corresponding results per class (first the 10+5 and then the 10+10). The experiments were run with the described F2 setting and with the C parameter chosen as the best on the target by no transfer.

Part III
Moving Forward

6 Transfer Initialized Online Learning

Open ended learning is a dynamic process based on the continuous processing 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). Generally these two aspects of the learning process are tackled separately. In this chapter we propose a strategy to take the best of both worlds. We present a theoretical analysis coupled with extensive experiments on visual classification problems to show that our approach performs well in terms of online number of training mistakes as well as in terms of performance on separate test sets.

6.1   Motivation

An efficient artificial intelligent system should be able 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.

The ability to learn a new object 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 on different but related source classes when learning a new one, especially in presence of few training data. A second component is that of being able to update continuously the learned category, as new samples 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.

Most of the existing transfer learning strategies are based on batch approaches and re-evaluate

the relevance of the source knowledge for the new task every time a new target training sample is available. This results in a considerable effort in terms of computational complexity. On the other hand, online learning methods are usually evaluated in terms of total mistakes on the progressively incoming samples, and data that are not in this sequence are considered irrelevant. However, this might result in shortsighted learning approaches with weak generalization properties.

Here we propose to merge transfer and online learning such that each of them gets a benefit from the other to overcome the described issues. The idea is to use 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) it is possible to show theoretically that a good initialization for the online learning process produces a tight mistake bound, 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.

Up to our knowledge, only [189] presents an online transfer learning method (OTL) and it is based on ensemble learning. It builds online a prediction function on the data of the target task and mixes it with the old prediction function learned on the source. 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 and has been analyzed only in terms of standard mistake rate.

Regarding the idea of exploiting a good initialization for online learning, it has been shown that for recommender problems [1] and robotics applications [46] this solution is very helpful. [144, 149] 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 [71] 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.

In the following we recap some basic elements of the online learning theory together with the OTL approach proposed in [189], before presenting our method that we name TRansfer initialized Online Learning (TROL, [167]).

6.2   Online Learning

We use here the notation and the general mathematical framework already introduced in the previous chapters. We focus on binary problems 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 assume without loss of generality that $\|\boldsymbol{x}\| \leq 1$. We also consider the hinge loss $\ell^H(\boldsymbol{w} \cdot \boldsymbol{x}, y)$ with margin 1 of a classifier $\boldsymbol{w}$ over an instance/label pair $(\boldsymbol{x}, y)$.

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 [30]. 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.

6.2.1 The Passive Aggressive Algorithm

The Passive Aggressive algorithm (PA) was presented in [35]. 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

$$\boldsymbol{w}_{t+1} = \underset{\boldsymbol{w}}{\text{argmin}} \frac{1}{2} \|\boldsymbol{w} - \boldsymbol{w}_t\|^2 + C\xi \tag{6.1}$$

subject to $\quad \ell^H(\boldsymbol{w} \cdot \boldsymbol{x}_t, y_t) \leq \xi \quad$ and $\quad \xi \geq 0$, $\tag{6.2}$

that results in a simple closed form

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \gamma_t y_t \boldsymbol{x}_t \quad \text{where} \quad \gamma_t = \min\left\{C, \frac{\ell^H(\boldsymbol{w}_t \cdot \boldsymbol{x}_t, y_t)}{\|\boldsymbol{x}_t\|^2}\right\}, \tag{6.3}$$

and C is the aggressiveness parameter that trades-off the two quantities in (6.1). 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 [35], 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 $\mathcal{O}(T^2)$.

For PA initialized with the null vector $\boldsymbol{w}_1 = (0, \ldots, 0) \in \mathbb{R}^d$ it is possible to prove the following mistake bound

Theorem 1. *[35] Let* $(\boldsymbol{x}_t, y_t)$, $t = 1, \ldots, T$ *be a sequence of examples where* $\boldsymbol{x}_t \in \mathbb{R}^d$, $y_t \in$

---

1. In particular we consider here the Passive Aggressive version defined as PA-I in [35] but we call it PA here for simplicity.

$\{+1, -1\}$ *and* $\|\boldsymbol{x}_t\| \leq 1$ *for all* t . *Then, for any vector* $\boldsymbol{u} \in \mathbb{R}^d$ *the number of prediction mistakes* M *made by PA on this sequence of examples is bounded from above by*

$$M \leq 2 \max\left\{1, \frac{1}{C}\right\} \left(\frac{1}{2}\|\boldsymbol{u}\|^2 + C \sum_{t=1}^{T} \ell^H(\boldsymbol{u} \cdot \boldsymbol{x}_t, y_t)\right), \tag{6.4}$$

where C is the aggressiveness parameter provided to PA.

Since $\boldsymbol{u}$ is arbitrary we can always define it as the solver of the SVM problem in batch mode on the full set of samples seen by the online learner (the SVM objective function correspond to the content of the round brackets in (6.4) [37]). Hence the performance of PA in terms of mistakes over the data sequence is close to the one of a batch optimal classifier, measured with respect to the hinge loss.

6.2.2 OTL: Online Transfer Learning

The OTL algorithm proposed in [189] 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 task. Specifically in the first step 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 (6.3) with $f_t(\boldsymbol{x}_t) = \boldsymbol{w}_t \cdot \boldsymbol{x}_t$. Then the second step is the combination of prior and new knowledge: the sample class label is predicted by the following ensemble function [189]:

$$\hat{y}_t = \text{sign}\left(\sigma_t \Pi(h(\boldsymbol{x}_t)) + \tau_t \Pi(f_t(\boldsymbol{x}_t)) - \frac{1}{2}\right), \tag{6.5}$$

where $\Pi(x) = \max\left\{0, \min\left\{1, \frac{x+1}{2}\right\}\right\}$ 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 [189]

$$\sigma_{t+1} = \frac{\sigma_t s_t(h)}{\sigma_t s_t(h) + \tau_t s_t(f_t)}, \qquad\qquad \tau_{t+1} = \frac{\tau_t s_t(f_t)}{\sigma_t s_t(h) + \tau_t s_t(f_t)}, \tag{6.6}$$

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

The proposed method originally assumes the existence of one unique source. 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 possible solution consists in assigning one weight to each source knowledge. In this case we start from $\sigma_1 = \sum_{j=1}^{J} \sigma_{j,1} = \tau_1 = \frac{1}{2}$ with $\sigma_{j,1} = \frac{1}{2J}$ for $j = 1, \ldots, J$ and then we update the weights with

$$\sigma_{j,t+1} = \frac{\sigma_{j,t} s_t(h_j)}{\sum_{j=1}^{k} \sigma_{j,t} s_t(h_j) + \tau_t s_t(f_t)}, \qquad \tau_{t+1} = \frac{\tau_t s_t(f_t)}{\sum_{j=1}^{k} \sigma_{j,t} s_t(h_j) + \tau_t s_t(f_t)}. \tag{6.7}$$

If we neglect the prior knowledge learning process, the total computational complexity of OTL

matches the one of the online learning method used, since the cost of (6.6) (and (6.7)) 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 the OTL algorithm has a theoretical support given by the possibility to prove an upper bound on the number of mistakes made during the online learning process.

Theorem 2. *[189] Let us denote by M the number of mistakes made by the OTL algorithm, we have then M bounded from above by :*

$$M \leq 4\min\{\Sigma_h, \Sigma_f\} + 8\ln 2 \,, \tag{6.8}$$

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 [189] 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 that $\ell^S(\Pi(h(\boldsymbol{x}_t)), \Pi(y_t)) \approx \frac{1}{4}M_h$ and $\ell^S(\Pi(f_t(\boldsymbol{x}_t)), \Pi(y_t)) \approx \frac{1}{4}M_f$, then $M \leq \min\{M_h, M_f\} + 8\ln 2$.

6.3 TRansfer initialized 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. We propose here an alternative solution for online and transfer learning integration.

In chapter 3 we presented the KT algorithm that aims at learning the target task model $\boldsymbol{w}$ from few examples, constraining it to be close to a linear combination of known $\hat{\boldsymbol{w}}_j$ related to source classes. For simplicity we report here the KT learning problem

$$\min_{\boldsymbol{w},b} \frac{1}{2} \left\| \boldsymbol{w} - \sum_{j=1}^{J} \beta_j \hat{\boldsymbol{w}}_j \right\|^2 + \frac{C}{2} \sum_{i=1}^{N} (y_i - \boldsymbol{w} \cdot \boldsymbol{x}_i - b)^2 \,. \tag{6.9}$$

We recall that the weights β_j assigned to each prior knowledge are found by minimizing $\sum_{i=1}^{N} \ell^H(\tilde{y}_i, y_i)$ subject to $\|\boldsymbol{\beta}\|_2 \leq 1$ and $\beta_j \geq 0$, where $\tilde{y}_i$ is the leave-one-out prediction for the i−th sample, and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_J)$.

6.3.1 TROL : Fixed transfer weights

KT is a batch approach directly meant to minimize the generalization error of the obtained target model. Since it operates in the small setting scenario, we can use it to define an hybrid batch-online learning approach based on two phases: at the beginning N target training samples are given as input to 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 KT together with the proper introduction of new information when necessary.

Formally, training KT on N target samples (typically $N \leq 10$) consists in solving the optimization problem in (6.9). The obtained model is then introduced in (6.1) as initialization (w_1) when learning from the $(N+1)$−th training sample on. Hence the final cost is $\mathcal{O}(T^2 + N^3 + JN^2)$, that for enough samples T is dominated by the complexity of PA. In other words the added complexity of using KT on N samples is negligible.

Theoretical Analysis. With respect to PA, a good initialization model can improve the mistake bound. In fact we can generalize (6.4) to the case of using a w_1 different from the null vector. In this way the number M of prediction mistakes satisfies

$$M \leq 2 \max\left\{1, \frac{1}{C}\right\} \left(\frac{1}{2}\|u - w_1\|^2 + C \sum_{t=1}^{T} \ell^H(u \cdot x_t, y_t)\right). \tag{6.10}$$

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

6.3.2 TROL+ : 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 w_1, we can evaluate its prediction on each new training samples as $w_1 \cdot x_t$. We crop the obtained value between -1 and 1, similarly to OTL, to limit the norm of the added dimension. The prediction is then used as the $(d + 1)$-th element in the feature vector descriptor of x_t. So we define

$$x'_t = (x_t, v_t) \in \mathbb{R}^{d+1} \quad \text{where} \quad v_t = \max\{-1, \min\{1, w_1 \cdot x_t\}\}.$$

The samples with such a modified representation enter the PA algorithm initialized now with $w'_1 = (0, \ldots, 0, 1) \in \mathbb{R}^{d+1}$. At $t = 1$ PA predicts with $\text{sign}(w'_1 \cdot x'_1) = \text{sign}(v_1)$ while for any t the updating rule in (6.3) results in

$$w'_{t+1} = w'_t + \gamma_t y_t x'_t \quad \text{where} \quad \gamma_t = \min\left\{C, \frac{\ell^H(w'_t \cdot x'_t, y_t)}{\|x'_t\|^2}\right\}, \tag{6.11}$$

and the predictions are

$$\boldsymbol{w}'_t \cdot \boldsymbol{x}'_t = \sum_{i=1}^{t-1} \gamma_i y_i (\boldsymbol{x}_i \cdot \boldsymbol{x}_t + v_i v_t) \,. \tag{6.12}$$

Hence the hyperplane $\boldsymbol{w}'_t$ can be thought as composed of 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 J different prior models $\boldsymbol{w}_1^j$, $j = 1, \ldots, J$, expanding the input vectors with J new dimensions

$$\boldsymbol{x}'_t = (\boldsymbol{x}_t, v_{1,t}, \ldots, v_{J,t}) \in \mathbb{R}^{d+J} \quad \text{where} \quad v_{j,t} = \max\{-1, \min\{1, \boldsymbol{w}_1^j \cdot \boldsymbol{x}_t\}\} \,. \tag{6.13}$$

Theoretical Analysis. From the bound (6.10), taking into account the increased dimensionality of the instances, we can derive directly the following

Theorem 3. *[167] Let $(\boldsymbol{x}'_t, y_t)$, $t = 1, \ldots, T$ be a sequence of transformed instances as in (6.13), $y_t \in \{+1, -1\}$ and $\|\boldsymbol{x}_t\| \leq 1$ for all t. Then, for any vector $\boldsymbol{u} \in \mathbb{R}^{d+J}$ the number of prediction mistakes made by TROL+ on this sequence of examples is bounded from above by*

$$M \leq 2 \max\left\{(1+J), \frac{1}{C}\right\} \left(\frac{1}{2}\|\boldsymbol{u} - \boldsymbol{w}'_1\|^2 + C \sum_{t=1}^{T} \ell^H(\boldsymbol{u} \cdot \boldsymbol{x}'_t, y_t)\right), \tag{6.14}$$

where C is the aggressiveness parameter provided to TROL+.

To compare this bound to the one of OTL, let us set $C = 1$ and use only one prior knowledge, i.e. $J = 1$. Given that the bound in (6.10) holds for any $\boldsymbol{u}$, we can worsen (6.14) by setting $\boldsymbol{u}$ to be the optimal one for the new knowledge alone or the prior knowledge alone. As a consequence we have that

$$M \leq 4 \min\{\Sigma_h, \Sigma_f\}$$

$$\text{where} \quad \Sigma_h = \sum_{t=1}^{T} \ell^H(v_t, y_t) \leq \sum_{t=1}^{T} \ell^H(w_1 \cdot \boldsymbol{x}_t, y_t)$$

$$\text{and} \quad \Sigma_f = \min_{\boldsymbol{u}} \frac{1}{2}\|\boldsymbol{u}\|^2 + \sum_{t=1}^{T} \ell^H(\boldsymbol{u} \cdot \boldsymbol{x}_t, y_t) \,. \tag{6.15}$$

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 [17, 140]. Moreover, as discussed in section 6.2.2, the OTL bounds does not directly link the performance to the two stages of the algorithm. On the other hand, TROL+ integrates the prior knowledge directly in the target learning process in one unique layer. 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
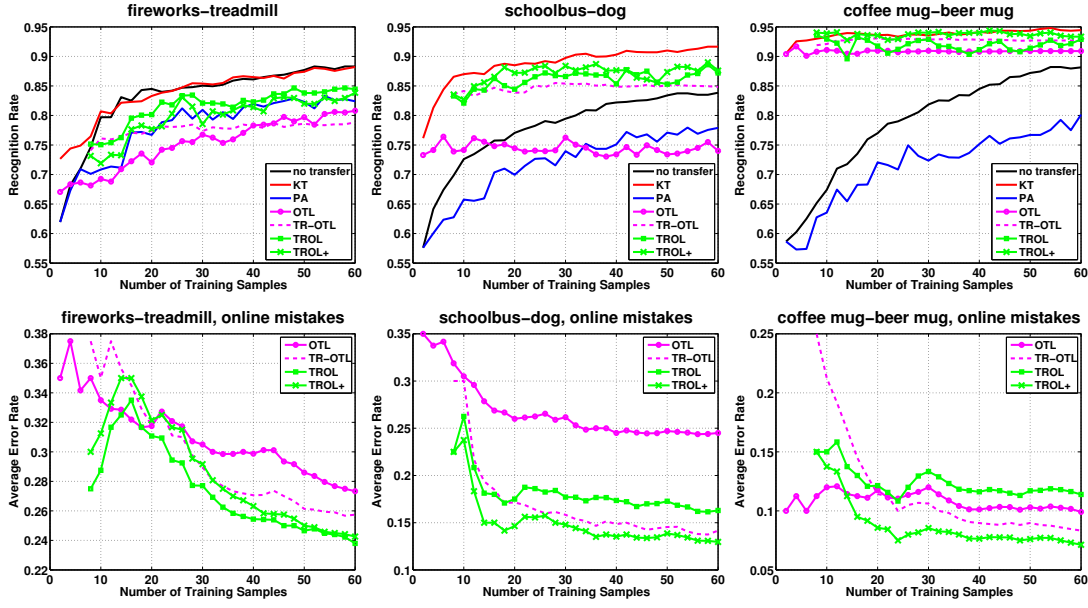
Figure 6.1 – Single source experiments: pairs of classes with increasing relatedness from left to right, as empirically shown by the growing advantage of KT over no transfer. Top line: recognition rate results on the test set plotted as a function of the number of training samples. Bottom line: corresponding rate of mistakes for the online learning methods. Here the performance of PA is always much worse than the considered methods so we neglect the corresponding line in the plots for the sake of clarity.

all the samples. In the next section we will show that this theoretical advantage is also evident in the empirical experiments.

6.4 Experiments

We performed experiments on visual object classification problems using the Caltech-256 dataset and following the setting already defined in chapter 3. Feature-wise, we used the publicly available SIFT descriptors of [61]. 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 all these 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 is organized such that any positive sample is always followed by a negative one and vice-versa. For all experiments we used the Gaussian kernel fixing γ 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, the second is a linear kernel applied on the extra feature elements obtained by the prediction of the priors.
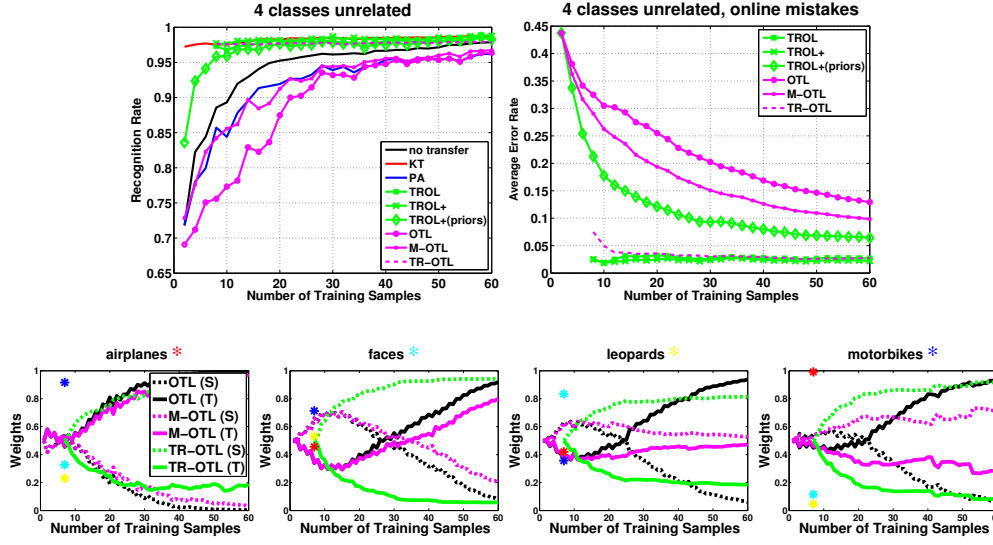
Figure 6.2 – Four unrelated classes: airplanes, motorbikes, faces and leopards. Top left: recognition rate results on the test set plotted as a function of the number of training samples. Top right: corresponding rate of mistakes for the online learning methods. Second row: value of the weights given 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 KT and used in the input model to TR-OTL.

We benchmarked TROL and TROL+ against PA trained on the target samples, 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:

no transfer : this is a batch strategy corresponding to learning using only the target data. LS-SVM is applied 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 (6.7).
TR-OTL : this method considers as source knowledge for OTL the same KT output that we use as initialization in TROL and TROL+.

All the online techniques initialized with 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 (KT and no transfer) applied on the new task. The C value for all the online methods is instead fixed to 1.
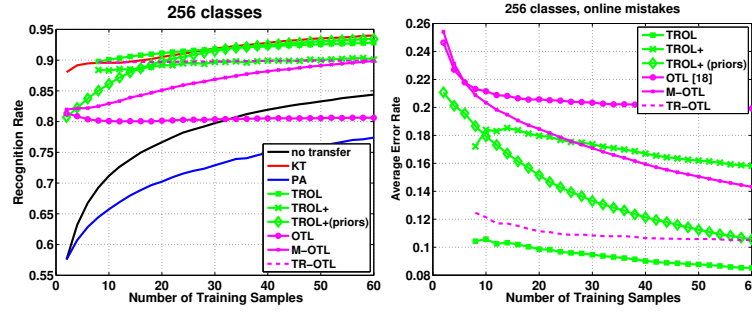
Figure 6.3 – Recognition rate results on the test set plotted as a function of the number of training samples and corresponding average error rate for the online methods on the whole Caltech-256 dataset. All the results are obtained as average over each of the classes considered as target task with the remaining 255 used as sources.

6.4.1 Single source

We ran a first group of experiments considering the case of a single available source, the same setting in which OTL was originally presented and evaluated. We considered different pairs 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 pairs we considered 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 6.1. For the unrelated pair fireworks-treadmill (left column), TROL and TROL+ matches the recognition performance of the corresponding no transfer online learning method PA, while OTL and TR-OTL suffer from 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 mistake rate performance. 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.

In general we can state that initializing the online learning process with KT is always beneficial and produces better results with respect to OTL and learning from scratch with PA. In terms of recognition rate on a new test set the performance of TROL is always comparable or only slightly worse than what can be gathered with the batch KT method.

6.4.2 Multiple sources

We focus here on the case where multiple priors are available. Figure 6.2 shows the results on a group of four unrelated classes (originally used in [57]). Each of them is considered in turn as target task while the remaining ones define three source knowledge sets. Despite the difference among the object categories, KT is able to define a good combination of priors and exploits it when learning on the target, obtaining extremely good results in classification.

All the online methods initialized with KT (TROL, TROL+, TR-OTL) matche the recognition performance of the batch transfer method after 10 training samples. OTL considering the average source knowledge shows instead negative transfer. The corresponding M-OTL version, based on different weights for each source classifier, does not have any advantage with respect to 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 $J = 3$ 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 KT after 20 training samples.

The four plots on the second row in Figure 6.2 show how the weights given to source and target knowledge change in the OTL-related methods. The information obtained as output from KT, used as source in TR-OTL, maintains a high weight in time. This demonstrates its usefulness for the learning process. We also see that the source knowledge loses its importance in time, or show a small weight, for OTL and M-OTL.

Figure 6.3 presents the results for the full Caltech-256 dataset. Here the online method TROL performs as the batch algorithm KT and shows the best results with respect to 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 priors outperforms OTL and M-OTL, and matches TR-OTL in terms of average error rate with 60 training samples.

6.5 Conclusion and Discussion

We addressed the issue of open ended learning of visual categories and we proposed an approach based on the combination of our KT algorithm with online learning. It results into a method where the available source knowledge is used in a principled manner to initialize the online learning process. This allows us to use the potentiality of the transfer method without paying the computational cost of a batch approach, possibly limited to an initial budget.

If TROL builds directly over KT, the variant TROL+, that considers a feature augmentation approach, is actually related to MKTL as presented in chapter 5. The value of the proposed solutions is demonstrated both theoretically by the derived mistake bounds, and empirically by the experimental results both in terms of error rate on the training sequence and on an unseen test set.

Although our online transfer approaches appear promising on visual object categorization problems, to fully analyze their potential it would be interesting to consider an experimental testbed where the data present a clear concept drift with gradual changes in time. One possible

scenario could be that of video streams such as images from surveillance cameras.

7 Multi-task Unaligned Shared Knowledge Transfer

Many visual data resources have been created for research purposes in the last years. Each one presents a specific focus which ranges from object classification, detection and segmentation to scene categorization, and learning with captions or attributes. Despite the particular goal that motivated their authors, all these collections aim at representing the real world and consequently the distribution of the object categories depicted in the images is not uniform. Some classes are frequent and usually shared across the datasets, others are rare and considered only in some of them. Moreover the shared classes, although presenting some general aspect in common, suffer from biases related to the collection process. Being able to use those existing resources for real applications means facing a multi-task problem over data with unaligned label sets, and combine domain adaptation with knowledge transfer to exploit the extracted information on a new task. We propose here an image representation that decomposes orthogonally into two subspaces: a part specific to each dataset and a part generic to, and therefore shared between, all the considered source sets. Through experiments on five public datasets we show that our approach allows cross-datasets generalization.

7.1 Motivation

The long standing ambition of the visual recognition community is to enable artificial visual systems to recognize reliably not only specific instances of a category, such as *my car*, but *cars* in general. Many visual databases (e.g. Caltech-101 [58], PASCAL VOC [53], Animals with Attributes [89], ImageNet [47]) have been created to support such quest. However, recent studies [169, 129] have questioned if the results obtained so far are a reliable indicator of real generalization abilities. Indeed, it seems that high performance on a data collection often does not reflect on the ability to classify correctly the same classes, imaged in another dataset.

One of the main reasons behind this problem is the data selection bias [169]: images contained in two databases under the same category label can represent instead different related subcategories, e.g. in ImageNet the class *car* has a strong preference for race cars. Conversely (category label bias [169]), it might happen that different labels are used for the same type of
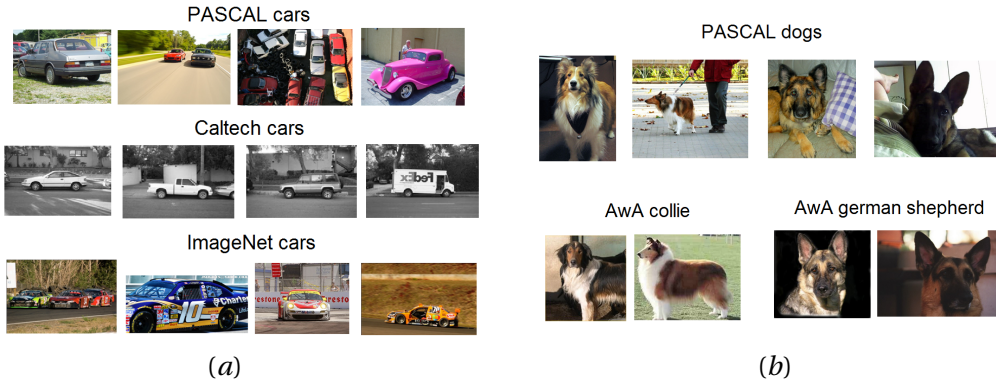
Figure 7.1 – (a) An example of the data selection bias; (b) An example of the category label bias (AwA refers to Animal with Attributes).

object, e.g. the class *dog* in PASCAL VOC presents images of *collie* and *german shepherd* breed dogs that correspond instead to two separate classes in Animals with Attributes (see Figure 7.1).

When looking at the disappointing cross-dataset generalization results reported in [169] keeping in mind the biases described above, one could formulate an hypothesis: a classifier trained on a specific dataset learns, for each object class, a model containing some generic knowledge about the semantic categorical problem, and some specific knowledge about the bias contained into that dataset. For example for the object category *car*, a classifier trained on ImageNet would learn a racing car model. Still, the specific ability to classify correctly race cars implies having some knowledge about the general category car.

Issues arise even when focusing only on common classes across multiple existing datasets, as their label name is not sufficient to select and align them. It is necessary to inspect visually their content or use a pre-defined hierarchical ontology (like Wordnet [153]). Moreover, analyzing one class at a time implies the definition of binary problems where the negative class is obtained by sampling from the remaining set of classes, specific to each database. Thus, the definition of *what an object is not* is intrinsically biased (negative bias [169]).

In this chapter we propose a method to overcome these issues. We exploit existing visual datasets preserving their multiclass structure and relying on the fact that they are many: each of them presents specific characteristics, but all together they cover different nuances of the real world. As the data are not uniformly distributed [145], it often happens that some classes overlap across the datasets, giving us the possibility to learn on them decoupling explicitly the generic and specific knowledge. Our Multi-task Unaligned Shared knowledge Transfer (MUST, [168]) algorithm learns jointly a shared and a private image representation from multiple datasets, and then transfers the common information when training on a new dataset. Considering an heterogeneous learning condition, we show that it is even possible to exploit features previously proposed, pre-computed and publicly available for download for
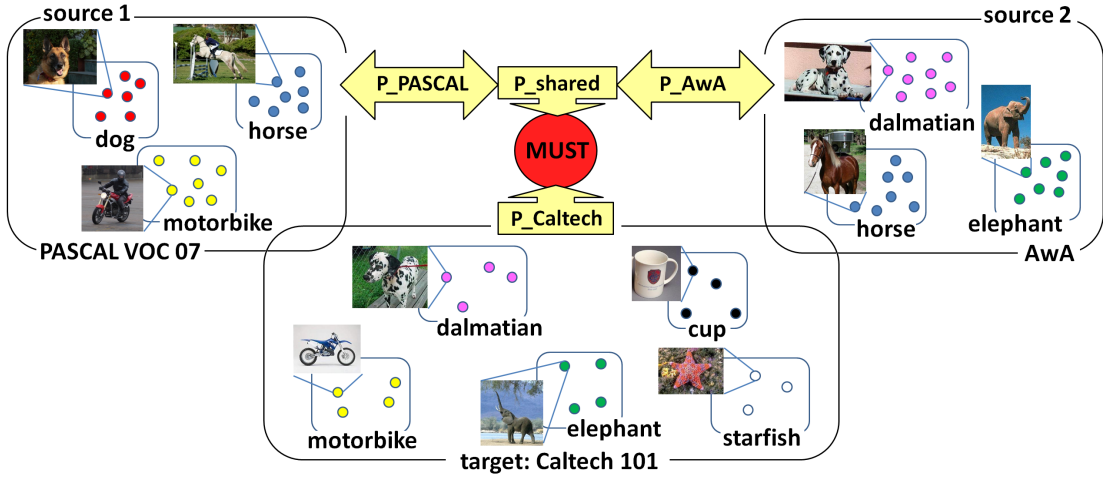
Figure 7.2 – Schematic representation of the MUST algorithm [168]: shared and private information are extracted from two existing datasets (AwA stands for Animals with Attributes). The shared knowledge is then transferred to solve a new multiclass problem on a different dataset. Notice that no explicit alignment is requested between *dalmatian* and *dog* classes.

each dataset.

This approach faces the dataset bias [169] without restricting it to binary problems, tackling directly the realistic setting of multiclass tasks with heterogeneous features: often the biases are induced by a specific research focus which turns in some features being more appropriate for some databases. Moreover we define a leave-one-dataset-out experimental setup over five existing datasets (Caltech-101, PASCAL VOC 07, MSRCORID, Animal with Attributes and CIFAR 100) that can be considered a valid test bed for any cross-dataset generalization method.

7.2   Problem Statement

Let's consider the problem of learning a classifier on a target task t when J source tasks are available, in the hypothesis of a domain shift across them and partial overlapping label sets. The difference in the domains can be caused by both a distribution mismatch and by different feature descriptors.

Our aim is to be able to extract general information from all the sources in *multi-task* fashion, and to use it when learning on a new target. We expect a general advantage both on the known categories, as in *domain adaptation*, and on new ones as in *transfer learning*. Thus we define an approach that fits in the global framework of all the mentioned methods, while at the same time covering issues orthogonal across all of them. With respect to classical multi-task learning, we break the symmetry adding a transfer part to a target problem. At the same time, we overcome the transfer learning issue of evaluating the task relatedness leveraging on the possibility to extract a common useful knowledge from multiple sources. Finally, we go beyond

domain adaptation which does not cover the case of completely new classes in the target task. Moreover, considering multiple sources (with possibly different features) we show that the hypothesis of relying on a flat average knowledge is not helpful in the case of tasks with partially overlapping label sets.

We pursue our goal by defining a method that allows us to exploit existing visual resources with a *minimal effort* every time we would like to learn on a new multiclass problem. The effort is low in three aspects: (1) we do not need to know explicitly which classes are present in each task and therefore, no manual alignment is necessary, (2) we do not need to keep the source data when learning on the target, (3) we leverage over multiple sources regardless of their feature space. MUST is inspired by recent research on finding shared and private projections [146, 93, 72] for multi-view setting. In problems where multiple modalities or multiple views of the same data are available, it has proven useful to factorize the information and learn separate latent spaces for modeling the shared (correlated) and private (independent) parts of the data. Recently, [125] exploited this notion in the context of multi-task learning.

7.3 Formulation

Starting from the availability of multiple visual object datasets, our goal is to use them to learn a projection function that maps the data points into one shared and several private latent spaces with an *orthogonality* constraint between them. We can then transfer the knowledge encoded in the shared space to a new dataset and use the available training samples to learn only the remaining private orthogonal part (see Figure 7.2). The new problem will benefit from this approach only if the shared space captures generic, non-dataset-specific, information which we will call *common sense*.

More formally, we are given J sets of N_j observed data points, $\mathcal{D}_j = \{(\boldsymbol{x}_1^j, y_1^j), \ldots, (\boldsymbol{x}_{N_j}^j, y_{N_j}^j)\} \subset \mathcal{X}^j \times \mathcal{Y}^j$ for $j = 1, \ldots, J$. Here we use $\mathcal{X}^j$ and $\mathcal{Y}^j$ to denote the input space and output space of the j-th dataset. For the purpose of explaining the key idea, we assume that the same representation is used for all the datasets, thus all of them have the same dimensionality, $\mathcal{X}^j = \mathbb{R}^d$ for all j. We discuss the more general case of each dataset admitting its own representation, and therefore living in a different dimensional space in section 7.4. We further require some overlap in the output spaces, i.e. $\mathcal{Y}^i \cap \mathcal{Y}^k \neq \emptyset$ for all $(i, k) \in \{1, \ldots, J\}$. The existence of such partial superposition in the label sets allows us to introduce the notion of common sense as generic knowledge among the tasks. It is important to underline that we want to define an approach which does not require explicit label correspondences among datasets, and we are interested in models that do not build those correspondences as an intermediate learning step.

We seek functions

$$g_j : \mathbb{R}^d \to \mathbb{R}^D \quad \text{for} \quad j = 1, \ldots, J, \tag{7.1}$$

which project the original space into a novel one with potentially much smaller dimension $D \ll d$. We assume a *linear* parametrization of the functions and an *additive* model for the shared-private spaces. Thus, the projection functions admit the following form

$$g_j(\boldsymbol{x}^j) := (P_j + P_s)\boldsymbol{x}^j \tag{7.2}$$

for a private projection matrix over the j-th dataset $P_j \in \mathbb{R}^{D \times d}$, and a shared projection matrix $P_s \in \mathbb{R}^{D \times d}$. We learn those matrices based on the *folk-wisdom* principle [63, 179] of pulling objects or data samples together if they are of the same type (keeping your friends close), and pushing them apart if they are not (keeping your enemies far away). This principle is formalized by the regularized risk functional described in the following section.

7.3.1 Regularized Risk Functional

We want to learn a transformation over the data by minimizing a function which penalizes large distances between samples of the same class, and small distances between samples with non-matching class labels. We assume that for each sample, it is possible to identify a set of genuine neighbors or friends. The notation $i \sim k$ is used to indicate that $\boldsymbol{x}_i$ and $\boldsymbol{x}_k$ are friends as belonging to the same class, and the notation $i \nsim l$ describes that $\boldsymbol{x}_i$ and $\boldsymbol{x}_l$ are enemies as associated to different class labels. Our optimization problem has the following form:

$$\min_{\substack{P_s \\ P_{1,\dots,J}}} \eta\Omega(P_j) + \gamma\Omega(P_s) + \sum_{j=1}^{J} \underbrace{\sum_{i \sim k} d_j^2(\boldsymbol{x}_i^j, \boldsymbol{x}_k^j) + \sum_{\substack{i \sim k \\ i \nsim l}} \max\left\{0, 1 + d_j^2(\boldsymbol{x}_i^j, \boldsymbol{x}_k^j) - d_j^2(\boldsymbol{x}_i^j, \boldsymbol{x}_l^j)\right\}}_{\text{Loss}(\cdot)} \tag{7.3}$$

subject to $\quad P_s^\top P_j = 0 \quad$ for all $\quad j = 1, \dots, J$,

where $d_j^2(\boldsymbol{x}_i^j, \boldsymbol{x}_k^j) := \|(P_j + P_s)(\boldsymbol{x}_i^j - \boldsymbol{x}_k^j)\|^2$ is the squared distance in the projected space. In (7.3), Loss$(\cdot)$ is the loss function, $\Omega(\cdot)$ is a regularizer on the projection matrices, and the trade-off variables η and λ control the relative influence of loss and regularization terms. For $\Omega(\cdot)$, one typically chooses the L_2 norm, or the L_1 norm if one wants to induce sparsity in the projection matrices. The loss function consists of two terms: the first requires small distances among friend samples, while the second asks that the distance between each sample and its enemies is a unit greater than the corresponding distance to the friends. Finally, the constraints ensure that the inferred shared space is orthogonal to each of the private spaces.

Given a new dataset of N_t observed data points $\mathcal{D}_t = \{(\boldsymbol{x}_1^t, y_1^t), \dots, (\boldsymbol{x}_{N_t}^t, y_{N_t}^t)\} \subset \mathbb{R}^d \times \mathcal{Y}^t$ with $\mathcal{Y}^t \cap (\bigcup_{j=1,\dots,J} \mathcal{Y}^j) \neq \emptyset$ we want to learn its specific representation while enforcing it to be orthogonal to the common sense obtained from the previous J datasets.

This corresponds to finding a private projection matrix P_t *given* the shared projection matrix

P_s , and can be expressed with the following optimization problem:

$$\min_{P_t} \quad \eta\Omega(P_t) + \sum_{i \sim k} d_t^2(\boldsymbol{x}_i^t, \boldsymbol{x}_k^t) + \sum_{\substack{i \sim k \\ i \neq l}} \max\{0, 1 + d_t^2(\boldsymbol{x}_i^t, \boldsymbol{x}_k^t) - d_t^2(\boldsymbol{x}_i^t, \boldsymbol{x}_l^t)\} \tag{7.4}$$

$$\text{subject to} \quad P_s^\top P_t = 0,$$

where $d_t^2(\boldsymbol{x}_i^t, \boldsymbol{x}_k^t) := \|(P_t + P_s)(\boldsymbol{x}_i^t - \boldsymbol{x}_k^t)\|^2$. Intuitively, whenever the common sense knowledge given by P_s is sufficient to enforce the folk-wisdom principle, there is no penalty incurred in (7.4). The learning capacity of the private projection matrix P_t can thus be focused on those hard cases specific to this new dataset. In the following section, we go on describing the methods to optimize problems (7.3) and (7.4).

7.3.2 Optimization

The optimization problem (7.3) (and (7.4) likewise) is non-convex with respect to the projection matrices $P_s, P_1, \ldots, _J$, thus it is hard to optimize. However, [179] and more recently [125] presented two ideas to turn a problem analogous to (7.3) into a convex optimization problem through semi-definite programming (SDP). The first idea is to replace the second term of the loss function, with a soft margin constraint. This is achieved by introducing a non-negative slack variable ξ_{ikl} for every pair of friends and enemies such that $d_j^2(\boldsymbol{x}_i^j, \boldsymbol{x}_l^j) - d_j^2(\boldsymbol{x}_i^j, \boldsymbol{x}_k^j) \geq 1 - \xi_{ikl}$. In this way we allow the samples to have the distance to their enemies less than a unit greater than the distance to their friends. To prevent this behavior from occurring often, there is a budget on the slack variables $\sum_{\substack{i \sim k \\ i \neq l}} \xi_{ikl}$ that needs to be minimized. The second intuition is to substitute the optimization over the projection matrix P with the optimization over the corresponding metric $M := P^\top P$, therefore imposing a semi-definite constraint on $M \succeq 0$.

Weinberger and Saul [179] described a convex solver based on alternating sub-gradient descent methods for their learning problem re-formulated according to the introduced tricks. Recently, Kleiner, Rahimi, and Jordan [82] devised an approach to tackle SDPs by repeatedly solving randomly generated optimization problems over two-dimensional subcones of the PSD cone. This approach produces only approximate solutions due to randomization, but it scales to number of samples orders of magnitude larger than have previously been possible. Here, we show that the same solvers can still be used for our constrained problem as the linearity and additive model assumptions allow us to write

$$\begin{aligned} d_j^2(\boldsymbol{x}_i^j, \boldsymbol{x}_k^j) &= \|(P_j + P_s)(\boldsymbol{x}_i^j - \boldsymbol{x}_k^j)\|^2 \\ &= \|P_j(\boldsymbol{x}_i^j - \boldsymbol{x}_k^j)\|^2 + \|P_s(\boldsymbol{x}_i^j - \boldsymbol{x}_k^j)\|^2 \,, \end{aligned} \tag{7.5}$$

and its analogous for $d_t^2(x_i^t, x_k^t)$. The last equality follows directly from our orthogonality assumptions. Note that $\|P_s(\boldsymbol{x}_i^t - \boldsymbol{x}_j^t)\|^2$ is fixed for each set of neighbors and thus can be precomputed. Here, we use the solver presented in [179]. The full method MUST is summarized

in Appendix in Algorithm 6.

7.4 Heterogeneous Features for Multiple Datasets

We now consider the case where each of the given J datasets lies in its own feature space, that is $\mathcal{X}^j = \mathbb{R}^{d_j}$ for $j = 1,\ldots,J$. This setting easily appears since most of the visual datasets are released together with their own pre-extracted features. For this heterogeneous problem, we seek additional projection functions $f_j : \mathbb{R}^{d_j} \to \mathbb{R}^d$ that map all inputs from different databases to an intermediate space in addition to finding the shared and private metrics. We assume a linear parametrization for these functions $f_j := W_j \boldsymbol{x}_i^j$ where $W_j \in \mathbb{R}^{d \times d_j}$ is the projection matrix for the j-th dataset. Our heterogeneous distance with the orthogonality constraint between shared and private spaces made explicit is:

$$
\begin{aligned}
\tilde{d}_j^2(\boldsymbol{x}_i^j, \boldsymbol{x}_k^j) &= (W_j(\boldsymbol{x}_i^j - \boldsymbol{x}_k^j))^\top (P_j^\top P_j + P_s^\top P_s)(W_j(\boldsymbol{x}_i^j - \boldsymbol{x}_k^j)) \\
&= \mathrm{trace}(M_s W_j v_{ik}^j v_{ik}^{j\top} W_j^\top) + \mathrm{trace}(M_j W_j v_{ik}^j v_{ik}^{j\top} W_j^\top) \\
&\quad \text{with} \quad M_s \succeq 0 \quad \text{and} \quad M_j \succeq 0,
\end{aligned}
\tag{7.6}
$$

where $v_{ik}^j = (\boldsymbol{x}_i^j - \boldsymbol{x}_k^j)$. We use the above distance function as a drop-in replacement to the problem in (7.3). Thus the optimization will be over the projection matrices W_js and over the metrics M_s, M_js. Similarly to the homogeneous case, given a new dataset, we will solve the optimization problem in (7.4), but now an additional projection matrix $f_t : \mathbb{R}^{d_t} \to \mathbb{R}^d$ that brings the new datasets to the same intermediate space of the old training datasets has also to be found.

Optimization. The optimization problem in (7.3) with the modified distance function $\hat{d}_j^2(\boldsymbol{x}_i^j, \boldsymbol{x}_k^j)$ is convex with respect to the metrics given all the projection matrices W_js, and is non-convex with respect to the projection matrices given the shared and private metrics M_s and M_j. We pursue an alternating approach: fix all the projection matrices and solve the shared and private metrics M_s and M_j with [179]; subsequently, fix the metrics and optimize all the projection matrices W_js with fast sub-gradient descent algorithm. Here we use nonsmooth BFGS [95]. The heterogeneous version of our method (MUST-HET [168]) is summarized in Appendix in Algorithm 7.

7.5 Experiments

We present here two groups of experiments designed to study how MUST performs on *cross-database* generalization problems both in the case with all sets having the same feature representation (homogeneous setting, Section 7.5.1) and when each of the datasets lies in its own feature space (heterogeneous setting, Section 7.5.2). To this purpose, we selected five visual object databases which are actively used in present computer vision research and have some partial overlap in the label space: Caltech-101 [58] with 101 class labels, PASCAL VOC 07

Classes	Caltech 101	PASCAL VOC 07	MSRCORID	AwA	CIFAR 100
car	X	X	X		
elephant	X			X	X
bottle		X			X
butterfly	X				X
cloud			X		X
dalmatian	X			X	
horse		X		X	X
bicycle		X	X		
sheep		X	X	X	
motorbike	X	X			X
chandelier	X				
grand piano	X				
potted plant		X			
boat		X			
chimneys			X		
doors			X		
walrus				X	
giraffe				X	
mushrooms					X
wardrobe					X
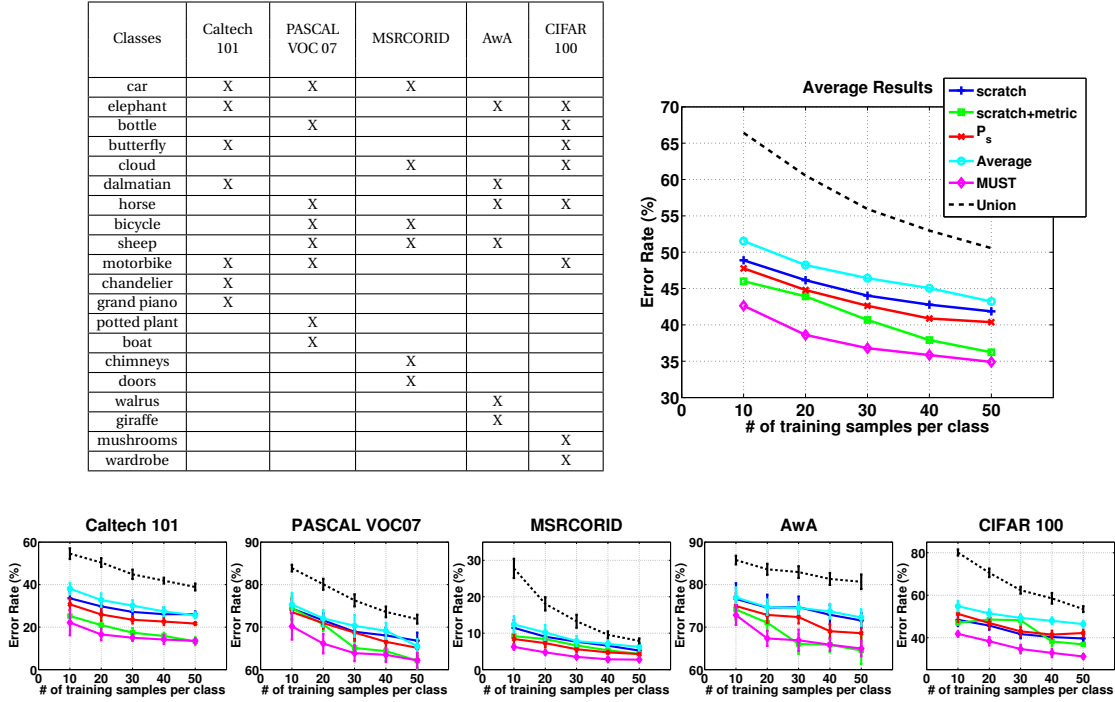
Figure 7.3 – Visual object classification across different datasets using the same feature. Top: (left) the table describing the experimental setup, (right) plot of the average results on five datasets. Bottom: separate results on each target dataset. Results over 10 repetitions for all methods except Union with 5 repetitions due to high computational demand.

[53] with 20 class labels, MSRCORID [114] with 20 class labels, Animals with Attributes (AwA) [89] with 50 class labels, and CIFAR 100 [85] with 100 class labels. We applied a one-dataset-out strategy, extracting the general knowledge from four datasets and giving the chance to each database in turn to be used as a new problem.

7.5.1 Homogeneous setting

For the homogeneous experiments we extracted Gist features [118] (descriptor vector dimension $d_j = d_t = 320$) from the images converted to grayscale and ran metric learning on the sources with 15 genuine neighbors, both considering the multi-task approach (using [125]) and keeping the task separated (using [179]). We considered $D = d_j$ and we fixed the maximum number of enemies to a very high value (10^6), letting the algorithm almost free to find all the active neighbors belonging to a different class. A first set of experiments was run on a subset of the listed datasets described in Figure 7.3 (top, left): each dataset has a partial class overlapping with the others and two completely new categories. Here for each source database we have randomly chosen 90/30/30 samples per class for training, validation and test. For the new target database we fixed a test set of 50 samples and considered an increasing number of available training samples $N = \{10, 20, 30, 40, 50\}$. Only for Caltech-101 we reduced the described sets respectively to 30/10/10 and we used 10 samples as test set, due to the smaller

number of available data per class. We repeated the experiments with 10 random splits of the data.

The performance of MUST is compared to four baselines, two corresponding to learning from scratch and two exploiting the shared knowledge with naïve transfer approaches:

scratch : we used the Identity as projection matrix (Euclidean metric);
scratch+metric : we learn a metric from the available new training data (with [179]);
P_s : the shared projection matrix P_s learned on multiple datasets is applied on the new one;
Average : private projection matrices P_j are learned separately (with [179]) on each database supposing no sharing, they are then averaged and the obtained mean matrix is applied on the new dataset.

We can in principle combine all the samples from the visual datasets. This simple solution, apart from suffering for an explosion in the number of data, requires an explicit alignment procedure to be sure that all the samples with the same label in different datasets are combined together to form a unique final class. Moreover, as pointed out by [169], the dataset bias is an inherent problem that cannot be solved by simply mixing the available samples. However, as a reference to the results that could be obtained in this setting, we ran metric learning [179] on the *Union* of all the training data. All the final classification are performed using K-Nearest Neighbor with $K = 15$ ($K = 8$ used only for the experiments with 10 training samples).

From the results in Figure 7.3 we can state that averaging over all the sources does not directly provide a good solution for the target problem. On the other hand, when only few training samples are available (10-20), by learning on them we get just slightly better performance with respect to using directly the general knowledge in P_s . However, when the number of samples increases, P_s is no longer enough by itself to solve the learning problem on the new task. Finally, inferring the specific private knowledge on the new dataset and combining it with the shared common sense with our MUST algorithm *always* improves the average classification performance. By looking closely at the results on each new dataset, MUST mostly improves but *never degrades* the performance in comparison to not utilizing the available sources (scratch+metric in the plot).

Figure 7.4 shows some examples of the images in the target set of the described experiments. The visual appearance of objects misclassified by P_s but correctly recognized by MUST, indicates that MUST captures the bias of the specific target dataset. In particular, when using as source the dataset Caltech-101 and MSRCORID, the obtained general knowledge helps if the depicted target object is in the center of the image and fully visible. Specific knowledge is instead necessary in case of objects with very small dimensions, cluttered scenes and partial views, typical of the PASCAL VOC 07 and AwA datasets.

We also performed a second set of experiments considering all the available classes in each dataset. We defined 10 splits randomly extracting 20/10 train/test samples from each class
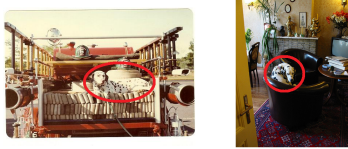
	P_s, MUST	MUST
PASCAL VOC 07		
AwA		

Figure 7.4 – Some examples of the images from the target test set of PASCAL VOC 07 class car and AwA class dalmatian. The second column of the table contains images classified correctly both when using only the shared projection matrix P_s and MUST. Instead the third column contains images that are misclassified with P_s but correctly recognized when the specific representation is combined to the common sense with MUST. Red circles have been drawn around the objects only for the sake of clarity to indicate their position: more than one instance can be present in each image.

of the target task dataset, 15 genuine neighbors and 100 enemies. The obtained results are reported in Table 7.1. Since the test set changes at each run, the standard deviations are only barely indicative. We evaluated the difference between scratch+metric and MUST separately for all the splits: the sign test [62] on the obtained output confirms that MUST significantly outperforms scratch+metric with $p \le 0.05$. There is only one exception for AwA, the animals are highly confused among each other and in this case it is probably necessary to increase the number of enemies in the method to reach significant results.

target	scratch+metric (%)	P_s(%)	Average (%)	MUST (%)
Caltech 101	65.69 ± 0.99	70.66 ± 1.87	75.35 ± 1.56	62.55 ± 1.08
Pascal VOC07	84.94 ± 3.14	84.50 ± 2.15	85.38 ± 3.42	80.66 ± 2.12
MSRCORID	45.80 ± 4.26	51.79 ± 2.73	52.59 ± 2.93	40.24 ± 3.11
AwA	94.02 ± 1.20	93.98 ± 0.84	94.24 ± 1.11	92.32 ± 1.18
CIFAR 100	90.91 ± 0.97	87.84 ± 1.06	92.76 ± 0.80	87.48 ± 0.78
overall	76.27	77.75	80.06	72.65

Table 7.1 – Error rate results obtained on the single-view experiments considering all the classes of each dataset: the lower the better. The last row reports the average result over all the datasets.

7.5.2 Heterogeneous setting

In the heterogeneous setting we considered different features for each dataset. We used bag of words SIFT features [1] for Caltech-101 ($d_j = 300$), Hue color histogram [2] for PASCAL VOC07 ($d_j = 300$), the already calculated Gist for MSRCORID ($d_j = 320$) and PHOG features [3] for AwA ($d_j = 252$). Finally we calculated PHOG for CIFAR ($d_j = 600$) but choosing different parameters with respect to the features used for AwA. We ran the experiments on the same data subset described above (Figure 7.3 (top, left)): we applied PCA separately on the multiple tasks to project all of them in the same dimensional space with $d = \{10, 50\}$ before running the metric learning process (we keep $D = d$) to define the shared knowledge. On the novel dataset, we can again use PCA and proceed with MUST to learn the specific metric, or we can activate the optimization for the projection matrix W . We consider the first approach as a reference baseline and compare with MUST-HET.

The number of genuine neighbors and enemies for these experiments are fixed to 3 to infer the general and specific knowledge on each task and to 5 for learning the projections from the heterogeneous features to the common space. These choices are done on the basis of two considerations. First, we want a good balance between computational cost and accuracy performance. Further, we aim to put a little more emphasis on retaining dataset-specific characteristics before inferring the shared knowledge in successive iterations. The last point lead us also to observe that for the heterogeneous problem, it is beneficial to have a dataset specific constant in (7.3) and (7.4) when enforcing the large difference between friends and enemies. Thus we substituted the value 1 in the second term of the loss function with the median of the squared pairwise distances in each dataset own feature space.

The results reported in Figure 7.5 show that on average MUST-HET is more suitable for the heterogeneous problem than the original MUST. Looking at the single target results, the advantage given by learning the projection matrix W_t is more evident the smaller is the dimension D . We also notice that MUST-HET performs always better (or at least equal) than MUST with one only exception when CIFAR 100 is used as target task with $D = 50$. We believe that in this particular case the combination of general and specific knowledge should be better weighted giving more importance to the common sense. This explanation is corroborated by the single-view CIFAR 100 results (Figure 7.3, bottom right) that show an initial abnormal increasing behavior for the scratch+metric baseline when the number of available training samples grows, while exploiting the common knowledge together with the specific one we get the best results.

---

1. available from http://www.vision.ee.ethz.ch/~pgehler/projects/iccv09/
2. DenseHueV3H1 available from http://lear.inrialpes.fr/people/guillaumin/data.php
3. available from http://attributes.kyb.tuebingen.mpg.de/
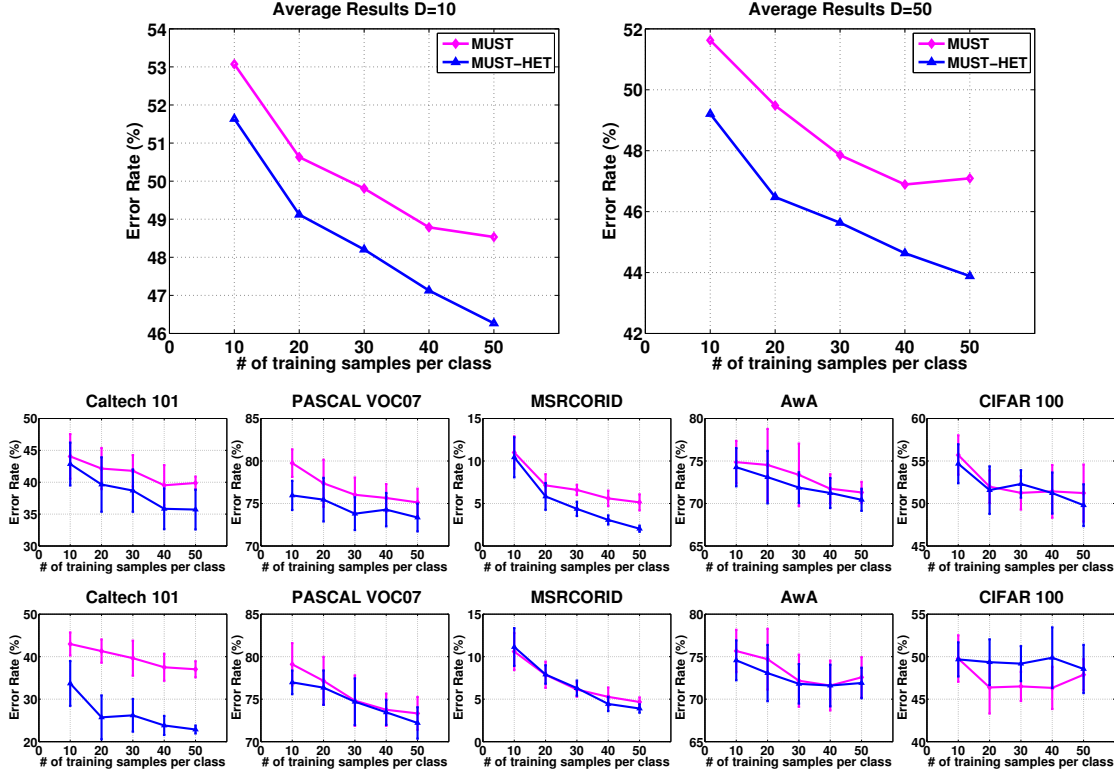
Figure 7.5 – Top: average error rate results on the five datasets considering the projection of all the different features to a space of dimension $D = 10$ (left) and $D = 50$ (right). Middle: separate results on each target datasets over 10 repetitions for $D = 10$. Bottom: separate results on each target datasets over 10 repetitions for $D = 50$. All the reported error rates for MUST-HET correspond to the best results obtained over the multiple iterations of the alternating optimization process.

7.6 Conclusion and Discussion

Exploiting existing resources in terms of pre-collected visual datasets when learning on a new multiclass problem should be a must. This asks for learning algorithms able to overcome the dataset bias problem and at the same time suggests an alternative solution to the need of building a unique large scale dataset.

We presented here our MUST algorithm: it decomposes each datasets into two orthogonal subspaces, one specific and one shared between all of them. Then the common information is transferred to help on a new task. On average, MUST *always* demonstrates cross-dataset generalization, assessed via a one-dataset-out strategy.

The proposed idea has been fully analyzed in the metric learning framework by using KNN. To assess its generality, it would be interesting to cast it into state-of-the-art kernel learning methods. In this respect we can make some preliminary considerations.

The first is about the comparison of MUST with the well known multi-task SVM method

presented in [54] and a very recent approach for the dataset bias problem proposed in [80]. Both the strategies consider the combination of a specific and a common discriminative model across several tasks: in [54] the common model is only used to share information while in [80] it is also constrained to perform well on any task on its own. However, being based on SVM, they both present a direct correspondence between one class and one model which makes them intrinsically limited to binary problems. Any multiclass extension would need tasks with identical sets of classes or would anyway require side-information about the label correspondence. By relying on metric learning, MUST overcomes the constraints of the SVM multiclass solutions, taking at the same time advantage of the max-margin framework by exploiting a formulation similar to that proposed in [179].

As second remark, we notice that the final outputs of MUST are a generic and a specific metric that can be used in a second stage for any method that requires distance computations between the samples, including kernel learning approaches. However it must be noticed that the first and the second phase in such a solution would have different goals: optimizing the distances among samples for KNN in the first case, and optimizing the margin for linear classification in the second. This drawback could be possibly overcome by establishing a theoretical link between the learned metric and its performance in classification by following the idea in [13].

Finally we presented MUST restricting the formulation to linear projection functions. However, the full method may be kernelized by building on the solutions already presented in [179] and [109].

A general comment is also necessary about the difference of MUST and our MKTL method presented in chapter 5. Both can be defined as feature transfer methods and are able to exploit heterogeneous sources of information. However the form of the sources is different. For MKTL the prior knowledge actually consists in black box experts declaring their opinion on any new sample. In this sense, the target learning process relies on the confidence of each expert, independently from how its knowledge was built and considering all the experts separately. MUST instead, exploits the consensus of multiple experts in the definition of the shared source knowledge and store details about their different opinions in the specific metric, by relying directly on the available samples. An experimental comparison on the presented homogeneous setup showed a rough equivalence of the results produced by MKTL and MUST, but a more detailed analysis is necessary to evaluate their relative positive and negative aspects.

To conclude, we stress that the aim of the work described in this chapter was mainly to present an efficient idea to attack the dataset bias problem and allow cross-database generalization. We didn't target the next state of the art accuracy on any of the considered databases, but rather we wanted to show that, in spite of the bias afflicting each of them, they do all carry a useful knowledge which is learnable and exploitable. Besides showing encouraging results, we have clearly only touched the surface of possibilities to be explored.

8 Conclusion and Perspective

This chapter summarizes the main results and contributions presented in this thesis, discusses the open issues and sketches possible future directions of research.

8.1 Summary

The ability to transfer information from one context to another that shares similar characteristics is a hallmark of human intelligence. We rapidly learn many kind of regularities and we recognize them when facing new problems. As a consequence we are able to re-use relevant knowledge from previous learning experience and make reliable inductive inferences even after a brief exposure to a single example. This indicates that we *learn to learn*, progressively finding out how to generalize across many learning tasks.

By analyzing the current state-of-the-art categorization methods, it is clear that many machine learning approaches based on a large amount of samples reach impressive results on difficult datasets [53, 128]. However, most of them traditionally address each learning task in isolation and provide very few guarantees when only a small amount of training samples is available, or more in general, if there is a mismatch between the training and the testing distribution [169, 129].

The purpose of this thesis has been to start closing the gap between human and artificial intelligence systems by developing principled methods able to exploit prior knowledge and boost the learning process. Our main contribution consists in defining adaptive methods that determine automatically on which known information to rely (from where to transfer), the degree of adaptation (how much to transfer) and if it is worth transferring something at all. Thanks to these properties all the resulting algorithms autonomously evaluate the relation between old and new knowledge and show a clear advantage in performance with respect to learning from scratch.

We started proposing a transfer learning method for binary classification problems based on Least Square Support Vector Machine (chapter 3). Prior knowledge is represented in the

form of pre-trained models: for any new target problem we learn a model close to the linear combination of the sources. The weights assigned to each prior model is found by solving a convex optimization problem which guarantees to have the minimal leave-one-out error on the training set of the target task. Extensive experiments on visual object categories were run to analyze the properties of our KT method and show its effectiveness in different learning conditions.

The same approach can be extended to multiclass domain adaptation problems (chapter 4). We considered different variants of the prior knowledge weighting technique and we performed experiments on classification and regression tasks that indicate their strength and weaknesses.

Exploiting source information in terms of pre-trained models means having access to the source learning process and it constrains to use the same approach on the target problem. To overcome these limits and be able to leverage over existing knowledge regardless of how it was defined, we proposed a feature transfer learning approach (chapter 5). Instead of relying on the prior learning models we just consider their output on the new target samples and use them as extra descriptors. By casting the problem in the multi-kernel learning framework we obtained the MKTL method which performs as well as the original model transfer approach on binary problems and is able of principled multiclass transfer learning.

The long term goal of this work is to enable autonomous systems to learn continuously from experience in an open-ended fashion (life long learning). To this end we addressed two issues.

First of all the existing knowledge must be continuously updated in time whenever new samples arrive. We showed that previous experience can be used to initialize an online learning process (chapter 6). Prior knowledge indicates which part of the original space the correct solution (the best in term of generalization capacity) should be sought and the relevance of each source information is progressively adapted with a computationally cheap solution. The obtained learning technique TROL has a theoretical support in terms of mistake bound and shows promising experimental results.

A second problem is that of finding reliable sources of information for an artificial intelligent agent, avoiding expensive human supervision. In the last years many visual data resources have been collected for different purposes and publicly released through the world wide web. Despite this constitutes a huge, useful and free knowledge repository, an artificial learner would not be directly able to use it to solve a new situated classification task. Indeed recent studies have demonstrated that the cross-dataset domain mismatch prevent any real advantage in the learning performance [169, 129]. We propose the algorithm MUST to overcome this problem (chapter 7): the great variety of available information covers different aspects of the real world and it is always possible to decompose it into a specific and a general part. By transferring only the last one to a new learning problem we get a clear advantage in the cross-database performance.

In conclusion our work demonstrated that properly exploiting pre-existing knowledge brings a real benefit to different aspects of the learning process. Principled techniques enabling knowledge transfer represent a progress towards making machine learning as efficient as human learning.

8.2 Open Issues

All the proposed algorithmic solutions for transfer learning and domain adaptation have been presented together with an analysis of their properties, discussing which is the best setting to apply them and evaluating their limits. We briefly describe in the following a few aspects of this work that might be somehow improved and remain relevant for future work.

Regarding the KT algorithm (chapter 3), a possible computational benefit could be obtained by imposing [191, 145], or trying to learn, a structure among the source knowledge set. If a defined source hierarchy is available, the target task can evaluate in a first step how much each parent node is relevant before passing to the leaf nodes. This would cut off the less relevant part of the hierarchy and let the target problem rely only on the most useful source knowledge set. Moreover, in our classification experiments we used only features extracted from the whole images and the semantic similarity among the source and the target task can be due to the context around the object as well as to the specific object itself. Combining segmentation [86, 81] and classification could restrict the transfer process to the object of interest and can give a relevant learning advantage in case of object detection problems. Finally, the weight assigned to each source task changes in time when the number of training samples increases. The stability of the weight vector can give some insight about how different the target task is with respect to the previous knowledge. A first analysis of this aspect has been proposed in [162], but a thorough evaluation could led to the proper definition of a rareness measure.

Regarding the MKTL algorithm (chapter 5), it would be interesting to analyze weights assigned to each source in transferring to see if for this method there is a correspondence between the evaluated task relatedness and the semantic similarity, as seen for the KT approach. Moreover, in our experiments the considered number of classes in the target set is limited to a maximum of 10 (20 considering the possibility of an overlapping among the source and target label set). The behavior of MKTL in a larger scale setting (10^2 classes) is a further and valuable direction to investigate. We have seen that MKTL can be directly applied to domain adaptation problems (section 5.6.2). In this setting, the known relation between the source and the target task label set could be used to reduce the number of considered feature mapping functions and corresponding kernels. An experimental analysis on this point would be worthwhile.

Regarding the TROL algorithm (chapter 6), although we have seen its performance on object category detection tasks, there is still space for an extensive evaluation on different classification conditions. A possible case is that of target concept drift. In this setting the statistical properties of the target samples, change over time in unforeseen ways and it is necessary to progressively follow it. Moreover TROL relies on the label of each incoming training sample,

but it would be interesting to consider a selective sampling solution [121]. In this condition, the algorithm would evaluate autonomously if it wants to receive or not the true label, possibly reducing the number of explicit training annotations.

Finally, to fully evaluate the strength of the MUST algorithm (chapter 7), it would be interesting to cast it in the kernel learning framework. This would allow an immediate comparison with MKTL and a direct benchmark with other existing algorithms can be obtained by reducing the classification problem to the standard domain adaptation setting.

Considering a more general vision over all the tackled problems, it is possible to identify two major challenges related to the dynamism of the learning process, and to some extent connected to each other: how to start and stop transferring and how to distinguish a new class from what is already known.

In all our work we supposed the existence of some difference between the source problems faced in the past and the target problem to be tackled in the future. Instead, standard machine learning methods consider only the case of uniform sources and targets. These actually represent two extreme cases in learning. An intelligent system should be able to modulate among them, it should recognize autonomously if an input is really new or if it can be classified as something already known. At the same time, it should exploit prior related knowledge till when enough information on the new concept is available and then stop the transfer process. Quantifying the necessary information in these terms is an open problem and asks for a proper integration of novelty detection and transfer learning.

Moreover, knowledge transfer leverages on prior experience exploiting at the maximum the *similarity* with the new target task. However it is important to underline that if a task is new, it also presents some *distinctive* aspect with respect to what is already known. Both the positive and negative components in analogical reasoning are relevant to define a concept. By relying only on the similarities we might end up with an over-generalizing learning system. Although obtaining optimal performance in learning something new when it is considered in isolation, such a system would be completely unable to distinguish it from the past knowledge. This makes impossible to properly memorize the new concept and use it as source of information in the future. An effective learning system should not only be able to exploit previous experience but also to progressively enlarge its knowledge.

A An appendix

Appendix A. An appendix

Algorithm 1 MultiSourceTrAdaBoost [183]

Input: the source $D_{S_1},\dots,D_{S_J}$ and target D_T training samples, a base learning algorithm **Learner**, and the maximum number of iterations M.

Set $\alpha_S = \frac{1}{2}\ln\left(1+\sqrt{2\ln\frac{n_S}{M}}\right)$, where $n_S = \sum_j n_{S_j}$ and $n_{S_j} = |D_{S_j}|$, $n_T = |D_T|$.

Initialize the weight vector $(\boldsymbol{w}^{S_1},\dots,\boldsymbol{w}^{S_J},\boldsymbol{w}^T)$ to the desired distribution, with $\boldsymbol{w}^{S_j} = (w_1^{S_j},\dots,w_{n_{S_j}}^{S_j})$ and $\boldsymbol{w}^T = (w_1^T,\dots,w_{n_T}^T)$

   for $m = 1,\dots,M$ **do**

      Empty the set of candidate weak classifiers, $\mathscr{F} \leftarrow \emptyset$

      Normalize to 1 the weight vector $(\boldsymbol{w}^{S_1},\dots,\boldsymbol{w}^{S_J},\boldsymbol{w}^T)$

      for $j = 1,\dots,J$ **do**

         Call **Learner**, providing it the combined training set $D_{S_j} \cup D_T$ with the distribution given by the weights $(\boldsymbol{w}^{S_j},\boldsymbol{w}^T)$.

         Get back an hypothesis $h_m^j : \mathscr{X} \to \mathscr{Y}$.

         Calculate the error on the target training samples D_T : $\quad \epsilon_m^j = \sum_i \frac{w_i^T\,[y_i^T \neq h_m^j(\boldsymbol{x}_i^T)]}{\sum_k w_k^T}$

         $\mathscr{F} \leftarrow \mathscr{F} \cup (h_m^j,\epsilon_m^j)$

      end for

      Find the weak classifier $h_m : \mathscr{X} \to \mathscr{Y}$ such that $(h_m,\epsilon_m) = \operatorname{argmin}_{(h,\epsilon)\in\mathscr{F}} \epsilon$

      Set $\alpha_m = \frac{1}{2}\ln\frac{1-\epsilon_m}{\epsilon_m}$, where $\epsilon_m < 0.5$

      Update the weight vector

      $w_i^{S_j} \leftarrow \quad w_i^{S_j}\exp\{-\alpha_S|h_m(\boldsymbol{x}_i^{S_j})-y_i^{S_j}|\}$

      $w_i^T \leftarrow \quad\;\; w_i^T\exp\{\alpha_m|h_m(\boldsymbol{x}_i^T)-y_i^T|\}$

   end for

Output: the target classifier function $\quad f_M(\boldsymbol{x}) = \operatorname{sign}\left(\sum_m \alpha_m h_m(\boldsymbol{x})\right)$

This algorithm reduces to TrAdaBoost as presented in [41] for a single source $J = 1$. For the experiments we chose the Learner algorithm following [183] and other technical details obtained by personal comunication from the authors of the same paper. In particular we considered linear SVM by using the SVMlight [76] implementation which allows for an automatic choice of the C parameter.

Algorithm 2 Phase I - TaskTrAdaBoost [183]

Input: the source $D_{S_1},\dots,D_{S_J}$, a base learning algorithm **Learner**, the maximum number of iterations M and the regularization threshold τ.

Empty the set of candidate weak classifiers, $\mathscr{H} \leftarrow \varnothing$

    for $j = 1,\dots,J$ **do**

        Initialize the weight vector $\boldsymbol{w}^{S_j} = (w_1^{S_j},\dots,w_{n_{S_j}}^{S_j})$ to the desired distribution

        for $m = 1,\dots,M$ **do**

            Normalize to 1 the weight vector $\boldsymbol{w}^{S_j}$

            Find the candidate weak classifier $h_m^j : \mathscr{X} \to \mathscr{Y}$ that minimizes the classification error over the set D_{S_j}, weighted according to $\boldsymbol{w}^{S_j}$

            Get back an hypothesis $h_m^j : \mathscr{X} \to \mathscr{Y}$.

            Compute the error $\epsilon_m^j = \sum_i w_i^{S_j}[y_i^{S_j} \neq h_m^j(\boldsymbol{x}_i^{S_j})]$

            $\alpha = \frac{1}{2}\ln\frac{1-\epsilon}{\epsilon}$, where $\epsilon < 0.5$

            if $\alpha > \tau$ **then**

                $\mathscr{H} \leftarrow \mathscr{H} \cup h_m^j$

            end if

            Update the weights $w_i^{S_j} \leftarrow w_i^{S_j} \exp\{-\alpha y_i^{S_j} h_m^j(\boldsymbol{x}_i^{S_j})\}$

        end for

    end for

Output: the set of candidate weak classifiers $\mathscr{H}$

Algorithm 3 Phase II - TaskTrAdaBoost [183]

Input: the target training data D_T, a base learning algorithm **Learner**, the maximum number of iterations M and the set of weak classifiers $\mathscr{H}$.

Initialize the weight vector $\boldsymbol{w}^T = (w_1^T,\dots,w_{n_T}^T)$ to the desired distribution

    for $m = 1,\dots,M$ **do**

        Normalize to 1 the weight vector $\boldsymbol{w}^T$

        Empty the current weak classifier set $\mathscr{F} \leftarrow \varnothing$

        for all $h = 1 \in \mathscr{H}$ **do**

            Compute the error h on D_T : $\epsilon = \sum_i w_i^T[y_i^T \neq h(\boldsymbol{x}_i^T)]$

            if $\epsilon > 0.5$ **then**

                $h \leftarrow -h$

                Update ϵ

            end if

            $\mathscr{F} \leftarrow \mathscr{F} \cup (h,\epsilon)$

        end for

        Find the weak classifier $h_m : \mathscr{X} \to \mathscr{Y}$ such that $(h_m,\epsilon_m) = \mathrm{argmin}_{(h,\epsilon)\in\mathscr{F}} \epsilon$

        $\mathscr{H} \leftarrow \mathscr{H}\backslash h_m$

        Set $\alpha_m = \frac{1}{2}\ln\frac{1-\epsilon_m}{\epsilon_m}$

        Update the weights $w_i^T \leftarrow w_i^T \exp-\alpha_m y_i^T h_m^j(\boldsymbol{x}_i^T)$

    end for

Output: the target classifier function $f_M(\boldsymbol{x}) = \mathrm{sign}\left(\sum_m \alpha_m h_m(\boldsymbol{x})\right)$

Algorithm 4 Projected Sub-gradient Descent Algorithm   [165]

Input: calculate $\boldsymbol{a}'$ and $\boldsymbol{a}''_j$ according to Proposition 1 .
Initialize $\beta \leftarrow 0$ and $t \leftarrow 1$.

  repeat

$$\tilde{y}_i \leftarrow y_i - \frac{\boldsymbol{a}'_i}{P_{ii}} + \sum_{j=1}^{J} \beta_j \frac{\boldsymbol{a}''_{ij}}{P_{ii}} \quad \forall \quad i = 1, \ldots, N$$

$$d_i \leftarrow \mathbf{1}\{y_i \tilde{y}_i > 0\}, \quad \forall \quad i = 1, \ldots, N$$

$$\beta_j \leftarrow \beta_j - \frac{1}{\sqrt{t}} \sum_{i=1}^{N} d_i y_i \frac{\boldsymbol{a}''_{ij}}{P_{ii}}, \quad \forall \quad j = 1, \ldots, J$$

    if $\|\boldsymbol{\beta}\|_2 > 1$ **then**

$$\boldsymbol{\beta} \leftarrow \boldsymbol{\beta} / \|\boldsymbol{\beta}\|_2$$

    end if

$$\beta_j \leftarrow \max(\beta_j, 0), \quad \forall \quad j = 1, \ldots, J$$

$$t \leftarrow t + 1$$

  until convergence

Output: $\boldsymbol{\beta}$

Algorithm 5 Projected Sub-gradient Descent Algorithm – Multiclass   [166]

Input: calculate $\boldsymbol{A}'$ according to (4.3) and $\boldsymbol{A}''^j$ according to (4.4) .
Initialize $\boldsymbol{\beta} = [\beta_1 \ldots \beta_J] \leftarrow \boldsymbol{0}$ and $t \leftarrow 1$.

  repeat

$$\tilde{\boldsymbol{Y}}_i \leftarrow \boldsymbol{Y}_i - \frac{\boldsymbol{A}'_i}{P_{ii}} + \sum_{j=1}^{J} \beta_j \frac{\boldsymbol{A}''^j_i}{P_{ii}} \quad \forall \quad i = 1, \ldots, N$$

$$g_i^* \leftarrow \operatorname{argmax}_{g \neq y_i}\{\tilde{Y}_{gi}\}, \quad \forall \quad i = 1, \ldots, N$$

$$d_i \leftarrow \mathbf{1}\{1 - \tilde{Y}_{y_i i} + \tilde{Y}_{g_i^* i} > 0\}, \quad \forall \quad i = 1, \ldots, N$$

$$\beta_j \leftarrow \beta_j - \frac{1}{\sqrt{t}} \sum_{i=1}^{N} d_i \frac{(A''^j_{g_i^* i} - A''^j_{y_i i})}{P_{ii}}, \quad \forall \quad j = 1, \ldots, J$$

    if $\|\boldsymbol{\beta}\|_2 > 1$ **then**

$$\boldsymbol{\beta} \leftarrow \boldsymbol{\beta} / \|\boldsymbol{\beta}\|_2$$

    end if

$$\beta^j \leftarrow \max(\beta^j, 0), \quad \forall \quad j = 1, \ldots, J$$

$$t \leftarrow t + 1$$

  until convergence

Output: $\boldsymbol{\beta}$

Here $\mathbf{1}\{\cdot\}$ denotes the indicator function.

Algorithm 6 MUST [168]

Input: J source datasets $\mathscr{D}_j = \{(\boldsymbol{x}_1^j, y_1^j), \ldots, (\boldsymbol{x}_{N_j}^j, y_{N_j}^j)\} \subset \mathbb{R}^d \times \mathscr{Y}^j$ and a target dataset $\mathscr{D}_t = \{(\boldsymbol{x}_1^t, y_1^t), \ldots, (\boldsymbol{x}_{N_t}^t, y_{N_t}^t)\} \subset \mathbb{R}^d \times \mathscr{Y}^t$

Solve the following optimization problem in for shared P_s and private $P_{1,\ldots,J}$

$$\min \quad \eta\Omega(P_j) + \gamma\Omega(P_s) + \sum_{j=1}^{J} \sum_{i \sim k} d_j^2(\boldsymbol{x}_i^j, \boldsymbol{x}_k^j) + \sum_{\substack{i \sim k \\ i \neq l}} \max\left\{0, 1 + d_j^2(\boldsymbol{x}_i^j, \boldsymbol{x}_k^j) - d_j^2(\boldsymbol{x}_i^j, \boldsymbol{x}_l^j)\right\}$$

(A.1)

subject to $\quad P_s^\top P_j = 0 \quad$ for all $\quad j = 1, \ldots, J$,

where $\quad d_j^2(\boldsymbol{x}_i^j, \boldsymbol{x}_k^j) := \|(P_j + P_s)(\boldsymbol{x}_i^j - \boldsymbol{x}_k^j)\|^2$.

Transfer the common sense as captured by P_s to the target dataset and find P_t by solving

$$\min \quad \eta\Omega(P_t) + \sum_{i \sim k} d_t^2(x_i^t, x_k^t) + \sum_{\substack{i \sim k \\ i \neq l}} \max\{0, 1 + d_t^2(x_i^t, x_k^t) - d_t^2(x_i^t, x_l^t)\}$$

(A.2)

subject to $\quad P_s^\top P_t = 0$,

where $\quad d_t^2(\boldsymbol{x}_i^t, \boldsymbol{x}_k^t) := \|(P_t + P_s)(\boldsymbol{x}_i^t - \boldsymbol{x}_k^t)\|^2$.

Output: P_s, $P_{1,\ldots,J}$, P_t

Algorithm 7 MUST-HET [168]

Input: J source datasets $\mathscr{D}_j = \{(\boldsymbol{x}_1^j, y_1^j), \ldots, (\boldsymbol{x}_{N_j}^j, y_{N_j}^j)\} \subset \mathbb{R}^d \times \mathscr{Y}^j$ and a target dataset $\mathscr{D}_t = \{(\boldsymbol{x}_1^t, y_1^t), \ldots, (\boldsymbol{x}_{N_t}^t, y_{N_t}^t)\} \subset \mathbb{R}^d \times \mathscr{Y}^t$. The number of alternations A.

Define $d_j^2(\boldsymbol{x}_i^j, \boldsymbol{x}_k^j) = (W_j(\boldsymbol{x}_i^j - \boldsymbol{x}_k^j))^\top (P_j^\top P_j + P_s^\top P_s)(W_j(\boldsymbol{x}_i^j - \boldsymbol{x}_k^j))$

Initialize $W_j^{d_j} \leftarrow W_j^{\text{PCA}}$ for all $j = 1, \ldots, J$

$a = 0$

repeat

    $a \leftarrow a + 1$

    Solve the optimization problem in (A.1) for P_s and $P_{1,\ldots,J}$ and fixed W_j

    Solve the optimization problem in (A.1) for W_j and fixed P_s, $P_{1,\ldots,J}$

until $a = A$

Initialize $W_t^{d_t} = W_t^{\text{PCA}}$

Transfer the common sense as captured by P_s to the target and find P_t with

$a = 0$

repeat

    $a \leftarrow a + 1$

    Solve the optimization problem in (A.2) for P_t and fixed W_t

    Solve the optimization problem in (A.2) for W_t and fixed P_t

until $a = A$

Output: P_s, $P_{1,\ldots,J}$, P_t, $W_j^{d_j}$ and $W_t^{d_t} \in \mathbb{R}^{d_t \times d}$

Bibliography

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

[2] G. M. Allenby and P. E. Rossi. Marketing models of consumer heterogeneity. *Journal of Econometrics*, 89(1-2):57–78, 1998.

[3] R. K. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853, 2005.

[4] A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008.

[5] A. Arnold, R. Nallapati, and W. W. Cohen. A comparative study of methods for transductive transfer learning. In *ICDM Workshop on Mining and Management of Biological Data*, 2007.

[6] D. J. Atkins, D. C. Y. Heard, and W. H. Donovan. Epidemiologic overview of individuals with upper-limb loss and their reported research priorities. *Journal Of Prosthetics And Orthotics*, 8(1):2–11, 1996.

[7] M. Atzori, A. Gijsberts, S. Heynen, A.-G. Mittaz Hager, O. Deriaz, P. Van der Smagt, C. Castellini, B. Caputo, and H. Müller. Building the NinaPro database: a resource for the biorobotics community. In *IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, 2012.

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

[9] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan. Multiple kernel learning, conic duality, and the SMO algorithm. In *Proceedings of the twenty-first international conference on Machine learning (ICML)*, 2004.

[10] M.T. Bahadori, Y. Liu, and D. Zhang. Learning with minimum supervision: A general framework for transductive transfer learning. In *International Conference on Data Mining (ICDM)*, 2011.

[11] J. Baxter. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12:149–198, 2000.

[12] N. Bel, C. H. A. Koster, and M. Villegas. Cross-lingual text categorization. In *European Conference on Research and Advanced Technology for Digital Libraries (ECDL)*, 2003.

[13] A. Bellet, A. Habrard, and M. Sebban. Similarity Learning for Provably Accurate Sparse Linear Classification. In *International Conference on Machine Learning (ICML)*, 2012.

[14] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira. Analysis of representations for domain adaptation. In *Advances in Neural Information Processing Systems (NIPS)*, 2007.

[15] S. Bickel, M. Brückner, and T. Scheffer. Discriminative learning under covariate shift. *Journal of Machine Learning Research*, 10:2137–2155, 2009.

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

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

[18] J. Blitzer, R. McDonald, and F. Pereira. Domain adaptation with structural correspondence learning. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2006.

[19] K. M. Borgwardt, A. Gretton, M. J. Rasch, H.P. Kriegel, B. Schölkopf, and A. J. Smola. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22(14):e49–e57, 2006.

[20] A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. In *ACM international conference on Image and video retrieval (CIVR)*, 2007.

[21] A. H. Bottomley. Myoelectric control of powered prostheses. *Journal of Bone & Joint Surgery, British*, 47-B(3):411—415, 1965.

[22] O. Bousquet and A. Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, 2002.

[23] L. Bruzzone and M. Marconcini. Domain adaptation problems: A DASVM classification technique and a circular validation strategy. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 32(5):770–787, 2010.

[24] P. H. Calais Guerra, A. Veloso, W. Jr. Meira, and V. Almeida. From bias to opinion: a transfer-learning approach to real-time sentiment analysis. In *International Conference on Knowledge discovery and data mining (ACM SIGKDD)*, 2011.

[25] R. Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.

[26] C. Castellini, A. E. Fiorilla, and G. Sandini. Multi-subject / daily-life activity EMG-based control of mechanical hands. *Journal of Neuroengineering and Rehabilitation*, 6(41), 2009.

[27] C. Castellini, E. Gruppioni, A. Davalli, and G. Sandini. Fine detection of grasp force and posture by amputees via surface electromyography. *Journal of Physiology (Paris)*, 103(3-5):255–262, 2009.

[28] C. Castellini and P. van der Smagt. Surface EMG in advanced hand prosthetics. *Biological Cybernetics*, 100(1):35–47, 2009.

[29] G. C. Cawley and N. L. C. Talbot. Preventing over-fitting during model selection via bayesian regularisation of the hyper-parameters. *Journal of Machine Learning Research*, 8:841–861, May 2007.

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

[31] C. Chelba and A. Acero. Adaptation of maximum entropy capitalizer: Little data can help a lot. *Computer Speech & Language*, 20(4):382–399, 2006.

[32] D. S. Childress. A myoelectric three-state controller using rate sensitivity. In *International Conference on Medical and Biological Engineering (ICMBE)*, 1969.

[33] C. Cortes, M. Mohri, M. Riley, and A. Rostamizadeh. Sample selection bias correction theory. In *International conference on Algorithmic Learning Theory (ALT)*, 2008.

[34] T. F. Cox and M. A. A. Cox. *Multidimensional Scaling.* Chapman and Hall, 2001.

[35] K. Crammer, O. Dekel, S. Shalev-Shwartz, and Y. Singer. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7:2006, 2003.

[36] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292, March 2002.

[37] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines (and Other Kernel-Based Learning Methods).* Cambridge University Press, 2000.

[38] W. Dai, Y. Chen, G.R. Xue, Q. Yang, and Y. Yu. Translated learning: Transfer learning across different feature spaces. In *Advances in Neural Information Processing Systems (NIPS)*, 2008.

[39] W. Dai, O. Jin, G.R. Xue, Q. Yang, and Y. Yu. Eigentransfer: a unified framework for transfer learning. In *International Conference on Machine Learning (ICML)*, 2009.

[40] W. Dai, G.R. Xue, Q. Yang, and Y. Yu. Transferring naive bayes classifiers for text classification. In *Association for the Advancement of Artificial Intelligence Conference (AAAI)*, 2007.

[41] W. Dai, Q. Yang, G.R. Xue, and Y. Yu. Boosting for transfer learning. In *International Conference on Machine Learning (ICML)*, 2007.

[42] W. Dai, Q. Yang, G.R. Xue, and Y. Yu. Self-taught clustering. In *International Conference on Machine learning (ICML)*, 2008.

[43] H. Daumé III. Frustratingly easy domain adaptation. In *Association for Computational Linguistics Conference (ACL)*, 2007.

[44] H. Daumé III. Bayesian multitask learning with latent hierarchies. In *Conference on Uncertainty in Artificial Intelligence (UAI-09)*, 2009.

[45] J. Davis and P. Domingos. Deep transfer via second-order markov logic. In *International Conference on Machine Learning (ICML)*, 2009.

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

Bibliography

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

[48] T. Deselaers and T. Deserno. Medical image annotation in ImageCLEF 2008. In *working notes CLEF*, 2008.

[49] P. S. Dhillon and L. H. Ungar. Transfer learning, feature selection and word sense disambguation. In *ACL-IJCNLP Conference Short Papers*, 2009.

[50] L. Duan, I. W. Tsang, and D. Xu. Domain transfer multiple kernel learning. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 34, 2012.

[51] L. Duan, D. Xu, and I. W.-H. Tsang. Domain adaptation from multiple sources: A domain-dependent regularization approach. *IEEE Transactions on Neural Networks and Learning Systems*, 23(3):504–518, 2012.

[52] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the l1-ball for learning in high dimensions. In *International Conference on Machine Learning (ICML)*, 2008.

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

[54] T. Evgeniou, C. A. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6:615–637, 2005.

[55] T. Evgeniou and M. Pontil. Regularized multi–task learning. In *International Conference on Knowledge discovery and data mining (ACM SIGKDD)*, 2004.

[56] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *Computer Vision and Pattern Recognition (CVPR)*, 2009.

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

[58] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding*, 106(1):59–70, 2007.

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

[60] M. Fink. Object classification from a single example utilizing class relevance metrics. In *Advances in Neural Information Processing Systems (NIPS)*, pages 449–456, 2004.

[61] P. Gehler and S. Nowozin. On feature combination for multiclass object classification. In *International Conference on Computer Vision (ICCV)*, 2009.

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

[63] J. Goldberger, S.T. Roweis, G. E. Hinton, and R. Salakhutdinov. Neighbourhood components analysis. In *Advances in Neural Information Processing Systems (NIPS)*, 2004.

[64] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Computer Vision and Pattern Recognition Conference (CVPR)*, 2012.

[65] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *International Conference on Computer Vision (ICCV)*, 2011.

[66] W. H. Greene. *Econometric Analysis.* 2003.

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

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

[69] D. Hush, P. Kelly, C. Scovel, and I. Steinwart. QP algorithms with guaranteed accuracy and run time for support vector machines. *Journal of Machine Learning Research*, 2006.

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

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

[72] Y. Jia, M. Salzmann, and T. Darrell. Factorized latent spaces with structured sparsity. In *Advances in Neural Information Processing Systems (NIPS)*, 2010.

[73] J. Jiang. A Literature Survey on Domain Adaptation of Statistical Classifiers. http://sifaka.cs.uiuc.edu/jiang4/domain_adaptation/survey/.

[74] J. Jiang and C. Zhai. Instance weighting for domain adaptation in NLP. In *Annual Meeting of the Association of Computational Linguistics (ACL)*, 2007.

[75] W. Jiang, E. Zavesky, S.F. Chang, and A. Loui. Cross-Domain Learning Methods for High-Level Visual Concept Classification. In *IEEE International Conference on Image Processing (ICIP)*, 2008.

[76] T. Joachims. Making large-scale SVM learning practical. In *Advances in Kernel Methods - Support Vector Learning.* MIT Press, 1999.

[77] T. Kanamori, S. Hido, and M. Sugiyama. A least-squares approach to direct importance estimation. *Journal of Machine Learning Research*, 10, 2009.

[78] Z. Kang, K. Grauman, and F. Sha. Learning with whom to share in multi-task feature learning. In *International Conference on Machine Learning (ICML)*, 2011.

[79] F. Peterson Kendall, E. Kendall McCreary, P. Geise Provance, M. McIntyre Rodgers, and W.A. Romani. *Muscles: Testing and Function, with Posture and Pain.* Lippincott Williams & Wilkins, 530 Walnut St.Philadelphia, PA 19106-3621, 2005.

[80] A. Khosla, T. Zhou, T. Malisiewicz, A. Efros, and A. Torralba. Undoing the damage of dataset bias. In *European Conference on Computer Vision (ECCV)*, 2012.

[81] Jaechul Kim and Kristen Grauman. Shape Sharing for Object Segmentation. In *European Conference on Computer Vision (ECCV)*, 2012.

[82] A. Kleiner, A. Rahimi, and M. I. Jordan. Random conic pursuit for semidefinite programming. In *Advances in Neural Information Processing Systems (NIPS)*, 2010.

[83] M. Kloft, U. Brefeld, S. Sonnenburg, P. Laskov, K.-R. Müller, and A. Zien. Efficient and accurate L_p-norm multiple kernel learning. In *Advances in Neural Information Processing Systems (NIPS)*. 2009.

[84] A. Kovashka, S. Vijayanarasimhan, and K. Grauman. Actively selecting annotations among objects and attributes. In *International Conference on Computer Vision (ICCV)*, 2011.

[85] A. Krizhevsky. Learning multiple layers of features from tiny images. Technical Report MSc thesis, University of Toronto, USA, 2007.

[86] Daniel Kuettel, Matthieu Guillaumin, and Vittorio Ferrari. Segmentation propagation in imagenet. In *European Conference on Computer Vision (ECCV)*, 2012.

[87] B. Kulis, K. Saenko, and T. Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *Computer Vision and Pattern Recognition Conference (CVPR)*, 2011.

[88] I. Kuzborskij, A. Gijsberts, and B. Caputo. On the challenge of classifying 52 hand movements from surface electromyography. In *International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2012.

[89] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between class attribute transfer. In *Computer Vision and Pattern Recognition Conference (CVPR)*, 2009.

[90] H. Larochelle, D. Erhan, and Y. Bengio. Zero-data learning of new tasks. In *AAAI Conference on Artificial Intelligence*, 2008.

[91] A. Lazaric, M. Restelli, and A. Bonarini. Transfer of samples in batch reinforcement learning. In *International Conference on Machine learning (ICML)*, 2008.

[92] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition Conference (CVPR)*, 2006.

[93] G. Leen. *Context assisted information extraction*. PhD thesis, University of the West of Scotland, 2008.

[94] T.M. Lehmann, H.Schubert, D. Keysers, M. Kohnen, and B.B. Wein. The IRMA code for unique classification of medical images. In *International Society for Optical Engineering (SPIE)*, 2003.

[95] A. S. Lewis and M. L. Overton. Nonsmooth optimization via quasi-newton methods. *Mathematical Programming*, 2012.

[96] 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 *Advances in Neural Information Processing Systems (NIPS)*, 2010.

[97] X. Li and J. Bilmes. A bayesian divergence prior for classifier adaptation. In *Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2007.

[98] J. J. Lim, R. Salakhutdinov, and A. Torralba. Transfer learning by borrowing examples for multiclass object detection. In *Advances in Neural Information Processing Systems (NIPS)*, 2011.

[99] J. Liu, S. Ji, and J. Ye. Multi-task feature learning via efficient $l_{2,1}$-norm minimization. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2009.

[100] J. Liu, M. Shah, B. Kuipers, and S. Savarese. Cross-view action recognition via view knowledge transfer. In *Computer Vision and Pattern Recognition Conference (CVPR)*, 2011.

[101] N. Loeff and A. Farhadi. Scene discovery by matrix factorization. In *European Conference on Computer Vision (ECCV)*, 2008.

[102] T. Lorrain, N. Jiang, and D. Farina. Influence of the training set on the accuracy of surface EMG classification in dynamic contractions for the control of multifunction prostheses. *J Neuroeng Rehabil*, 8:25, 2011.

[103] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[104] J. Luo, T. Tommasi, and B. Caputo. Multiclass transfer learning from unconstrained priors. In *International Conference on Computer Vision (ICCV)*, 2011.

[105] M. M. Mahmud and S. R. Ray. Transfer learning using kolmogorov complexity: Basic theory and empirical evaluations. In *Advances in Neural Information Processing Systems (NIPS)*, 2007.

[106] Y. Mansour, M. Mohri, and A. Rostamizadeh. Domain adaptation with multiple sources. In *Advances in Neural Information Processing Systems (NIPS)*, 2008.

[107] Y. Mansour, M. Mohri, and A. Rostamizadeh. Domain adaptation: Learning bounds and algorithms. In *Conference on Learning Theory (COLT)*, 2009.

[108] M. Markou and S. Singh. Novelty detection: a review—part 1: statistical approaches. *Signal Process.*, 83(12):2481–2497, December 2003.

[109] B. McFee, C. Galleguillos, and G. Lanckriet. Contextual object localization with multiple kernel nearest neighbor. *Trans. Img. Proc.*, 20(2):570–585, February 2011.

[110] R. Merletti, M. Aventaggiato, A. Botter, A. Holobar, H. Marateb, and T.M.M. Vieira. Advances in surface EMG: Recent progress in detection and processing techniques. *Critical reviews in biomedical engineering*, 38(4):305—345, 2011.

[111] R. Merletti, A. Botter, C. Cescon, M.A. Minetto, and T.M.M. Vieira. Advances in surface EMG: Recent progress in clinical research applications. *Critical reviews in biomedical engineering*, 38(4):347—379, 2011.

[112] R. Merletti, A. Botter, A. Troiano, E. Merlo, and M.A. Minetto. Technology and instrumentation for detection and conditioning of the surface electromyographic signal: State of the art. *Clinical Biomechanics*, 24:122—134, 2009.

Bibliography

[113] S. Micera, J. Carpaneto, and Stanisa Raspopovic. Control of hand prostheses using peripheral information. *IEEE Reviews in Biomedical Engineering*, 3:48–68, October 2010.

[114] Microsoft. Microsoft Research Cambridge Object Recognition Image Database. http://research.microsoft.com/en-us/downloads/b94de342-60dc-45d0-830b-9f6eff91b301/default.aspx, 2005.

[115] T. Mitchell. *Machine Learning (Mcgraw-Hill International Edit)*. 1997.

[116] G. Obozinski, B. Taskar, and M. I. Jordan. Joint covariate selection and joint subspace selection for multiple classification problems. *Statistics and Computing*, 20(2):231–252, 2010.

[117] T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51–59, January 1996.

[118] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42:145–175, 2001.

[119] F. Orabona. *DOGMA: a MATLAB toolbox for Online Learning*, 2009. Software available at http://dogma.sourceforge.net.

[120] F. Orabona, L. Jie, and B. Caputo. Online-batch strongly convex multi kernel learning. In *Computer Vision and Pattern Recognition (CVPR)*, 2010.

[121] Francesco Orabona and Nicolò Cesa-Bianchi. Better algorithms for selective sampling. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, 2011.

[122] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, 2011.

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

[124] S. J. Pan, V. W. Zheng, Q. Yang, and D. Hao Hu. Transfer learning for wifi-based indoor localization. In *Workshop on Transfer Learning for Complex Task of AAAI Conference on Artificial Intelligence*, 2008.

[125] S. Parameswaran and K.Q. Weinberger. Large margin multi-task metric learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2010.

[126] P. Parker, K. Englehart, and B. Hudgins. Myoelectric signal processing for control of powered limb prostheses. *Journal of Electromyography and Kinesiology*, 16:541–548, 2006.

[127] B. Peerdeman, D. Boere, H. Witteveen, R. Huis in't Veld, H. Hermens, S. Stramigioli, H. Rietman, P. Veltink, and S Misra. Myoelectric forearm prostheses: State of the art from a user-centered perspective. *Journal of Rehabilitation Research & Development*, 48(6):719–738, 2011.

[128] F. Perronnin and J. Sánchez. Compressed fisher vectors for LSVRC. In *PASCAL VOC / ImageNet Workshop, International Conference on Computer Vision (ICCV)*, 2011.

[129] F. Perronnin, J. Sánchez, and Y. Liu. Large-scale image categorization with explicit data embedding. In *Computer Vision and Pattern Recognition (CVPR)*, 2010.

[130] L. Philipson, D. S. Childress, and J. Strysik. Digital approaches to myoelectric state control of prostheses. *Bulletin of Prosthetics Research*, 18(2):3—11, 1981.

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

[132] N. Quadrianto, A. J. Smola, T. S. Caetano, S. V. N. Vishwanathan, and J. Petterson. Multitask learning without label correspondences. In *Advances in Neural Information Processing Systems (NIPS)*, 2010.

[133] A. Quattoni, M. Collins, and T. Darrell. Transfer learning for image classification with sparse prototype representations. In *Computer Vision and Pattern Recognition Conference (CVPR)*, 2008.

[134] R. Raina, A. Battle, Honglak Lee, B. Packer, and A. Y. Ng. Self-taught learning: Transfer learning from unlabeled data. In *International Conference on Machine Learning (ICML)*, 2007.

[135] E. Rodner and J. Denzler. One-shot learning of object categories using dependent gaussian processes. In *Proceedings of the 32nd DAGM conference on Pattern recognition*, 2010.

[136] E. Rodner and J. Denzler. Learning with few examples for binary and multiclass classification using regularization of randomized trees. *Pattern Recogn. Lett.*, 32(2):244–251, 2011.

[137] M. Rohrbach, M. Stark, and B. Schiele. Evaluating knowledge transfer and zero-shot learning in a large-scale setting. In *Computer Vision and Pattern Recognition Conference (CVPR)*, 2011.

[138] M. Rohrbach, M. Stark, G. Szarvas, I. Gurevych, and B. Schiele. What helps where – and why? semantic relatedness for knowledge transfer. In *Computer Vision and Pattern Recognition Conference (CVPR)*, 2010.

[139] B. Romera-Paredes, A. Argyriou, N. Berthouze, and M. Pontil. Exploiting unrelated tasks in multi-task learning. *Journal of Machine Learning Research - Proceedings Track*, pages 951–959, 2012.

[140] 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.

[141] M. Rosenstein, Z. Marx, T. Dietterich, and L. P. Kaelbling. Transfer learning with an ensemble of background tasks. In *NIPS Workshop on Inductive Transfer*, 2005.

[142] U. Rückert and S. Kramer. Kernel-based inductive transfer. In *European conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD)*, 2008.

[143] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010.

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

[145] R. Salakhutdinov, A. Torralba, and J. Tenenbaum. Learning to Share Visual Appearance for Multiclass Object Detection. In *Computer Vision and Pattern Recognition Conference (CVPR)*, 2011.

[146] M. Salzmann, C.H. Ek, R. Urtasun, and T. Darrell. Factorized orthogonal latent spaces. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2009.

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

[148] C.-W. Seah, I. W.H. Tsang, and Y.S. Ong. Healing sample selection bias by source classifier selection. In *International Conference on Data Mining (ICDM)*, 2011.

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

[150] J. Simm, M. Sugiyama, and T. Kato. Computationally efficient multi-task learning with least-squares probabilistic classifiers. *IPSJ Transactions on Computer Vision and Applications*, 3:1–8, 2011.

[151] S. P. Singh. Transfer of learning by composing solutions of elemental sequential tasks. *Mach. Learn.*, 8(3-4):323–339, 1992.

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

[153] M. M. Stark and R. F. Riesenfeld. WordNet: An electronic lexical database. In *Eurographics Workshop on Rendering*. MIT Press, 1998.

[154] M. Sugiyama, S. Nakajima, H. Kashima, P. von Bünau, and M. Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *Advances in Neural Information Processing Systems (NIPS)*, 2007.

[155] Q. Sun, R. Chattopadhyay, S. Panchanathan, and J. Ye. A two-stage weighting framework for multi-source domain adaptation. In *Advances in Neural Information Processing Systems (NIPS)*, 2011.

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

[157] M. E. Taylor, G. Kuhlmann, and P. Stone. Accelerating search with transferred heuristics. In *ICAPS-07 workshop on AI Planning and Learning*, 2007.

[158] F. V. Tenore, A. Ramos, A. Fahmy, S. Acharya, R. Etienne-Cummings, and N. V. Thakor. Decoding of individuated finger movements using surface electromyography. *IEEE Trans. Biomed. Eng.*, 56(5):1427—1434, 2009.

[159] S. Thrun. Is learning the n-th thing any easier than learning the first? In *Advances in Neural Information Processing Systems (NIPS)*, 1996.

[160] S. Thrun. Learning to learn. chapter Learning To Learn: Introduction. Kluwer Academic Publishers, 1996.

[161] T. Tommasi and B. Caputo. The more you know, the less you learn: from knowledge transfer to one-shot learning of object categories. In *Proc. of British Machine Vision Conference (BMVC)*, 2009.

[162] T. Tommasi and B Caputo. Towards a quantitative measure of rareness. In *DIRAC Workshop at the European Conference on Machine Learning (ECML)*, 2012.

[163] T. Tommasi and T. Deselaers. ImageCLEF: The information retrieval series. volume 32, chapter The Medical Image Classification Task. Springer, 2010.

[164] T. Tommasi, F. Orabona, and B. Caputo. An SVM confidence-based approach to medical image annotation. In *Evaluating Systems for Multilingual and Multimodal Information Access – Proceedings of CLEF*, 2008.

[165] 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.

[166] T. Tommasi, F. Orabona, C. Castellini, and B. Caputo. Improving control of dexterous hand prostheses using adaptive learning. *IEEE Transaction on Robotics*, accepted as regular paper, September 2012, to appear.

[167] T. Tommasi, F. Orabona, M. Kaboli, and B. Caputo. Leveraging over prior knowledge for online learning of visual categories. In *British Machine Vision Conference (BMVC)*, 2012.

[168] T. Tommasi, N. Quadrianto, B. Caputo, and C.H. Lampert. Beyond dataset bias: Multi-task unaligned shared knowledge transfer. In *Asian Conference on Computer Vision*, 2012.

[169] A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *Computer Vision and Pattern Recognition (CVPR)*, 2011.

[170] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing visual features for multiclass and multiview object detection. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 29(5):854–869, 2007.

[171] L. Torresani, M. Szummer, and A. Fitzgibbon. Efficient object category recognition using classemes. In *European Conference on Computer Vision (ECCV)*, 2010.

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

[173] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *International Conference on Machine Learning (ICML)*, 2004.

[174] O. Tuzel, F. Porikli, and P. Meer. Human detection via classification on riemannian manifolds. In *Computer Vision and Pattern Recognition (CVPR)*, 2007.

[175] L. G. Valiant. A theory of the learnable. *Communications ACM*, 27(11):1134–1142, 1984.

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

[177] J. Vogel and B. Schiele. Semantic modeling of natural scenes for content-based image retrieval. *International Journal of Computer Vision (IJCV)*, 2008.

[178] Z. Wang, Y. Song, and C. Zhang. Transferred dimensionality reduction. In *European conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD)*, pages 550–565, 2008.

[179] K.Q. Weinberger and L.K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10:207–244, 2009.

[180] J. Yang, R. Yan, and A. G. Hauptmann. Adapting SVM classifiers to data with shifted distributions. In *International Conference on Data Mining Workshops (ICDM)*, 2007.

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

[182] Q. Yang, Y. Chen, G.R. Xue, W. Dai, and Y. Yu. Heterogeneous transfer learning for image clustering via the social web. In *Annual Meeting of the ACL and International Joint Conference on Natural Language Processing of the AFNLP*, 2009.

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

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

[185] B. Zadrozny. Learning and evaluating classifiers under sample selection bias. In *International Conference on Machine Learning (ICML)*, 2004.

[186] M. Zecca, S. Micera, M. C. Carrozza, and P. Dario. Control of multifunctional prosthetic hands by processing the electromyographic signal. *Critical Reviews in Biomedical Engineering*, 30(4–6):459–485, 2002.

[187] Y. Zhang and D.Y. Yeung. A convex formulation for learning task relationships in multi-task learning. In *Conference on Uncertainty in Artificial Intelligence (UAI-10)*, 2010.

[188] Y. Zhang and D.Y. Yeung. Transfer metric learning by learning task relationships. In *ACM SIGKDD International Conference on Knowledge discovery and data mining*, 2010.

[189] P. Zhao and S. C. H. Hoi. OTL: A framework of online transfer learning. In *International Conference on Machine Learning (ICML)*, 2010.

[190] Y. Zhu, Y. Chen, Z. Lu, S. J. Pan, G.R. Xue, and Q. Yu, Y.and Yang. Heterogeneous transfer learning for image classification. In *AAAI Conference on Artificial Intelligence*, 2011.

[191] A. Zweig and D. Weinshall. Exploiting object hierarchy: Combining models from different category levels. In *International Conference on Computer Vision (ICCV)*, 2007.

Tatiana Tommasi

Ph.D. candidate in Electrical Engeneering

Born March 21, 1981, Rome, Italy
Italian citizenship

Contacts:	c/o Idiap Research Institute	Telephone	+41 277 217 712
	Centre du Parc		mobile +41 774 583292
	P.O. Box 592		
	Rue Marconi 19	e-Mail	ttommasi@idiap.ch
	CH-1920 Martigny		
	Switzerland	web page	http://www.idiap.ch/~ttommasi

Tatiana Tommasi has been a Research Assistant at the Idiap Research Institute, Switzerland, since 2008. In the same year she was enrolled in the Doctoral School of Electrical Engineering at the Ecole Polytechnique Fédérale de Lausanne EPFL, Switzerland. She previously studied Physics at the University of Rome La Sapienza, Italy, where she ranked in the top 5% students graduated in 2004. In 2008 she received a Master of Science in Medical Physics at the same University where she received a scholarship as one of the three top best students entering the Postgraduate Specialization School in 2005.

Her main research interests are computer vision and machine learning with applications in the domain of open ended learning systems, medical imaging and robotics. Her work on transfer learning presented at the Computer Vision and Pattern Recognition conference in June 2010, has 28 citations and won the best poster award in the same year at the INRIA Visual Recognition and Machine Learning Summer School, Grenoble, France. Her work on principled algorithms for the automatic annotation of x-ray images resulted into two successful participations to the international ImageCLEF challenge (2007, 2008). Afterwards, she was invited to take part in the organization of the subsequent edition of the same challenge in 2009. Her first journal publication describing the approach proposed for the medical annotation task won the Idiap best paper award in 2008. This prize is assigned to the paper which receives the highest rank among six papers selected by the Idiap senior commission and finally evaluated by an international external committee. The same paper has up to now 37 citations. The referred number of citations are obtained through Google Scholar-My Citations which reports for Tatiana Tommasi an H-index of 7.

Tatiana Tommasi worked as teaching assistant for the course 'Cognitive Vision for Cognitive System' in 2010 at the Doctoral School of Electrical Engineering EPFL, Switzerland. She co-supervises Idiap internship students since 2011.

▬▬▬▬ Education

Doctoral School	**Oct. 2008 – Pres**.
	Ecole Polytechnique Fédérale de Lausanne EPFL, Switzerland
	PhD Thesis expected: Fall 2012
	Title: "Learning to Learn by Exploiting Prior Knowledge"
	Advisors: Dr. B. Caputo, Prof. H. Bourlard.
Postgraduate Specialization School in Medical Physics	**Jan. 2005 – Oct. 2008**
	University of Rome La Sapienza, Italy. Grade: 70/70 cum laude
	Thesis: "Multiple Cue Integration for Medical Image Annotation"
	Advisors: Prof. G.E. Gigante, Dr. B. Caputo.
Laurea degree in Physics	**Oct. 1999 – Jul. 2004**
	University of Rome La Sapienza, Italy. Grade: 110/110 cum laude
	Thesis: "Color study of skin lesions' images for the identification of melanoma's characteristics"
	Advisors: Prof. G. E. Gigante, Dr. V. Panichelli.

Professional Experience

Research Assistant

Sept. 2008 – Pres.
Idiap Research Institute, Martigny, Switzerland
Supervisor: Dr. B. Caputo.

Visiting Researcher

Jun. 2011 – Sept. 2011
Institute of Science and Technology Austria, Klosterneuburg, Austria
Supervisor: Prof. C. H. Lampert.

Jan. 2007 – Jun. 2007 and Feb. 2008 – Jul. 2008
Idiap Research Institute, Martigny, Switzerland
Supervisor: Dr. B. Caputo.

ImageCLEF organizer

Jan. 2009 – Sep. 2009
Medical Image Annotation Task.

Medical Physicist Intern

Jul. 2007 – Dec. 2007
Hospital "San Camillo-Forlanini", Rome, Italy
Supervisor: Dr. E. Santini.

Academic Activities

Teaching Assistant

Sept. 2010 – Jan. 2011
Ecole Polytechnique Fédérale de Lausanne EPFL, Switzerland
PhD course: Cognitive Vision for Cognitive Systems, Lecturer Dr. B. Caputo.

Student Co-supervisor

Feb. 2011 – Aug. 2011
Sriram Prasath, Bachelor student, internship at Idiap
Current Affiiliation: Master student at Royal Institute of Technology (KTH), Stockholm, Sweden.

Dec. 2011 – June 2012
Mohsen Kaboli, Master student, internship at Idiap.

Reviewer

Since 2008, reviewer for conferences (Computer Vision and Pattern Recognition (CVPR), International Conference on Computer Vision (ICCV), European Conference on Computer Vision (ECCV), British Machine Vision Conference (BMVC), Neural Information Processing Systems(NIPS), International Conference on Robotics and Automation (ICRA), and journal (IEEE Transaction on Robotics, Pattern Analysis and Machine Intelligence) papers in collaboration with Dr. B. Caputo.

Awards and Honors

Google Scholarship

2012, Google Anita Borg Memorial Scholarship. Finalist for Europe, Middle East and Africa.

Best Poster Award

2010, assigned to C7. INRIA Visual Recognition and Machine Learning Summer School, Grenoble, France.

Idiap Best Paper Award

2008, assigned to J4. Committee: Prof. C. Bishop (Microsoft Research, U.K.), Dr. J. Cohen (SRI International), Prof. J. Flanagan (Rutgers University), Prof. N. Morgan (ICSI, Berkeley), Dr. D. Nahamoo (IBM Research, Yorktown Heights), Prof. B. Yegnanarayana (IIIT, Hyderabad), Prof. S. Young (Cambridge University), Dr. H. Zhang (Microsoft Research Asia, Advanced Technology Center).

Ranked First

2007 and 2008, ImageCLEF challenge (www.imageclef.org), Medical Annotation Task.

Scholarship	**2007 and 2008,** winner of the Stilftelsen BLANCEFLOR Boncompagni Ludovisi (www.blanceflor.se) grant for study abroad.
Scholarship	**2005**, winner of the Specialization School Medical Physics student grant for the three top students in the selective contest to access the school.
Top 5% students	**2004**, belonging to the top 5% students graduated in Physics at the University of Rome, La Sapienza.

▬▬▬▬ Publications

Journals

J1.   **T. Tommasi**, F. Orabona, C. Castellini, B. Caputo. "Improving Control of Dexterous Hand Prostheses Using Adaptive Learning", IEEE Transaction on Robotics, accepted 2012.

J2.   C. Claudio, **T. Tommasi**, N. Noceti, F. Odone, B. Caputo. "Using Object Affordances to Improve Object Recognition", IEEE Transactions on Autonomous Mental Development, 3:207-215, 2010.

J3.   E. La Torre, B. Caputo, **T. Tommasi.** "Learning Methods for Melanoma Recognition", International Journal of Imaging Systems and Technology, 20:316-322, 2010.

J4.   **T. Tommasi**, F. Orabona, B. Caputo: "Discriminative Cue Integration for Medical Image Annotation", Pattern Recognition Letters, 29 (15), 2008.

Peer Reviewed Conference Proceedings

C1.   **T. Tommasi**, N. Quadrianto, B. Caputo, C.H. Lampert. "Multi-Task Unaligned Shared Knowledge Transfer", Asian Conference on Computer Vision (ACCV), 2012.

C2.   **T. Tommasi**, F. Orabona, M. Kabohli, B. Caputo. "Leveraging over prior knowledge for online learning of visual categories", British Machine Vision Conference (BMVC), 2012.

C3.   L.Jie*, **T. Tommasi***, B. Caputo. "Multiclass Transfer Learning from Unconstrained Priors", International Conference on Computer Vision (ICCV), 2011. (* equal authors listed in alphabetic order).

C4.   F. Nater*, **T. Tommasi***, H. Grabner, L. Van Gool, B. Caputo. "Transferring Activities: Updating Human Behavior Analysis", Visual Surveillance Workshop at the International conference on Computer Vision (ICCV) 2011. (* equal authors listed in alphabetic order).

C5.   **T. Tommasi**, B. Caputo. "Towards a quantitative measure of rareness", DIRAC Workshop at the European Conference on Machine Learning (ECML), 2010.

C6.   A. Gijsberts, **T. Tommasi**, G. Metta, B. Caputo. "Object Recognition using Visuo-Affordance Maps", International Conference on Intelligent Robots and Systems (IROS), 2010.

C7.   **T. Tommasi**, F. Orabona, B. Caputo. "Safety in Numbers: Learning Categories from Few Examples with Multi Model Knowledge Transfer", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010.

C8.   **T. Tommasi**, B. Caputo, P. Welter, M. O. Guld and T. M. Deserno. "Overview of the CLEF 2009 Medical Image Annotation Track", Proceedings of the 10th CLEF Workshop 2009, Lecture Notes in Computer Science.

C9.   **T. Tommasi**, B. Caputo. "The more you know, the less you learn: from knowledge transfer to one-shot-learning of objects categories", British Machine Vision Conference (BMVC), 2009.

C10.   **T. Tommasi**, F. Orabona, B. Caputo. "An SVM Confidence-Based Approach to Medical Image Annotation", Proceedings of the 9th CLEF Workshop 2008, Lecture Notes in Computer Science.

C11.   **T. Tommasi**, F. Orabona, B. Caputo. "Cue Integration for Medical Image Annotation", Proceedings of the 8th CLEF Workshop 2007, Lecture Notes in Computer Science.

C12.   E. La Torre, **T. Tommasi**, B. Caputo, G. E. Gigante. "Kernel Methods for Melanoma Recognition" Stud.Health Technol Inform, Proceedings of the International Congress of the European Federation for Medical Informatics (MIE), 2006.

C13.   **T. Tommasi**, E. La Torre, B. Caputo. "Melanoma Recognition Using Representative and Discriminative Kernel Classifiers", Proceedings of Workshop on Computer Vision Approaches to Medical Image Analysis (CVAMIA), 2006.

Book Chapters

B1.   **T. Tommasi**, T. Deselaers: "The Medical Image Classification Task", ImageCLEF, The Information Retrieval Series, Springer, Vol 32, Part 2, 221-238, 2010.

B2.   **T. Tommasi**, F. Orabona: "Idiap on Medical Image Classification", ImageCLEF, The Information Retrieval Series, Springer, Vol 32, Part 3, 453-465, 2010.

Technical Reports

T1.   S. Prasath Elango, **T. Tommasi**, B. Caputo: "Transfer Learning of Visual Concepts across Robots: a Discriminative Approach", Idiap-RR-06-2012.

──────── **Participation in Research Programmes**

DIRAC   Detection and Identification of Rare Audio-visual Cues. EC 6th Framework Programme supported IST Integrated Project (01/2006-12/2010). http://www.dirac.uni-oldenburg.de/

EMMA   Enhanced Medical Multimedia data Access. Supported by the Hasler Foundation. http://www.idiap.ch/scientific-research/projects/enhanced-medical-multimedia-data-access

PASCAL2   Pattern Analysis, Statistical Modelling and Computational Learning. In particular, my internship at IST Austria was partially supported by the Liaison Programme. http://pascallin2.ecs.soton.ac.uk/Programmes/LI/

──────── **Languages**

Italian   native speaker
English   fluent
French   basic

──────── **Computer Skills**

Operating Systems   Linux, Windows
Programming Languages   C/C++ and Java Programming, Professional Course (Oct.2004 – June 2005)
Technical Computation   Matlab
Other   Latex, MS Office
Web development   HTML

──────── **References**

Dr. Barbara Caputo   Idiap Research Institute
                          Martigny, Switzerland
                          Email: bcaputo@idiap.ch

Prof. Christoph Lampert   Institute of Science and Technology (IST) Austria
                          Klosterneuburg, Austria
                          Email: chl@ist.ac.at

Prof. Francesco Orabona   Toyota Technological Institute at Chicago
                          Chicago, USA
                          Email: francesco@orabona.com

Prof. Henning Müller   Hes.so University of Applied Sciences, Western Switzerland, Sierre
                          University Hospitals of Geneva, Geneva
                          Switzerland
                          Email: henning.mueller@hevs.ch, henning.mueller@hcuge.ch

Prof. Giovanni E. Gigante University of Rome "La Sapienza", Physics Department
                          Rome, Italy
                          Email: giovanni.gigante@uniroma1.it