Knowl Inf Syst
DOI 10.1007/s10115-012-0516-7

REGULAR PAPER

Parsimonious unsupervised and semi-supervised domain adaptation with good similarity functions

Emilie Morvant · Amaury Habrard · Stéphane Ayache

Received: 10 December 2011 / Revised: 16 March 2012 / Accepted: 7 April 2012

Abstract In this paper, we address the problem of domain adaptation for binary classification. This problem arises when the distributions generating the source learning data and target test data are somewhat different. From a theoretical standpoint, a classifier has better generalization guarantees when the two domain marginal distributions of the input space are close. Classical approaches try mainly to build new projection spaces or to reweight the source data with the objective of moving closer the two distributions. We study an original direction based on a recent framework introduced by Balcan et al. enabling one to learn linear classifiers in an explicit projection space based on a similarity function, not necessarily symmetric nor positive semi-definite. We propose a well-founded general method for learning a low-error classifier on target data, which is effective with the help of an iterative procedure compatible with Balcan et al.'s framework. A reweighting scheme of the similarity function is then introduced in order to move closer the distributions in a new projection space. The hyperparameters and the reweighting quality are controlled by a reverse validation procedure. Our approach is based on a linear programming formulation and shows good adaptation performances with very sparse models. We first consider the challenging unsupervised case where no target label is accessible, which can be helpful when no manual annotation is possible. We also propose a generalization to the semi-supervised case allowing us to consider some few target labels when available. Finally, we evaluate our method on a synthetic problem and on a real image annotation task.

E. Morvant (✉) · S. Ayache
LIF-QARMA, CNRS, UMR 7279, Aix-Marseille University, 13013 Marseille, France
e-mail: emilie.morvant@lif.univ-mrs.fr

S. Ayache
e-mail: stephane.ayache@lif.univ-mrs.fr

A. Habrard
Lab. Hubert Curien, CNRS, UMR 5516, University of St-Etienne, 42000 St-Etienne, France
e-mail: amaury.habrard@univ-st-etienne.fr

Published online: 05 July 2012

Keywords Machine learning · Transfer learning · Domain adaptation · Good similarity functions

1 Introduction

In applications requiring automatic classification of new data, a usual method is to learn a classifier by using some machine learning technique. In general, most of the approaches for learning classifiers are built under the assumption that the learning data are representative of the test data. In other words, all train and test data are supposed to be drawn from the same (usually unknown) distribution.

While this assumption can be relevant for some tasks, it is not always true in every applications. For example, in spam filtering problems, the training data associated with one user can be very different from e-mails received by another target user [32]. A similar issue can also happen in some image annotation tasks where training data can be restricted to particular instances (due to the tricky and costly manual labeling of examples) like images crawled from particular Web sites. Such training data are not representative of future test data that can come from images extracted from movies or videos. To overcome this drawback, some transfer learning methods [5,27,32,37,44] have been proposed to adapt a model from a source domain to a target domain. In this paper, we address a particular transfer learning task named *domain adaptation* (DA) where test data are supposed to be drawn according to a distribution—the *target domain*—different from the one used for generating learning data—the *source domain*—[29,38]. DA is thus an important issue for the efficient application of machine learning methods leading to the development of many approaches in the literature.

Under the assumption that the two domains are somehow related, theoretical DA results state that whether the source and the target marginal distributions over the input space are relatively close, then a classifier only learned from the labeled source data can perform well on the target domain [9,10,36]. This suggests a natural approach for a successful DA: Moving closer the source and the target distributions while keeping a low-error classifier on the source domain.

In this context, Ben-David et al. [9,10] have theoretically analyzed the importance of data representation for DA tasks by deriving a generalization bound of the target error for binary classifiers. Mansour et al. [36] have extended this approach to real-valued classifiers with more general results and other generalization bounds. In a DA scenario, two settings are generally considered (see Fig. 1): One where labeled data are only available in the source learning sample. This setting is often called *unsupervised domain adaptation* as it works in an unsupervised way over the target domain.

In the second, a few labeled data are also available in the target sample that corresponds to the *semi-supervised domain adaptation* setting. In this context, it is generally assumed that the number of target labeled data is significantly smaller than that of source labeled instances and not sufficiently large enough to learn a performing model only from the target labeled examples.

The unsupervised case is clearly more challenging. Some methods, based on different hypotheses or discrepancy measures, have been explored for reweighting the learning source data in order to move them closer to target data [28,30,36,42]. In another context, [14] have designed a SVM-based procedure that iteratively replaces source labeled instances by self-labeled target points in the learning sample. Another idea consists in finding a common relevant feature space where the two distributions are close [5,9,10,13,48], but this often relies on *ad hoc* heuristics specific to particular tasks. We can also cite some approaches based

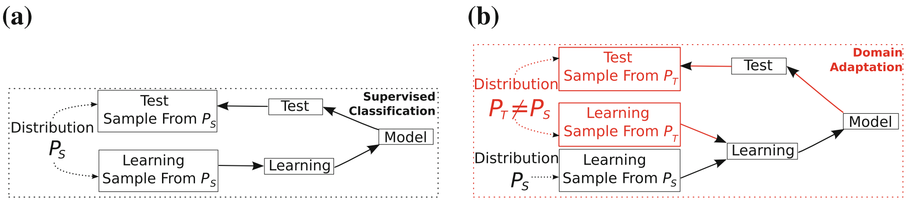


Fig. 1 The intuition behind the difference between a classical machine learning setting and a classical domain adaptation setting. P_S is the distribution generating the source data, and P_T is the distribution generating the target data. Note that to simplify the presentation, we use an abuse of notation for distributions generating the samples: unlabeled data of the test and target samples are not exactly drawn from the same distribution used to generate the labeled learning data, see Sect. 2 for a more formal description. **a** Usual supervised learning: the test data and learning data are generating from the same distribution. **b** Domain adaptation: the learning data are decomposed into two samples, the source (labeled) one and the target one. In the unsupervised DA, the target sample is provided without any label, while in the semisupervised DA, it includes some labels (color figure online)

on co-training [18] or regression [19] that can enter in this unsupervised setting. In general, all these previous methods have some natural extensions to the semi-supervised case in order to exploit target labeled information for improving the classifier induction. In this latter setting, some specific approaches have been proposed for statistical classifiers by using an extended linear projection space [20,21]. Some other techniques, using a combination of source and target labeled instances, have also been studied according to various frameworks [9,12,21,39].

Many of the previously cited methods are often based on either heuristics, a source reweighting scheme only, the presence target labels or kernel methods requiring the use of symmetric and positive semi-definite (PSD) similarity functions.

In this article, we propose a new domain adaptation approach based on the novel theory introduced in [6,7] for binary classification. This framework allows one to learn in an explicit projection space defined by a *good similarity function* that may be not symmetric nor PSD. In other words, it generalizes kernel functions of SVM-based methods and is thus more flexible in some sense. The authors show that it is possible to learn a low-error linear classifier in that space, defined by similarities to some relevant landmark examples. We claim that these landmarks offer a natural set of features to *transfer*. Our idea consists in automatically modifying this projection space for moving closer source and target points. For this purpose, we propose a general method based on the optimization of a regularized convex objective function where the regularization term plays a crucial role. Indeed, this term focuses on landmark points close to both source and target examples. Our optimization problem is in fact formulated in a 1-norm regularized linear program leading naturally to very sparse models. We also propose an iterative process, based on a reweighting of similarities, to improve the tractability of the method. The key point of our approach relies on the use of general similarities (i.e., neither symmetric nor PSD) to find a relevant projection space for domain adaptation allowing us to move closer source and target distributions. This explains, why we propose to stand in the framework of Balcan et al's to design our DA method.

Our contribution is twofold. First, we define a method for the challenging unsupervised case where no target label is available. It provides then a solution to compensate the lack of target labeled data when manual labeling is impossible. It can also be useful, for example, to design a "cold start" strategy in an active learning process to label the very first examples [3]. In this unsupervised setting, a crucial point is to find a reliable method for assessing the various hyperparameters of our approach. To solve this problem, we propose to make use of approaches based on the notion of reverse validation [14,51]. We exploit this notion to propose a stopping criterion for our iterative procedure. We also present a theoretical analysis of our models in terms of sparsity and generalization guarantees. We derive a new error bound based on the notion of algorithmic robustness [45]. Our second contribution takes the form of a generalization of our first approach to the semi-supervised case in order to take into account some existing target labels. It is inspired by [9] and based on the optimization of a linear combination of the source and target empirical errors. We also provide some theoretical justifications specific to this semi-supervised case. Our two methods are evaluated on a synthetic problem and on real image annotation corpora.

The paper is organized as follows. Section 2 introduces the domain adaptation framework of [9]. Section 3 deals with the theory of learning with good similarity functions [6]. Our unsupervised approach is presented in Sect. 4 and its iterative enhancement in Sect. 5. Our semi-supervised method using a few target labels is formulated in Sect. 6. The different approaches are experimentally evaluated in Sect. 7. Finally, we conclude and discuss some future work in Sect. 8.

2 Domain adaptation

Let $X \subseteq \mathbb{R}^d$ be the input space of dimension d and $Y = \{-1, +1\}$ the label set. A domain is defined as a probability distribution over $X \times Y$. In a DA framework [9,36], we have a *source domain* represented by a distribution P_S over $X \times Y$ and a *target domain* represented by a somewhat different distribution P_T, D_S, and D_T being the respective marginal distributions over X.

In the unsupervised case, a learning algorithm is provided with a *Labeled Source sample* $LS = \{(\mathbf{x}_i, y_i)\}_{i=1}^{d_l}$ drawn *i.i.d.* from P_S and an *unlabeled Target Sample* $TS = \{\mathbf{x}_{i'}\}_{i'=1}^{d_t}$ drawn *i.i.d.* from D_T. We also denote by $LS_{|X} = \{(\mathbf{x}_i)/(\mathbf{x}_i, y_i) \in LS\}_{i=1}^{d_l}$ the sample constituted of all the instances of LS without their label. Let $h : X \rightarrow Y$ be an hypothesis function in the form of a binary classifier. The expected errors of h over the source domain P_S and the target domain P_T are the probabilities that h commits an error on P_S and P_T, respectively,

$$\mathrm{err}_S(h) = \mathbb{E}_{(\mathbf{x},y)\sim P_S}\, L_{01}\big(h, (\mathbf{x}, y)\big)\,, \quad \mathrm{err}_T(h) = \mathbb{E}_{(\mathbf{x},y)\sim P_T}\, L_{01}\big(h, (\mathbf{x}, y)\big),$$

where $L_{01}(h, (\mathbf{x}, y)) = 1$ if $h(\mathbf{x}) \neq y$ and zero otherwise, corresponding to the 0–1 *loss function*. We denote by $\hat{\mathrm{err}}_S(h)$ and $\hat{\mathrm{err}}_T(h)$ the respective empirical errors. A hypothesis class $\mathcal{H}$ is a set of hypotheses from X to Y. For a DA task, the objective is then to learn a classifier $h \in \mathcal{H}$ with a low generalization error $\mathrm{err}_T(h)$ over the target domain (see Fig. 1b).

We now review the theoretical framework[1] of DA based on [9], where the authors give an upper bound for $\mathrm{err}_T(h)$.

Theorem 1 [9] *Let $\mathcal{H}$ be a hypothesis class, $\forall h \in \mathcal{H}$,*

$$\mathrm{err}_T(h) \;\leq\; \mathrm{err}_S(h) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(D_S, D_T) + \nu,$$

where $d_{\mathcal{H}\Delta\mathcal{H}}(D_S, D_T) = 2 \sup_{h,h' \in \mathcal{H}\Delta\mathcal{H}} |Pr_{D_S}(h(\mathbf{x}) \neq h'(\mathbf{x})) - Pr_{D_T}(h(\mathbf{x}) \neq h'(\mathbf{x}))|$ *is the $\mathcal{H}\Delta\mathcal{H}$-distance between D_S and D_T with $\mathcal{H}\Delta\mathcal{H} = \{h(\mathbf{x}) \oplus h'(\mathbf{x}) : h, h' \in \mathcal{H}\}$ the symmetric difference hypothesis space of $\mathcal{H}$ and $\nu = \mathrm{err}_S(h^*) + \mathrm{err}_T(h^*)$ is the error of the ideal joint hypothesis with $h^* = \mathrm{argmin}_{h\in\mathcal{H}}(\mathrm{err}_S(h) + \mathrm{err}_T(h))$.*

[1] Note that surveys can be found in [29,38].

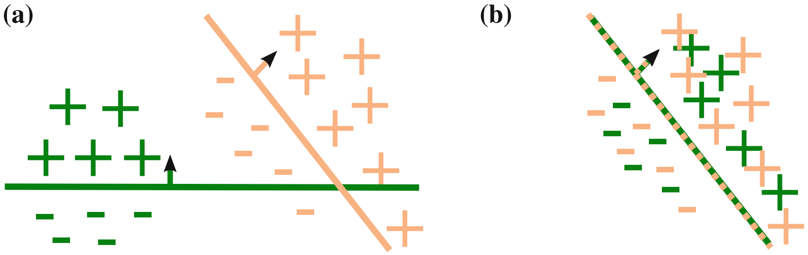


Fig. 2 The intuition behind Theorem 1. The source domain points are in (*dark*) *green* (pos.+, neg.−), the target domain points are in (*light*) *orange*. **a** A large distance between the marginal distributions: the samples are easily separable, the classifier learned from the source domain performs badly on the target one. **b** A small distance between the marginal distributions: the classifier learned from the source domain performs well on both domains (color figure online)

This bound depends on three terms:

(a) The source domain expected error $\mathrm{err}_S(h)$ which can be minimized by a learning algorithm based on the ERM principle.
(b) The $\mathcal{H}\Delta\mathcal{H}$-distance between the two marginal distributions which is related to $\mathcal{H}$ by measuring a maximum variation divergence over the set of points on which an hypothesis can commit errors.
(c) The last term ν is related to the ideal joint hypothesis h^* over the domains and can be seen as a quality measure of $\mathcal{H}$ for the considered DA task. If h^* performs poorly, then it seems to be hard to find a low-error hypothesis on the target domain.

Theorem 1 suggests that if the $\mathcal{H}\Delta\mathcal{H}$-distance is low, that is, if the two marginal distributions are close, then a low-error classifier over the source domain might be a good model over the target one. The intuition behind this idea is given on Fig. 2.

An interesting point, described by the following Lemma, is that when the VC dimension of $\mathcal{H}$ is finite (measuring the capacity of $\mathcal{H}$), $d_{\mathcal{H}\Delta\mathcal{H}}(D_S, D_T)$ can be estimated from finite samples.

Lemma 1 [9] *Let $\mathcal{H}$ be an hypothesis class with finite VC-dimension v. Let S and T be unlabeled samples of size m i.i.d. from D_S and D_T, respectively. Let $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(S, T)$ be the empirical $\mathcal{H}\Delta\mathcal{H}$-distance. Then, for any $\delta > 0$ with probability at least $1 - \delta$,*

$$d_{\mathcal{H}\Delta\mathcal{H}}(D_S, D_T) \leq \hat{d}_{\mathcal{H}\Delta\mathcal{H}}(S, T) + 4\sqrt{\frac{2v\log(2m) + \log\frac{2}{\delta}}{m}}.$$

Lemma 1 means that the empirical distance $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(S, T)$ converges thus to the real one $d_{\mathcal{H}\Delta\mathcal{H}}(D_S, D_T)$ with the size m of the samples. Consider a labeled sample made of $S \cup T$ where each instance of S is labeled as positive and each one of T as negative, we can directly estimate $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(S, T)$ ($\in [0, 2]$) by looking for the best classifier able to separate S from T,

$$\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(S, T) = 2\left(1 - \min_{h\in\mathcal{H}\Delta\mathcal{H}} \hat{\mathrm{err}}_{S\cup T}(h)\right), \tag{1}$$

$$\text{with } \hat{\mathrm{err}}_{S\cup T}(h) = \frac{1}{m}\left[\sum_{\substack{\mathbf{x}\in S\cup T:\\ h(\mathbf{x})=-1}} \mathbb{1}_{\mathbf{x}\in S} + \sum_{\substack{\mathbf{x}\in S\cup T:\\ h(\mathbf{x})=1}} \mathbb{1}_{\mathbf{x}\in T}\right], \text{ where } \mathbb{1}_{\mathbf{x}\in A} = \begin{cases} 1 & \text{if } \mathbf{x} \in A,\\ 0 & \text{otherwise.}\end{cases}$$

Finding the optimal hyperplane is an NP-hard problem in general. However, an estimation of $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(S, T)$, and thus of $d_{\mathcal{H}\Delta\mathcal{H}}(D_S, D_T)$, allows us to have an insight into the distribution distance and thus of the difficulty of the DA task for the class $\mathcal{H}$. Note that [36] have extended the $\mathcal{H}\Delta\mathcal{H}$-distance to real-valued functions and have provided Rademacher generalization bounds.

Following Theorem 1, one solution for a DA algorithm is to look for a data projection space where both the $\mathcal{H}\Delta\mathcal{H}$-distance (b) and the source domain expected error of a classifier (a) are low (see Fig. 2). According to [11], minimizing these two terms is necessary to ensure a good adaptation in general.

As a consequence, we need to define a projection space to work on in order to move closer the two distributions. Rather than working in the original input space, we propose to consider a projection space defined by similarity scores to particular points where a good predictor exists. This brings us to the framework of Balcan et al. making use of a notion of good similarity function and introduced in the next section.

3 Learning with good similarity functions

In this section, we present the framework of similarity-based binary linear classifiers introduced by [6,7]. Recall that a similarity function over X is any pairwise function $K : X \times X \to [-1, 1]$. Many algorithms use similarity functions, like support vector machines where the similarity needs to be a kernel (i.e., symmetric and positive semi-definite (PSD)) to ensure learning and convergence in an implicit high-dimensional Hilbert space. However, due to the PSD requirement, considering kernels can be a strong limitation and defining a relevant kernel is a tricky task in general (see [1] for a survey on kernel learning).

The recent learning framework proposed by [6] considers a rather intuitive definition of a good similarity function that overcomes some of these limitations.

Definition 1 [6] A similarity function K is an **(ϵ, γ, τ)-good similarity function** for a learning problem P if there exists a (random) indicator function $R(\mathbf{x})$ defining a (probabilistic) set of *reasonable points* such that the following conditions hold:

(i) A $1 - \epsilon$ probability mass of examples $(\mathbf{x}, y)$ satisfy

$$\mathbb{E}_{(\mathbf{x}', y') \sim P}\big[yy'K(\mathbf{x}, \mathbf{x}')|R(\mathbf{x}') = 1\big] \geq \gamma,$$

(ii) $Pr_{\mathbf{x}'}[R(\mathbf{x}') = 1] \geq \tau.$

This definition means that a large proportion of examples must be on average γ more similar to the reasonable points of the same class than to the reasonable points of the opposite class (condition (i)). Moreover, at least a proportion τ of the examples should be reasonable (condition (ii)). Definition 1 includes all valid kernels as well as some non-PSD similarity functions and is thus quite general [6,7]. The reasonable points are usually unknown a priori. Therefore, in the following, we denote by $R = \{\mathbf{x}'_j\}_{j=1}^{d_u}$ a set of *potential* reasonable points called *landmarks*. Given K an (ϵ, γ, τ)-good similarity function, the conditions of [6] are sufficient to learn a good linear classifier in a ϕ^R-space defined by the mapping function ϕ^R, which projects a point in the explicit space of the similarities to the landmarks such that,

$$\phi^R : \begin{cases} X \to \mathbb{R}^{d_u} \\ \mathbf{x} \mapsto \langle K(\mathbf{x}, \mathbf{x}'_1), \ldots, K(\mathbf{x}, \mathbf{x}'_{d_u})\rangle. \end{cases} \tag{2}$$

The following theorem justifies the existence of a good linear classifier in the ϕ^R-space.

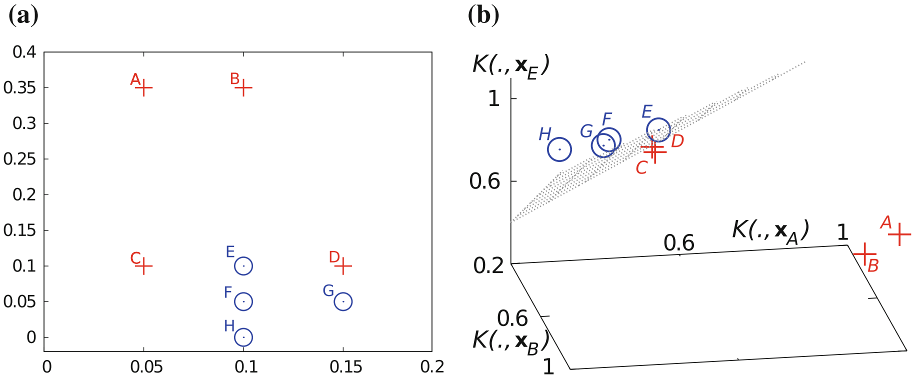


Fig. 3 A set of positive (*red crosses*) and negative (*blue circles*) points. **a** The data in the original 2D-space. **b** The projection of the data in the ϕ^R-space and the linear separator in *gray* (color figure online)

Theorem 2 [6] *Let K be an* **$(\boldsymbol{\epsilon},\boldsymbol{\gamma},\boldsymbol{\tau})$-good similarity function** *for a learning problem P. Let $R = \{\mathbf{x}'_1, \ldots, \mathbf{x}'_{d_u}\}$ be a (potentially unlabeled) sample of landmarks drawn i.i.d. from P such that $d_u = \frac{2}{\tau}\left(\log(2/\delta) + 8\frac{\log(2/\delta)}{\gamma^2}\right)$. Consider the mapping ϕ^R defined in Eq. (2). Then, with probability at least $1-\delta$ over the random sample R, the induced distribution $\phi^R(P)$ in $\mathbb{R}^{d_u}$ has a separator of error at most $\epsilon+\delta$ to L_1-margin at least $\gamma/2$.*

Thus, given an (ϵ,γ,τ)-good similarity function for a learning problem and—enough—landmarks, there exists with high probability a low-error linear separator in the explicit ϕ^R-space.

We now provide a little toy example in order to illustrate the notion of (ϵ,γ,τ)-good similarity function introduced by Definition 1. We consider a problem with only height labeled examples in $[0,1]\times[0,1]$ represented on Fig. 3a: $\mathbf{x}_A = \big((.05, .35), +1\big)$, $\mathbf{x}_B = \big((.10, .35), +1\big)$, $\mathbf{x}_C = \big((.05, .10), +1\big)$, $\mathbf{x}_D = \big((.15, .10), +1\big)$, $\mathbf{x}_E = \big((.10, .10), -1\big)$, $\mathbf{x}_F = \big((.10, .05), -1\big)$, $\mathbf{x}_G = \big((.15, .05), -1\big)$ and $\mathbf{x}_H = \big((.10, .00), -1\big)$. We can note here that because of $\mathbf{x}_E$, there exists no linear classifier that can achieve a null classification error in this original instance space.

We now consider a similarity function $K(\mathbf{x},\mathbf{x}') = 1-2\|\mathbf{x}-\mathbf{x}'\|_2$ with $\|\mathbf{x}-\mathbf{x}'\|_2$ the classical Euclidean distance. We take the opposite of the distance to obtain a similarity and the renormalization ensures that $K(\mathbf{x},\mathbf{x}') \in [-1,1]$. We suppose that three out of the height examples are reasonable points: $\mathbf{x}_A$, $\mathbf{x}_B$, and $\mathbf{x}_E$; τ can thus be estimated as $\frac{3}{8}$. We can then evaluate the goodness of K according to these reasonable points from the formula given in Definition 1. The corresponding values are shown in Table 1. If we take a margin $\gamma = 0.002$, we can remark that the goodness of each example is larger than γ, which makes the similarity $(0, 0.002, 3/8)$-good. Now, with $\gamma = 0.02$, the similarity is $(0.25, 0.02, 3/8)$-good since two examples out of the height do not achieve a goodness larger than 0.02.

Finally, in the explicit projection space defined by the similarities to the three reasonable points $\phi^R(\cdot) =< K(\cdot,\mathbf{x}_A), K(\cdot,\mathbf{x}_B), K(\cdot,\mathbf{x}_E) >$, there exists a linear classifier $sign\big(g(\cdot)\big)$ that has a null error, where g is of the form $g(\cdot) = \alpha_A K(\cdot,\mathbf{x}_A) + \alpha_B K(\cdot,\mathbf{x}_B) + \alpha_C K(\cdot,\mathbf{x}_E)$ (see Fig. 3b, a possible admissible solution is obtained with $\alpha_A = \alpha_B = 1$ and $\alpha_E = -1$).

The criterion given by Definition 1 requires to minimize the number of margin violations, which is a NP-hard problem generally difficult to approximate. To overcome this problem, the authors have then proposed to consider an adaptation of Definition 1 with the hinge loss formalized as follows.

Table 1 Example of the goodness of a similarity function

	$K(\cdot, \mathbf{x}_A)$	$K(\cdot, \mathbf{x}_B)$	$K(\cdot, \mathbf{x}_E)$	$Goodness\big((\mathbf{x}, y)\big) =$ $\mathbb{E}[yy'K(\mathbf{x}, \mathbf{x}')\|R(\mathbf{x}') = 1]$
$\mathbf{x}_A$	1	0.90	0.68	0.410
$\mathbf{x}_B$	0.90	1	0.70	0.400
$\mathbf{x}_C$	0.50	0.49	0.90	0.030
$\mathbf{x}_D$	0.46	0.49	0.90	0.017
$\mathbf{x}_E$	0.49	0.50	1	0.003
$\mathbf{x}_F$	0.39	0.40	0.90	0.037
$\mathbf{x}_G$	0.37	0.39	0.86	0.033
$\mathbf{x}_H$	0.29	0.30	0.8	0.070

The table provides for each example the similarity scores to every reasonable points and its associated goodness

Definition 2 [6] A similarity function K is an **$(\boldsymbol{\epsilon},\boldsymbol{\gamma},\boldsymbol{\tau})$-good similarity function in hinge loss** for a learning problem P if there exists a (random) indicator function $R(\mathbf{x})$ defining a (probabilistic) set of reasonable points such that the following conditions hold:

(i) $\mathbb{E}_{(\mathbf{x},y)\sim P}\Big[[1 - yg(\mathbf{x})/\gamma]_+\Big] \leq \epsilon$, where $g(\mathbf{x}) = \mathbb{E}_{(\mathbf{x}',y')\sim P}[y'K(\mathbf{x}, \mathbf{x}')|R(\mathbf{x}')]$ and $[1 - z]_+ = \max(0, 1 - z)$ is the hinge loss,
(ii) $Pr_{\mathbf{x}'}[R(\mathbf{x}')] \geq \tau$.

Using the same ϕ^R-space than Theorem 2, the authors have proved a similar theorem for this definition with the hinge loss.

Theorem 3 [6] *Let K be an* **$(\boldsymbol{\epsilon},\boldsymbol{\gamma},\boldsymbol{\tau})$-good similarity function in hinge loss** *for a learning problem P. For any $\epsilon_1 > 0$ and $0 < \delta < \frac{\gamma\epsilon_1}{4}$, let $R = \{\mathbf{x}'_1, \ldots, \mathbf{x}'_{d_u}\}$ be a sample of $d_u = \frac{2}{\tau}\left(\log(2/\delta) + 16\frac{\log(2/\delta)}{(\gamma\epsilon_1)^2}\right)$ landmarks drawn i.i.d. from P. Consider the mapping ϕ^R defined in Eq. 2. Then, with probability at least $1 - \delta$ over the random sample R, the induced distribution $\phi^R(P)$ in $\mathbb{R}^{d_u}$ has a separator achieving hinge loss of error at most of $\epsilon + \epsilon_1$ at margin γ.*

Finally, given LS a set of d_l labeled points and d_u landmark examples, one can efficiently find a separator $\boldsymbol{\alpha} \in \mathbb{R}^{d_u}$ by solving a linear program where the objective is to minimize the number of margin violations with the hinge loss. We give here an equivalent formulation of the $L1$-constrained problem presented by [6], called SF_{opt}, which is based on the hinge loss.

$$\begin{cases} \displaystyle\min_{\boldsymbol{\alpha}} \frac{1}{d_l}\sum_{i=1}^{d_l} L\big(g, (\mathbf{x}_i, y_i)\big) + \lambda\|\boldsymbol{\alpha}\|_1, \\ \text{with } L\big(g, (\mathbf{x}_i, y_i)\big) = \Big[1 - y_i g(\mathbf{x}_i)\Big]_+ \quad \text{and} \quad g(\mathbf{x}_i) = \displaystyle\sum_{j=1}^{d_u} \alpha_j K(\mathbf{x}_i, \mathbf{x}'_j), \end{cases} \qquad (SF_{opt})$$

where $g(.)$ is the learned model. The $L1$-regularization over $\boldsymbol{\alpha}$ produces an automatic selection of the reasonable points from the landmarks because of the sparsity property of the $L1$-norm. This leads to a natural two steps algorithm for learning the classifier: (1) Select a random set of potential landmarks and then (2) learn a binary classifier $h(\mathbf{x}) = \text{sign}[g(\mathbf{x})]$, in the space induced by the selected landmarks, that is, those with $\alpha_j \neq 0$. In practise, the landmarks are chosen from the learning sample.

once a ϕ^R-space (of dimension d') has been defined after this learning step, then the class $\mathcal{H}_{\phi^R}$ of linear classifiers learnable in this space has a finite VC-dimension $(d' + 1)$. Thus, according to Lemma 1, we can assess the distribution divergence in the ϕ^R-space by the empirical estimate $\hat{d}_{\mathcal{H}_{\phi^R}\Delta\mathcal{H}_{\phi^R}}(D_S, D_T)$.

In the following, a linear classifier learned in this framework by solving (SF_{opt}) is called a SF-classifier. For sake of simplicity, we will denote $\mathcal{H}_{\phi^R}$ by $\mathcal{H}$.

Finally, by considering the minimization of the DA bound of Theorem 1, we remark that (SF_{opt}) can be seen both as an empirical minimization of the error on the source domain $P_S = P$ and as a method for building a relevant ϕ^R-space. Our idea for DA is then to constrain the ϕ^R-space to minimize the $\mathcal{H}\Delta\mathcal{H}$-distance.

4 Unsupervised domain adaptation with similarity functions

We now present our unsupervised DA method, which consists in learning a classifier from (ϵ,γ,τ)-good similarity functions. Recall that following Theorem 1, the expected target domain error is bounded by three terms: (a) the source domain error, (b) the divergence between the marginal distributions, and (c) the smallest joint error over the domains. Our idea is to minimize the expected target error by decreasing this bound.

According to [6,7], solving Problem SF_{opt}, only on the source domain, involves to learn a relevant linear classifier in the explicit ϕ^R-space [Eq. (2)] of similarities to a landmark set. Then, it implies a natural decreasing of (a). For minimizing (b), we want to induce a new projection space allowing one to move closer the two domain marginal distributions by selecting landmarks that are both similar to the source and target examples. To achieve this goal, we propose to learn a classifier thanks to an additional regularization term on the weights $\boldsymbol{\alpha}$. Due to the lack of information on the target domain, the last term (c) is hard to decrease. However, we propose to use a reverse validation approach to try to control it.

4.1 Optimization problem

By solving Problem (SF_{opt}) for learning SF-classifiers, we not only minimize the expected source error but we also define a relevant projection space for the source domain. Indeed, irrelevant landmarks, that is, those associated with a null weight in the solution $\boldsymbol{\alpha}$, will not be considered. According to the notion of $\mathcal{H}\Delta\mathcal{H}$-distance [Eq. (1)], we propose a new additional regularizer that forces the model to provide similar outputs for pairs of source and target points. This will tend to decrease the $\mathcal{H}\Delta\mathcal{H}$-distance between the marginal distributions. To define our regularizer, we have investigated the framework of *algorithmic robustness* proposed by [45] (see Definition 3 in Sect. 4.2). Their underlying idea is based on the fact that "if a testing sample is similar to a training sample then the testing error is close to the training error". To ensure generalization guarantees, this framework requires that for a test point close to a training point of the same label, the deviation between the losses of each point has to be low. Note that this result assumes the test and training data to be generated from the same distribution, that is thus not valid in a DA scenario.

Despite this drawback, we propose to follow this principle by defining an heuristic to move closer source and target samples.

By considering the hinge loss of the Problem (SF_{opt}), for any learned model g and any pair $(\mathbf{x}_s, \mathbf{x}_t)$ of source and target examples of class y, we have

$$\begin{aligned}
&\Big|L\big(g,(\mathbf{x}_s,y)\big)-L\big(g,(\mathbf{x}_t,y)\big)\Big|\\
&\quad=\left|\Big[1-y\sum_{j=1}^{d_u}\alpha_j K(\mathbf{x}_s,\mathbf{x}'_j)\Big]_+-\Big[1-y\sum_{j=1}^{d_u}\alpha_j K(\mathbf{x}_t,\mathbf{x}'_j)\Big]_+\right|.
\end{aligned}$$

The hinge loss is 1-lipschitz ($|[X]_+ - [Y]_+| \leq |X-Y|$) then,

$$\begin{aligned}
\Big|L\big(g,(\mathbf{x}_s,y)\big)-L\big(g,(\mathbf{x}_t,y)\big)\Big| &\leq \left|\sum_{j=1}^{d_u}\alpha_j\big(K(\mathbf{x}_s,\mathbf{x}'_j)-K(\mathbf{x}_s,\mathbf{x}'_j)\big)\right|\\
&\leq \sum_{j=1}^{d_u}\left|\alpha_j\big(K(\mathbf{x}_s,\mathbf{x}'_j)-K(\mathbf{x}_t,\mathbf{x}'_j)\big)\right|\\
&\leq \left\|({}^t\phi^R(\mathbf{x}_s)-{}^t\phi^R(\mathbf{x}_t))\,\mathrm{diag}(\boldsymbol{\alpha})\right\|_1, \qquad (3)
\end{aligned}$$

where ${}^t\phi^R(\cdot)$ is the transposed vector of $\phi^R(\cdot)$ and $\mathrm{diag}(\boldsymbol{\alpha})$ is the diagonal matrix with $\boldsymbol{\alpha}$ as main diagonal.

Minimizing the term of line (3) amounts to reducing the deviation between source and target instances $\mathbf{x}_s$ and $\mathbf{x}_t$ of the same class. This would lead to select landmarks that move closer x_s and x_t and consequently reducing the domain divergence.

At this point, we assume that the pairs $(\mathbf{x}_s,\mathbf{x}_t)$ are known and let $\mathcal{C}_{ST}\subset LS_{|X}\times TS$ be the pair set. We propose to add the new regularization term of line (3) for each pair of $\mathcal{C}_{ST}$, weighted by a regularization parameter β to tune.

Let R be a set of d_u candidate landmarks and LS a source sample of d_l source labeled examples. Our global optimization Problem ($DASF_{opt}$) corresponds to Problem (SF_{opt}) with the addition of our regularizer and can be easily formulated as a linear program.

$$\begin{cases}
\min\limits_{\boldsymbol{\alpha}}\ F(\boldsymbol{\alpha})=\dfrac{1}{d_l}\displaystyle\sum_{i=1}^{d_l}L\big(g,(\mathbf{x}_i,y_i)\big)+\lambda\|\boldsymbol{\alpha}\|_1\\
\qquad\qquad +\beta\displaystyle\sum_{(\mathbf{x}_s,\mathbf{x}_t)\in\mathcal{C}_{ST}}\left\|\big({}^t\phi^R(\mathbf{x}_s)-{}^t\phi^R(\mathbf{x}_t)\big)\,\mathrm{diag}(\boldsymbol{\alpha})\right\|_1,\\
\text{with } L\big(g,(\mathbf{x}_i,y_i)\big)=\Big[1-y_i g(\mathbf{x}_i)\Big]_+ \quad\text{and}\quad g(\mathbf{x}_i)=\displaystyle\sum_{j=1}^{d_u}\alpha_j K(\mathbf{x}_i,\mathbf{x}'_j).
\end{cases} \qquad (DASF_{opt})$$

This linear problem is a convex program. It can be solved by using d_l slack variables for expressing the hinge loss that leads to a program with $O(d_l+d_u)$ variables with $O(d_l\times d_u)$ constraints.

4.2 Theoretical aspects

In this section, we provide a theoretical sparsity analysis of our optimization Problem ($DASF_{opt}$) and derive a generalization error bound.

We first need the following hypothesis about the pair set $\mathcal{C}_{ST}$. Concretely, since our additional regularizer is based on $\mathcal{C}_{ST}$ and contributes to find a relevant projection space, $\mathcal{C}_{ST}$ has to contain relevant information. We thus suppose a restriction on the coordinates in the ϕ^R-space of the points of $\mathcal{C}_{ST}$,

$$\forall \mathbf{x}'_j\in R,\ \max_{(\mathbf{x}_s,\mathbf{x}_t)\in\mathcal{C}_{ST}}\left|K(\mathbf{x}_s,\mathbf{x}'_j)-K(\mathbf{x}_t,\mathbf{x}'_j)\right| > 0. \qquad (4)$$

This means that for each coordinate $\mathbf{x}'_j$ in the ϕ^R-space, there is at least one pair of points that brings some diversity with different coordinate values. This is actually not a too strong restriction, since if the two domains are far from each other, then the assumption (4) occurs with high probability.

4.2.1 Sparsity analysis

As said before, in Balcan et al.'s Problem (SF_{opt}), the 1-norm regularizer $\|\boldsymbol{\alpha}\|_1$ on the learned vector $\boldsymbol{\alpha}$ implies a natural sparsity of the induced SF-classifier. Our Problem ($DASF_{opt}$) keeps this feature but analyzing its sparsity requires to also consider the additional regularization term over $\boldsymbol{\alpha}$. We provide here an analysis of the sparsity according to all the different hyperparameters.

Lemma 2 *For any hyperparameters $\lambda > 0$ and $\beta > 0$, and for any set of pairs $\mathcal{C}_{ST}$, let $B_R = \min_{\mathbf{x}'_j \in R} \big\{ \max_{(\mathbf{x}_s,\mathbf{x}_t)\in\mathcal{C}_{ST}} |K(\mathbf{x}_s,\mathbf{x}'_j) - K(\mathbf{x}_t,\mathbf{x}'_j)| \big\}$. If $\boldsymbol{\alpha}^*$ denotes the optimal solution of our Problem ($DASF_{opt}$), then we have $\|\boldsymbol{\alpha}^*\|_1 \leq \dfrac{1}{\beta B_R + \lambda}$.*

Proof Recall $F(.)$ refers to Problem ($DASF_{opt}$). For any solution $\boldsymbol{\alpha}$,

$$\begin{aligned}
&\sum_{(\mathbf{x}_s,\mathbf{x}_t)\in\mathcal{C}_{ST}} \|({}^t\phi^R(\mathbf{x}_s) - {}^t\phi^R(\mathbf{x}_t))\,\mathrm{diag}(\boldsymbol{\alpha})\|_1 \\
&\quad = \sum_{(\mathbf{x}_s,\mathbf{x}_t)\in\mathcal{C}_{ST}} \sum_{j=1}^{d_u} \left|\alpha_j \left(K(\mathbf{x}_s,\mathbf{x}'_j) - K(\mathbf{x}_t,\mathbf{x}'_j)\right)\right| \\
&\quad = \sum_{j=1}^{d_u} \left[|\alpha_j| \left(\sum_{(\mathbf{x}_s,\mathbf{x}_t)\in\mathcal{C}_{ST}} |K(\mathbf{x}_s,\mathbf{x}'_j) - K(\mathbf{x}_t,\mathbf{x}'_j)| \right) \right] \\
&\quad \geq \sum_{j=1}^{d_u} \left[|\alpha_j| \max_{(\mathbf{x}_s,\mathbf{x}_t)\in\mathcal{C}_{ST}} |K(\mathbf{x}_s,\mathbf{x}'_j) - K(\mathbf{x}_t,\mathbf{x}'_j)| \right].
\end{aligned}$$

From Hypothesis (4) and the definition of B_R, we have

$$B_R = \min_{\mathbf{x}'_j\in R} \Big\{ \max_{(\mathbf{x}_s,\mathbf{x}_t)\in\mathcal{C}_{ST}} |K(\mathbf{x}_s,\mathbf{x}'_j) - K(\mathbf{x}_t,\mathbf{x}'_j)| \Big\} > 0.$$

Thus, $\sum_{(\mathbf{x}_s,\mathbf{x}_t)\in\mathcal{C}_{ST}} \|({}^t\phi^R(\mathbf{x}_s) - {}^t\phi^R(\mathbf{x}_t))\,\mathrm{diag}(\boldsymbol{\alpha})\|_1 \geq \|\boldsymbol{\alpha}\|_1 B_R$.
Then, $\|\boldsymbol{\alpha}^*\|_1(\lambda + \beta B_R) + \frac{1}{d_l}\sum_{i=1} d_l \Big[1 - y_i \sum_{j=1} d_u \alpha^*_j K(\mathbf{x}_i,\mathbf{x}'_j)\Big]_+ \leq F(\boldsymbol{\alpha}^*)$.
Since $\boldsymbol{\alpha}^*$ is optimal, we have $F(\boldsymbol{\alpha}^*) \leq F(\mathbf{0}) = 1$, where $\mathbf{0}$ is the null vector.

Finally, we directly obtain $\|\boldsymbol{\alpha}^*\|_1 \leq \dfrac{1}{\beta B_R + \lambda}$. □

According to this lemma, the sparsity of the model depends on the hyperparameters λ, β and on the quantity B_R. This last term is in fact related to the distance between the points in the pair set $\mathcal{C}_{ST}$. In the projection space, it is the minimum of the maximum deviation between the coordinates of the pair's points belonging to $\mathcal{C}_{ST}$. Thus, when the two marginal distributions are far from each other, that is, the DA task is potentially hard, B_R tends to be high that can imply an increase of the sparsity. Indeed, with sparser models, the projection space defined is smaller (i.e., with less features), which tends to make closer source and target instances more easily with less constraints to take into account.

4.2.2 Generalization ability

Algorithmic robustness We recall now the definition of robustness and its associated theorem about the generalization ability of robust algorithms (proposed by [45]). Considering this framework for our method reveals two advantages. On the one hand, it allows us to take into account the regularizers in the generalization bound. On the other hand, the algorithmic robustness tolerates to handle non-standard learning setups like DA.

We begin with the definition of a *robust algorithm*, which stands in a standard setup where the learning sample and the test sample are drawn from the same distribution P.

Definition 3 [45] Given a Learning Sample LS of d_l examples drawn *i.i.d* from a distribution P, an algorithm $\mathcal{A}$ is $(\mathbf{M}, \boldsymbol{\epsilon}(\mathbf{LS}))$ **robust on P**, for $M \in \mathbb{N}$ and $\epsilon(.) : (X \times Y)^{d_l} \mapsto \mathbb{R}$, if $X \times Y$ can be partitioned into M disjoint sets, denoted as $\{C_i\}_{i=1}^{M}$, such that for every example $\mathbf{s}$ belonging to LS,

$$\mathbf{s}, \mathbf{u} \in C_i \Rightarrow \left|L(\mathcal{A}_{LS}, \mathbf{s}) - L(\mathcal{A}_{LS}, \mathbf{u})\right| \leq \epsilon(LS), \tag{5}$$

with $\mathcal{A}_{LS}$, the model learned from LS with $\mathcal{A}$ and $L(\cdot,\cdot)$ the loss function of $\mathcal{A}$.

Given a learning sample LS, the robustness of an algorithm, measured by the values of M and $\epsilon(LS)$, depends thus on the learning sample. Note that this definition has to be verified for every learning example. The authors have nevertheless relaxed it with the property of *pseudo-robustness*, where the condition is only required for a subset of the learning sample [45].

From Definition 3, the authors have proved the following generalization bound over the expected error on the distribution P.

Theorem 4 [45] *If a learning sample* $LS = \{(\mathbf{x}_i, y_i)\}_{i=1}^{d_l}$ *is drawn i.i.d. from a distribution* P *and if an algorithm* $\mathcal{A}$ *is* $(M, \epsilon(LS))$ *robust, then for any* $\delta > 0$*, with probability at least* $1 - \delta$*,*

$$\mathrm{err}_P(\mathcal{A}_{LS}) \;\leq\; \hat{\mathrm{err}}_P(\mathcal{A}_{LS}) + \epsilon(LS) + L^{UP}\sqrt{\frac{2M\ln 2 + 2\ln\frac{1}{\delta}}{d_l}},$$

where $\mathrm{err}_P(\mathcal{A}_{LS})$ *and* $\hat{\mathrm{err}}_P(\mathcal{A}_{LS})$ *are, respectively, the generalization and the empirical errors over* P *of the model* $\mathcal{A}_{LS}$ *learned from* LS, $L(\cdot,\cdot)$ *being upper bounded by* L^{UP}.

This bound is not proved in a DA scenario but the authors have argued that such a bound could be defined by adding a term depending on a domain divergence measure.

Generalization bound Following the previous idea, we propose to derive a bound for the target domain using the $\mathcal{H}\Delta\mathcal{H}$-distance, which is appropriate to the context our approach. First, we prove that our optimization problem ($DASF_{opt}$) is robust on the source domain and then deduce a generalization bound for the target domain.

Theorem 5 *Suppose that* (X, ρ) *is a compact metric space and* K *is a good similarity function continuous in its first argument. If the source Learning Sample* LS *is drawn i.i.d. from the source domain* P_S*, then given the hyperparameters* $\beta > 0$, $\lambda > 0$*, the landmark set* R *and a fixed pair set* $\mathcal{C}_{ST}$ *with* $B_R > 0$*, our Problem* ($DASF_{opt}$) *is* $\left(\mathbf{2M}_\eta, \frac{\mathbf{N}_\eta}{\beta\mathbf{B_R}+\lambda}\right)$ **robust on the source domain $\mathbf{P_S}$**, *where* $\eta > 0$, M_η *being the* η*-covering number*[2] *of* X *and* $N_\eta = \max_{\substack{\mathbf{x}_a,\mathbf{x}_b \sim D_S \\ \rho(\mathbf{x}_a,\mathbf{x}_b)\leq\eta}} \|{}^t\phi^R(\mathbf{x}_a) - {}^t\phi^R(\mathbf{x}_b)\|_\infty$.

[2] Meaning that X can be partitioned into M_η subsets, M_η finite, cf [45] for more details.

Proof Let (X, ρ) a compact metric space. Let $\eta > 0$, since X is compact, by the definition of the covering number, we can partition X in M_η subsets (M_η finite), such that for $\mathbf{x}_1, \mathbf{x}_2$ belonging to the same subset, we have $\rho(\mathbf{x}_1, \mathbf{x}_2) \leq \eta$. With Y divided in 2 subsets $\{\{-1\}, \{+1\}\}$ and following the proof principle of [45], we can partition $X \times Y$ in $2M_\eta$ subsets such that the points belonging to the same subset are of the same class. Given a good similarity function K continuous in its first argument, a source learning set $LS = \{(\mathbf{x}_i, y_i)\}_{i=1}^{d_l}$ drawn *i.i.d.* from P_S, a landmark set $R = \{\mathbf{x}'_j\}_{j=1}^{d_u}$, the hyperparameters $\lambda > 0$, $\beta > 0$ and a fixed pair set $\mathcal{C}_{ST}$, let $\boldsymbol{\alpha}^*$ be the optimal solution of Problem ($DASF_{opt}$). For any $\mathbf{s}_1 = (\mathbf{x}_1, y_1) \in LS$, any $\mathbf{s}_2 = (\mathbf{x}_2, y_2)$ such that $\mathbf{s}_1$ and $\mathbf{s}_2$ belong to the same subset, thus $y_1 = y_2$ and $\rho(\mathbf{x}_1, \mathbf{x}_2) \leq \eta$. Then,

$$\begin{aligned}&\left|L\left(g, (\mathbf{x}_1, y)\right) - L\left(g, (\mathbf{x}_2, y)\right)\right| \\ &\quad = \left|\left[1 - y_1 \sum_{j=1}^{d_u} \alpha_j^* K(\mathbf{x}_1, \mathbf{x}'_j)\right]_+ - \left[1 - y_1 \sum_{j=1}^{d_u} \alpha_j^* K(\mathbf{x}_2, \mathbf{x}'_j)\right]_+\right|.\end{aligned}$$

By the 1-lipschitz property of the hinge loss, the successive application of Holder inequality[3] and Lemma 2, we obtain,

$$\begin{aligned}\left|L\left(g, (\mathbf{x}_1, y)\right) - L\left(g, (\mathbf{x}_2, y)\right)\right| &\leq \|\boldsymbol{\alpha}^*\|_1 \|{}^t\phi^R(\mathbf{x}_1) - {}^t\phi^R(\mathbf{x}_2)\|_\infty \\ &\leq \|\boldsymbol{\alpha}^*\|_1 \max_{\substack{\mathbf{x}_a, \mathbf{x}_b \sim D_S \\ \rho(\mathbf{x}_a, \mathbf{x}_b) \leq \eta}} \left\{\|{}^t\phi^R(\mathbf{x}_a) - {}^t\phi^R(\mathbf{x}_b)\|_\infty\right\} \\ &\leq \frac{N_\eta}{\beta B_R + \lambda},\end{aligned}$$

with $N_\eta = \max_{\substack{\mathbf{x}_a, \mathbf{x}_b \sim D_S \\ \rho(\mathbf{x}_a, \mathbf{x}_b) \leq \eta}} \left\{\|{}^t\phi^R(\mathbf{x}_a) - {}^t\phi^R(\mathbf{x}_b)\|_\infty\right\}$, which is finite by the continuity of K in its first argument and the definition of covering number. Then, the algorithm associated with Problem ($DASF_{opt}$) is $\left(2M_\eta, \frac{N_\eta}{\beta B_R + \lambda}\right)$ robust on P_S. □

In our case, the hinge loss $\left[1 - y\sum_{j=1}^{d_u} \alpha_j K(., \mathbf{x}'_j)\right]_+$ is upper bounded by a constant L^{UP} and to lighten the notations we suppose $L^{UP} = 1$ (which is not true in general but can be easily obtained by a normalization step). Then, we directly derive the following generalization bound over the expected **source error** from Theorem 4.

Theorem 6 *With the same notations of Theorem 5, for every h in the hypothesis class $\mathcal{H}$ of SF-classifiers and for any $\delta > 0$, with probability at least $1 - \delta$,*

$$\mathrm{err}_S(h) \leq \hat{\mathrm{err}}_S(h) + \frac{N_\eta}{\beta B_R + \lambda} + \sqrt{\frac{4M_\eta \ln 2 + 2 \ln \frac{1}{\delta}}{d_l}}.$$

Proof From Theorem 5, Problem ($DASF_{opt}$) is $\left(2M_\eta, \frac{N_\eta}{\beta B_R + \lambda}\right)$ robust on the source domain P_S, the result is then obtained from Theorem 4. □

From this result, we can now derive a generalization bound for our unsupervised domain adaptation approach based on good similarity functions.

[3] Holder inequality: $\|\mathbf{uv}\|_1 \leq \|\mathbf{u}\|_p \|\mathbf{v}\|_q$, with $1 \leq p,\ q \leq \infty$ and $1/p + 1/q = 1$.

Theorem 7 *If $LS = \{(\mathbf{x}_i, y_i)\}_{i=1}^{d_l}$ is drawn i.i.d. from the source domain P_S, for every h in the hypothesis class $\mathcal{H}$ of SF-classifiers, for any $\delta > 0$, with probability at least $1-\delta$,*

$$\mathrm{err}_T(h) \;\le\; \hat{\mathrm{err}}_S(h) + \frac{N_\eta}{\beta B_R + \lambda} + \sqrt{\frac{4M_\eta \ln 2 + 2\ln\frac{1}{\delta}}{d_l}} + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(D_S, D_T) + \nu,$$

where ν is the joint error over the domains, $d_{\mathcal{H}\Delta\mathcal{H}}(D_S, D_T)$ is the $\mathcal{H}\Delta\mathcal{H}$-distance between the marginal distributions.

Proof It comes directly from the application of Theorems 1 and 6. □

In this bound, $d_{\mathcal{H}\Delta\mathcal{H}}$ measures the domain divergence and ν the adaptation capability of the hypothesis class. The constant $\frac{N_\eta}{\beta B_R+\lambda}$ clearly depends on the regularizers and on the value N_η that can be as small as wished, by choosing a small η and by the continuity of the similarity function $K(.,.)$ in its first argument, implying then an increase of M_η. The term with M_η converges in $O(1/\sqrt{d_l})$ and $\hat{\mathrm{err}}_S$ is the empirical error over the source sample. We can remark that with small values for β and λ, or B_R (indicating far domains), the process will need more examples to be reliable. In our method, the terms $d_{\mathcal{H}\Delta\mathcal{H}}$ and $\hat{\mathrm{err}}_S$ are actually decreased by solving our Problem ($DASF_{opt}$). In the next part, we present how to select the regularization parameters to keep these two terms low and we also introduce an heuristic aiming at decreasing an estimate of ν.

4.3 Reverse classifier and validation

A crucial point is the choice of the different hyperparameters λ, β, $\mathcal{C}_{ST}$ of our method. In a transfer learning context, [51] have proposed a *Transfer Cross-Validation* method for selecting the best parameters. This principle uses a *reverse validation* approach based on a so-called *reverse classifier* evaluated on the source domain. We propose to follow a similar reverse validation procedure.

Given k-folds on the source labeled sample and a learning algorithm, $k-1$ labeled folds are used as labeled examples for learning a classifier h'. Then, using the same algorithm, a reverse classifier h'^r is learned from a sample constituted by the union of the target sample $\{(\mathbf{x}, h'(\mathbf{x}))\}_{\mathbf{x}\in TS}$ self-labeled by h' and a given target labeled set. Finally, the reverse classifier h'^r is evaluated on the last k^{th} fold of the source labeled sample. Note that in its original definition, this method relies on a projection space defined by a kernel and uses some few target labels.

Since, we may consider non-PSD, non-symmetric similarity functions, and no label on the target domain, we then make a little adaptation: We perform the reverse validation directly in the projection space ϕ^R and we learn the reverse classifier only with the self-labeled target sample (see Fig. 4). The justification of this choice comes from the fact that if the domains are sufficiently close and related, then such a reverse classifier must be also efficient for the source task [14]. In other words, in the projection space, it is possible to pass from one problem to another. Recall that we do not have any information on the target labels. We then define our reverse classifier h^r as the best SF-classifier learned with SF_{opt}—in the current ϕ^R-space—from the target sample $\{(\mathbf{x}, h(\mathbf{x}))\}_{\mathbf{x}\in TS}$, self-labeled by the classifier h learned with our Problem ($DASF_{opt}$).

In summary, given k-folds on the source labeled sample ($LS = \cup_{i=1}^{k} LS_i$), a classifier h is learned from $k-1$ labeled folds and the unlabeled target sample by solving Problem ($DASF_{opt}$) and we evaluate the associated reverse classifier h^r on the last k^{th} fold. Its empirical source error corresponds to the mean of the error over the k-folds,

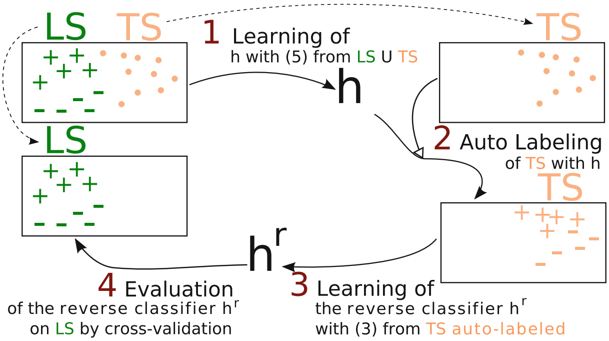

LS
TS
1 Learning of
h with (5) from LS U TS
h
TS
2 Auto Labeling
of TS with h
TS
LS
h^r
4 Evaluation
of the reverse classifier h^r
on LS by cross-validation
3 Learning of
the reverse classifier h^r
with (3) from TS auto-labeled

Fig. 4 The reverse validation process in the ϕ^R-space. Step **1**: *Learn* the classifier h with the Problem ($DASF_{opt}$). Step **2**: *Auto-label* the target sample with h. Step **3**: *Learn* the reverse classifier h^r on the auto-labeled target sample with the Problem (SF_{opt}). Step **4**: *Evaluate* h^r on the source sample (with a k-folds process) (color figure online)

$$\hat{\text{err}}_S(h^r) = \frac{1}{k}\sum_{i=1}^{k} \hat{\text{err}}_{LS_i}(h^r).$$

In Sect. 2, Theorem 1 suggests that one solution for DA is to minimize the three terms of the DA bound,[4] which are also present in our generalization bound in Theorem 7. Our Problem ($DASF_{opt}$) aims at minimizing the first two terms but does not consider at the moment the last term ν corresponding to the ideal joint classifier error and defined by $\nu = \text{err}_S(h^*) + \text{err}_T(h^*)$ with $h^* = \text{argmin}_{h\in\mathcal{H}}(\text{err}_S(h) + \text{err}_T(h))$. This hypothesis, unknown in general, measures the adaptation ability of the classifier. We propose then to use an estimation of ν for selecting the relevant hyperparameters. However, due to the absence of target labels, we are not able to compute or estimate this ideal hypothesis. Since h^* is clearly related to the capability to pass from one domain to another, we estimate it by the reverse classifier h^r. For this purpose, at each cross-validation step, we divide the self-labeled target sample $\{(\mathbf{x}, h(\mathbf{x}))\}_{\mathbf{x}\in TS}$ into two parts: one is used to learn h^r and then h^r is evaluated on the second to give an estimate of the target error, the evaluation of h^r on the current source fold giving an estimation of the source error. Then, $\hat{\text{err}}_S(h^r)$ (resp. $\hat{\text{err}}_T(h^r)$) corresponds to the mean over the k-folds of the estimation of the source error (resp. the target error) of h^r. We then consider the empirical estimation of ν defined by,

$$\hat{\nu} = \hat{\text{err}}_S(h^r) + \hat{\text{err}}_T(h^r), \tag{6}$$

where $\hat{\text{err}}_T(h^r)$ is evaluated over the self-labeled target sample. Motivated by the minimization of the DA bound, we finally select the parameters leading to the minimal $\hat{\nu}$.

5 An iterative reweighting: a way to lighten the search of the projection space

The constitution of the set $\mathcal{C}_{ST}$ is difficult *a priori* since we have no information on the target labels. Moreover, the set of relevant pairs allowing a good adaptation is generally dependent on the task at hand and testing all the possible pair sets is clearly intractable.[5] In order to

[4] The DA bound: $\forall h \in \mathcal{H},\ \text{err}_T(h) \ \leq\ \text{err}_S(h) + \frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(D_S, D_T) + \nu$.

[5] This intractability has been confirmed empirically: in such a case, none of the experiments have led to a result in a reasonable amount of time.

tackle this problem, we present an iterative approach based both on a selection of a limited number of pairs and on a reweighting scheme of the similarities keeping the distributions close. We finally present a stopping criterion using the empirical estimation of the ideal joint error.

5.1 Selecting the pairs of $\mathcal{C}_{ST}$

We propose to construct pairs from two subsets of the two samples provided to the algorithm $U_S \subseteq LS_{|X}$ and $U_T \subseteq TS$ of equal size. We select them, at a given iteration l, according to the *reverse model* g^r_{l-1} associated with the reverse classifier h^r_{l-1} computed in the previous iteration. They correspond to the examples on which this model is highly or weakly confident on the labels. Let $\delta^H_S, \delta^H_T, \delta^L_S, \delta^L_T$ be a set of positive parameters, U_S and U_T are defined as follows such that $|U_S| = |U_T| \leq N$,

$$\left\{\begin{array}{l} U_S = \big\{\mathbf{x} \in LS_{|X}\colon\ |g^r_l(\mathbf{x})| > \delta^H_S \text{ OR } |g^r_l(\mathbf{x})| < \delta^L_S\big\},\\ U_T = \big\{\mathbf{x} \in TS\ :\ |g^r_l(\mathbf{x})| > \delta^H_T \text{ OR } |g^r_l(\mathbf{x})| < \delta^L_T\big\}. \end{array}\right.$$

In practice, we use these two sets for building the matching $\mathcal{C}_{ST} \subset U_S \times U_T$ from U_S and U_T. We look for a bipartite matching minimizing the Euclidean distance in the new projection ϕ^R_l-space associated with an iteration l in the iterative process (described in the next Sect. 5.2). This can be done by solving the following problem. Note that in the particular case of bipartite matching, it can be achieved in polynomial time by linear programming for example.

$$\left\{\begin{array}{l} \displaystyle\min_{\substack{\chi_{st}\\ 1\leq s\leq |U_S|\\ 1\leq t\leq |U_T|}} \sum_{(\mathbf{x}_s,\mathbf{x}_t)\in U_S\times U_T} \chi_{st}\, \|\phi^R_l(\mathbf{x}_s) - \phi^R_l(\mathbf{x}_t)\|_2^2\\ \text{s.t.: } \forall(\mathbf{x}_s,\mathbf{x}_t) \in U_S \times U_T, \qquad \chi_{st} \in [0,1],\\ \qquad \displaystyle\forall \mathbf{x}_s \in U_S,\ \sum_{\mathbf{x}_t\in U_T} \chi_{st} = 1,\\ \qquad \displaystyle\forall \mathbf{x}_t \in U_T,\ \sum_{\mathbf{x}_s\in U_S} \chi_{st} \leq 1. \end{array}\right. \tag{7}$$

Then, $\mathcal{C}_{ST}$ corresponds to the pairs of $U_S \times U_T$ such that $\chi_{st} = 1$.

In our experiments, we limit the size of the subsets U_S and U_T to small[6] N to build efficiently this bipartite matching, since it has to be done many times according to the different iterations and cross-validation. This is not a too restrictive heuristic since the notion of pseudo-robustness of [45] does not require to consider all the points. In this case, the values $\delta^H_S, \delta^H_T, \delta^L_S, \delta^L_T$ correspond to the ones allowing us to select the first N elements of each type.

5.2 A new projection space by iterative reweighting

The landmarks selected[7] by solving Problem ($DASF_{opt}$) define a projection space where the two distributions tend to be close. We propose to reuse their weights α_j to force the new projection space to move closer the distributions. Indeed, we reweight the similarity function according to $\boldsymbol{\alpha}$. Suppose at a given iteration l, with a similarity function K_l, we obtain

[6] In our experiments, we take arbitrarily $N \leq 30$.

[7] i.e. those with a non-null weight α_j.

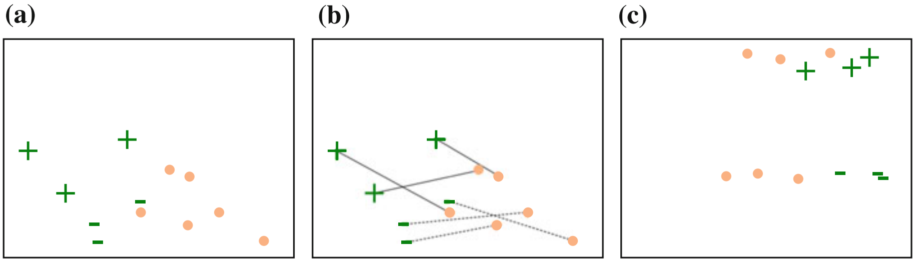


Fig. 5 An iteration of DASF. The source points are in (*dark*) *green* (pos. +, neg. −), the unlabeled target ones are (*light*) *orange circles*. **a** The points in a ϕ_l^R-space, $|R| = 2$. **b** The associated bipartite matching CST : a line corresponds to a pair $(\mathbf{x}_s, \mathbf{x}_t)$. **c** The points in the new ϕ_{l+1}^R-space after the reweighting procedure (color figure online)

new weights $\boldsymbol{\alpha}^l$. Then, we propose to define K_{l+1} by reweighting[8] K_l conditionally to each landmark of R such that

$$\forall \mathbf{x}'_j \in R,\ \ K_{l+1}(\mathbf{x}, \mathbf{x}'_j) = \alpha^l_j K_l(\mathbf{x}, \mathbf{x}'_j).$$

It can be seen as a kind of contraction of the space to keep the empirical $\mathcal{H}\Delta\mathcal{H}$-distance between the marginal distributions low. Indeed, in this new ϕ_{l+1}^R-space defined by K_{l+1}, the points of each pair of $\mathcal{C}_{ST}$ are naturally close since, by construction, our regularizer used at iteration l corresponds exactly to minimize their L_1-distance in the ϕ_{l+1}^R-space. Indeed, we have

$$\forall (\mathbf{x}_s, \mathbf{x}_t) \in \mathcal{C}_{ST},\ \ \|{}^t\phi_{l+1}^R(\mathbf{x}_s) - {}^t\phi_{l+1}^R(\mathbf{x}_t)\|_1 = \|{}^t(\phi_l^R(\mathbf{x}_s) - {}^t\phi_l^R(\mathbf{x}_t))\,\mathrm{diag}(\boldsymbol{\alpha}^l)\|_1.$$

An illustration of this procedure is provided on Fig. 5. We then iterate the process at iteration $l+1$ in the new ϕ_{l+1}^R-space.

The possible reweightings are related to the different hyperparameters $\delta_{S/T}^{H/L}$ (linked to $\mathcal{C}_{ST}$) and $\lambda,\ \beta$ of Problem ($DASF_{opt}$) that are selected according to reverse validation. Recall that, since we are not interested in using valid kernels, we do not have to keep any notion of symmetry or positive semi-definiteness for K_{l+1}.

However, our normalization remains valid only if the new similarity function is still good on the source domain. We can *empirically* estimate this goodness by evaluating ϵ, γ and τ of Definition 1 on LS. In practice, the empirical $\hat{\tau}$ corresponds to the number of landmarks selected by the algorithm. Therefore, we evaluate the best empirical $(\hat{\epsilon}, \hat{\gamma}, \hat{\tau})$-guarantee by

$$\hat{\gamma} = \begin{cases} \gamma_{\max} & \text{if } \operatorname{argmax}_{\gamma_{\max}>0} \left\{ \forall (\mathbf{x}_i, y_i) \in LS,\ \frac{y_i}{d_u} \sum_{j=1}^{d_u} K(\mathbf{x}_i, \mathbf{x}'_j) \geq \gamma_{\max} \right\} \text{ exists,} \\ 0 & \text{otherwise.} \end{cases}$$

$$\hat{\epsilon} = \begin{cases} 0 & \text{if } \hat{\gamma} > 0, \\ \dfrac{\left| \left\{ (\mathbf{x}_i, y_i) \in LS \ :\ \dfrac{y_i}{d_u} \displaystyle\sum_{j=1}^{d_u} K(\mathbf{x}_i, \mathbf{x}'_j) < 0 \right\} \right|}{|LS|} & \text{otherwise.} \end{cases}$$

[8] Eventually normalized to ensure $K_{l+1} \in [-1, 1]$.

In fact, we pay attention to keep only those that offer the best $(\hat{\epsilon}, \hat{\gamma}, \hat{\tau})$-guarantee ensuring a sufficiently good similarity. Concretely, the higher $\hat{\gamma}$ and the lower $\hat{\epsilon}$, the better the guarantee. Note that a bad similarity would lead to a dramatic increase of the expected source error and thus would not be selected by the reverse validation process.

5.3 Stopping criterion

We consider here the estimated joint error $\hat{\nu}$ (6) related to the adaptation capability in the current space. Controlling the decreasing of this term during the iterative process can provide a nice way to stop the algorithm. Following Sect. 4.3, at a given iteration l, this term is defined by $\hat{\nu}_l = \hat{\text{err}}_S(h_l^r) + \hat{\text{err}}_T(h_l^r)$, where h_l^r is the reverse classifier associated with h_l learned at iteration l. An increasing of $\hat{\nu}_l$ between two iterations means that the new projection space is no longer relevant and the current one must be preferred.

Then, our process stops at iteration l when the next $\hat{\nu}_{l+1}$ has reached a convergence point or has increased significantly. This criterion allows us to ensure the algorithm stops since the joint error is positive and bounded by 0. The global iterative algorithm (named DASF) is described in Algorithm 1.

Algorithm 1 DASF: Domain adaptation with similarity function

```
input similarity function $K$, landmark set $R$, source sample $LS$, and target sample $TS$
output classifier $h_{DASF}$
  $h_0(\cdot) \leftarrow \text{sign}\left[\frac{1}{|R|}\sum_{j=1}^{|R|} K(\cdot, \mathbf{x}'_j)\right]$
  $K_1 \leftarrow K$
  $l \leftarrow 1$
  while The stopping criterion is not verified do
    Select $U_S \subseteq LS_{|X}$, $U_T \subseteq TS$ with $h^r_{l-1}$
    $\mathcal{C}_{ST} \leftarrow$ Solve Problem (7)
    $\boldsymbol{\alpha}^l \leftarrow$ Solve Problem ($DASF_{opt}$) with $K_l$ and $\mathcal{C}_{ST}$
    $K_{l+1} \leftarrow$ Update $K_l$ according to $\boldsymbol{\alpha}^l$
    Update $R$
    $l++$
  end while
  return $h_{DASF}(\cdot) = \text{sign}\left[\sum_{x'_j \in R} \alpha^l_j K_l(\cdot, \mathbf{x}'_j)\right]$
```

5.4 Complexity

In practice, hyperparameters are selected according to a grid search, which has to be done at each iteration. It is thus very heavy and it is a clear disadvantage of our method. The global complexity at each iteration corresponds to solving three different linear programs: Problem ($DASF_{opt}$), the building of the pair set C_{ST} (Problem (7)) and computing the reverse classifier (with Problem (SF_{opt})) for each parameter set. Solving a linear program is in general costly $\sim \mathcal{O}(n^{3.5}L)$ [25] (or $\sim \mathcal{O}(n^3 L)$ with approximated solutions [49]) for a system with n variables that can be encoded in L bits. However, the optimization problems we consider are a bit sparse in the sense that all the constraints do not involve all the variables at the same time that makes the problem faster to solve. Moreover, it is important to notice that the iteration process combined with the reverse validation allows us to lighten the search of the parameters.

6 Considering a few target labels: semi-supervised DASF

So far, we have not considered any target label data. However, for some real applications, it is reasonable to assume that a few target labels are available, notably in multimedia indexing tasks. This complimentary information can be very useful for constraining the search of the classifier as we will see in the experimental section.

We propose to extend our approach to take into account the target labels by following the principle proposed in [9]. It considers a linear combination of source and target labeled learning data. In this case, the learning *Labeled Sample* $LS = (LS_{P_S}, LS_{P_T})$ is composed of a sample $LS_{P_S} = \{(\mathbf{x}_{i^S}, y_{i^S})\}_{i^S=1}^{d_l^S}$ of d_l^S labeled source examples *i.i.d.* from P_S and a sample $LS_{P_T} = \{(\mathbf{x}_{i^T}, y_{i^T})\}_{i^T=1}^{d_l^T}$ of d_l^T labeled target instances *i.i.d.* from P_T. Let $\theta \in [0,1]$ such that $d_l^T = \theta d_l$ and $d_l^S = (1-\theta) d_l$ ensuring LS has $d_l = d_l^T + d_l^S$ labeled instances. Recall that we aim at minimizing the target expected error err_T with d_l^T small regarding to d_l^S, that is, with few target labels. In this context, as mentioned in [9], minimizing directly the target empirical error $\hat{\mathrm{err}}_T$ from LS_{P_T} does not seem to be the best solution since this sample is not sufficiently representative of the target distribution.

Thus, following [9], we minimize a convex combination of the source and target empirical errors,

$$\hat{\mathrm{err}}_\kappa(h) = \kappa\, \hat{\mathrm{err}}_T(h) + (1-\kappa)\, \hat{\mathrm{err}}_S(h), \tag{8}$$

for some $\kappa \in [0,1]$, $\mathrm{err}_\kappa(h) = \kappa\ \mathrm{err}_T(h) + (1-\kappa)\,\mathrm{err}_S(h)$ being the associated weighted expected error.

This leads us to an adaption of our previous optimization Problem ($DASF_{opt}$), with some target labels. Given $LS = (LS_{P_S}, LS_{P_T})$ a sample of d_l instances, $\mathcal{C}_{ST} \subset LS_{P_{S|X}} \times TS$ a pair set and $\kappa \in [0,1]$, we propose the following minimization Problem ($SSDASF_{opt}$). The global iterative algorithm (named SSDASF) is described in Algorithm 2.

$$\left\{\begin{array}{l} \min\limits_{\boldsymbol{\alpha}} (1-\kappa)\dfrac{1}{d_l^S}\displaystyle\sum_{i^S=1}^{d_l^S} L\big(g, (\mathbf{x}_{i^S}, y_{i^S})\big) + \kappa\dfrac{1}{d_l^T}\sum_{i^T=1}^{d_l^T} L\big(g, (\mathbf{x}_{i^T}, y_{i^T})\big) + \lambda\|\boldsymbol{\alpha}\|_1 \\ \quad + (1-\kappa)\,\beta \displaystyle\sum_{(\mathbf{x}_s,\mathbf{x}_t)\in\mathcal{C}_{ST}} \left\| \big({}^t\phi^R(\mathbf{x}_s) - {}^t\phi^R(\mathbf{x}_t)\big)\,\mathrm{diag}(\boldsymbol{\alpha})\right\|_1, \\ \text{with } L\big(g,(\mathbf{x}_i,y_i)\big) = \Big[1 - y_i g(\mathbf{x}_i)\Big]_+ \quad \text{and} \quad g(\mathbf{x}_i) = \displaystyle\sum_{j=1}^{d_u} \alpha_j K(\mathbf{x}_i, \mathbf{x}'_j). \end{array}\right. \tag{$SSDASF_{opt}$}$$

Whereas our previous Problem ($DASF_{opt}$) focuses only on the minimization of the source empirical error, this optimization problem minimizes the convex combination of the source and target empirical errors (Eq. (8)). We can in fact make some connections showing that ($SSDASF_{opt}$) can be seen as a generalization of some previous problems. Firstly, when $\kappa = 0$, no target labeled data are used and we move back to our standard DASF algorithm with Problem ($DASF_{opt}$). Secondly, when $\kappa = 1$, we stand in a usual supervised framework where the learning and test samples are drawn according to the target domain P_T. Then, Theorem 4 of [45] about robustness can be proved on P_T.

Algorithm 2 SSDASF: Semi-supervised domain adaptation with similarity function

```
input  Similarity function $K$, landmark set $R$, labeled sample $LS=(LS_{P_S}, LS_{P_T})$, target sample $TS$
output  classifier $h_{SSDASF}$
  $h_0(\cdot) \leftarrow \text{sign}\left[\frac{1}{|R|}\sum_{j=1}^{|R|} K(\cdot, \mathbf{x}'_j)\right]$
  $K_1 \leftarrow K$
  $l \leftarrow 1$
  while  The stopping criterion is not verified  do
    Select $U_S \subseteq LS_{P_S|X}$, $U_T \subseteq TS$ with $h^r_{l-1}$
    $\mathcal{C}_{ST} \leftarrow$ Solve (7)
    $\boldsymbol{\alpha}^l \leftarrow$ Solve ($SSDASF_{opt}$) with $K_l$ and $\mathcal{C}_{ST}$
    $K_{l+1} \leftarrow$ Update $K_l$ according to $\boldsymbol{\alpha}^l$
    Update $R$
    $l++$
  end while
  return $h_{SSDASF}(\cdot) = \text{sign}\left[\sum_{x'_j \in R} \alpha^l_j K_l(\cdot, \mathbf{x}'_j)\right]$
```

We can also derive for our new Problem ($SSDASF_{opt}$) a generalization bound. First, we adapt the preceding result on sparsity analysis.

Lemma 3 *For any hyperparameters $\lambda > 0$ and $\beta > 0$, $\kappa \in [0, 1]$ and for any pair set $\mathcal{C}_{ST}$, let $B_R = \min_{\mathbf{x}'_j \in R} \left\{ \max_{(\mathbf{x}_s, \mathbf{x}_t) \in \mathcal{C}_{ST}} |K(\mathbf{x}_s, \mathbf{x}'_j) - K(\mathbf{x}_t, \mathbf{x}'_j)| \right\}$. If $\boldsymbol{\alpha}^*$ denotes the optimal solution of our Problem ($SSDASF_{opt}$), then we have $\|\boldsymbol{\alpha}^*\|_1 \leq \dfrac{1}{(1-\kappa)\beta B_R + \lambda}$.*

Proof We use the same proof process as in Lemma 2. □

This result suggests that when some additional target labels are used, that is, $(1 - \kappa) < 1$, the induced model is less sparse than the approach with no target label. From this result, we can now provide our generalization bound combining source and target labels.

Theorem 8 *Let $\theta \in [0, 1]$, $\kappa \in [0, 1]$, and LS be a labeled learning sample of size d_l constituted of θd_l instances i.i.d. from target distribution P_T and $(1 - \theta)d_l$ examples i.i.d. from source distribution P_S. Let $\eta', \eta > 0$, with $M = \max(M_\eta, M_{\eta'})$ a covering number for X, $\beta > 0$, $\lambda > 0$ and $B_R > 0$. For all $h \in \mathcal{H}$ minimizing the empirical error by Problem ($SSDASF_{opt}$), if $h^* = \text{argmin}_{h' \in \mathcal{H}}\{\text{err}_T(h')/\hat{\text{err}}_\kappa(h) \leq \hat{\text{err}}_\kappa(h')\}$, then with probability at least $1 - \delta$,*

$$\text{err}_T(h) \leq \text{err}_T(h^*) + \sqrt{\frac{\kappa^2}{\theta} + \frac{(1-\kappa)^2}{1-\theta}}\sqrt{\frac{\ln\frac{4}{\delta}}{2d_l}} + \frac{\kappa(N^T_{\eta'} - N^S_\eta) + N^S_\eta}{(1-\kappa)\beta B_R + \lambda}$$
$$+ \sqrt{\frac{4M\ln 2 + 2\ln\frac{4}{\delta}}{d_l}}\left(\frac{\kappa}{\sqrt{\theta}} + \frac{1-\kappa}{\sqrt{1-\theta}}\right) + 2(1-\kappa)\left(\frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(D_S, D_T) + \nu\right), \tag{9}$$

where $B_R = \min_{\mathbf{x}'_j \in R} \left\{ \max_{(\mathbf{x}_s, \mathbf{x}_t) \in \mathcal{C}_{ST}} |K(\mathbf{x}_s, \mathbf{x}'_j) - K(\mathbf{x}_t, \mathbf{x}'_j)| \right\}$, and

$$N^S_\eta = \max_{\substack{\mathbf{x}_a, \mathbf{x}_b \sim D_S \\ \rho(\mathbf{x}_a, \mathbf{x}_b) \leq \eta}} \|{}^t\phi^R(\mathbf{x}_a) - {}^t\phi^R(\mathbf{x}_b)\|_\infty, \quad N^T_{\eta'} = \max_{\substack{\mathbf{x}_a, \mathbf{x}_b \sim D_T \\ \rho(\mathbf{x}_a, \mathbf{x}_b) \leq \eta'}} \|{}^t\phi^R(\mathbf{x}_a) - {}^t\phi^R(\mathbf{x}_b)\|_\infty.$$

Proof Deferred to "Appendix". □

Note that, when $\kappa = 0$, *i.e.* when we ignore target data, the bound is similar to that of Theorem 7. On the other hand, when $\kappa = 1$, *i.e.* we do not use source information, the bound becomes a classical standard learning bound with robustness in a supervised setting with only target data.

In order to illustrate some properties of that bound, we provide an analysis of its behavior, following the same principle as in [9]. In its present form, the bound is a bit difficult to analyze and in order to simplify our study, we make some little assumptions. First, we assume $\kappa \in [0, a]$, where a is a positive number tending to 1 such that $a < 1$. This allows us to bound the following term

$$\frac{\kappa(N^T_{\eta'} - N^S_\eta) + N^S_\eta}{(1-\kappa)\beta B_R + \lambda} < \frac{\kappa(N^T_{\eta'} - N^S_\eta) + N^S_\eta}{(1-a)\beta B_R + \lambda}.$$

Then, we define

$$\begin{aligned} A &= \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(D_S, D_T) + \nu, \\ B &= (1-a)\beta B_R + \lambda, \\ C &= \sqrt{\frac{4M \ln 2 + 2 \ln \frac{4}{\delta}}{d_l}} \left(\frac{1}{\sqrt{\theta}} - \frac{1}{\sqrt{1-\theta}} \right) + \frac{(N^T_{\eta'} - N^S_\eta)}{B} - 2A, \end{aligned}$$

and D as the remaining constant terms. The bound (9) can then be rewritten as,

$$f(\kappa) = \sqrt{\frac{\kappa^2}{\theta} + \frac{(1-\kappa)^2}{1-\theta}} \sqrt{\frac{\ln \frac{4}{\delta}}{2d_l}} + C\kappa + D.$$

With $d^T_l = \theta d_l$ and $d^S_l = (1-\theta) d_l$, the general form of the highest root r of the derivative is

$$r = \theta \left(1 + \frac{1-\theta}{\sqrt{\frac{2 \ln \frac{4}{\delta}}{d_l C^2} - \theta(1-\theta)}} \right) = \frac{d^T_l}{d^S_l + d^T_l} \left(1 + \frac{d^S_l}{\sqrt{\frac{2 \ln \frac{4}{\delta} (d^S_l + d^T_l)}{C^2} - d^S_l d^T_l}} \right).$$

To simplify the analysis, we assume $N^T_{\eta'} = N^S_\eta - B \sqrt{\frac{4M \ln 2 + 2 \ln \frac{4}{\delta}}{d_l}} \left(\frac{1}{\sqrt{\theta}} - \frac{1}{\sqrt{1-\theta}} \right) > 0$ which can be obtained by choosing appropriate η, η' with d_l sufficiently large and/or by considering an upper bound of the function $f(\kappa)$ verifying this equality, we must then have $d^T_l < \frac{\ln \frac{4}{\delta}}{(1-\theta)2A^2}$ for r to be valid. Thus the optimal value is defined as follows,

$$\kappa = \begin{cases} a & \text{if } d^T_l \geq \dfrac{\ln \frac{4}{\delta}}{(1-\theta)2A^2}, \\ \min\{a, r\} & \text{if } d^T_l < \dfrac{\ln \frac{4}{\delta}}{(1-\theta)2A^2}. \end{cases}$$

An interesting remark is that when $A = 0$, the bound suggests $\kappa = \theta$, *i.e.* when the two domains are indistinguishable κ has to follow the repartition defined by the training sample. If we have no target label (*i.e.* $d^T_l = 0$) the bound suggests $\kappa = 0$. If we have only target labels ($d^S_l = 0$) or if d^T_l is large, κ must be chosen as 1, which is consistent with our framework. When the domains are very different, *i.e.* A is maximum, the bound says that κ tends to be 1, *i.e.* it is better to rely only on target labeled points. With other values of $N^T_{\eta'}/N^S_\eta$, this tendency is also confirmed with our value B_R. Indeed, when it is high, *i.e.* domains are far in the current representation, it seems better to put more weights on labeled target points. We can also see from B that when our hyperparameters are small, *i.e.* when we give a small importance to the decreasing of the distance or when we allow complex models, we should rather focus on target points with κ high, while a smaller value of κ can be better in the opposite case. As a conclusion, this analysis is close to the one provided in [9] but has the advantage to take into account our regularizers for explaining the behavior of the approach.

7 Experiments

In this section, we evaluate our approach DASF and its semi-supervised extension SSDASF on a synthetic toy problem and on a real image annotation task. We first present in Sect. 7.1 the similarity function used. More precisely, we propose an heuristic procedure to modify a priori the projection space for obtaining a relevant similarity, which is non-symmetric and non-PSD, for domain adaptation. Then in Sect. 7.2, we introduce the general setup for all our experiments. The results for the synthetic datasets are given in Sect. 7.3 and those for the image classification problems are presented in Sect. 7.4.

7.1 Similarity function

We propose here to introduce an intuitive preprocess to design a similarity function potentially non-PSD, non-symmetric. According to the theoretical result of DA of [9] (Theorem 1), the learned classifier should perform well on the target domain and also on the source one. Thus, we aim at facilitating the adaptation to the target domain in order to link the source and target domains by considering information from both of them. Concretely, we build our new similarity function K_{ST} by renormalizing a given similarity function K relatively to the unlabeled sample $ST = LS_{|X} \cup TS$. Our choice is clearly heuristic and our aim is just to evaluate the interest of renormalizing a similarity for DA problems. Recall that, from Definition 1, a similarity must be good relatively to a set of reasonable points R. We actually propose to renormalize the similarities to these points: We perform a specific normalization for each instance $\mathbf{x}'_j \in R$. The idea is to apply a scaling to mean zero and standard deviation one for the similarities of the instances of ST to each $\mathbf{x}'_j$. Given a similarity function K, which verifies the Definition 1, our normalized similarity function K_{ST} is defined by

$$\forall \mathbf{x}'_j \in R,\ K_{ST}(.,\mathbf{x}'_j) = \begin{cases} \dfrac{K(.,\mathbf{x}'_j) - \mu_{\mathbf{x}'_j}}{\sigma_{\mathbf{x}'_j}} & \text{if } -1 \leq \dfrac{K(.,\mathbf{x}'_j) - \hat{\mu}_{\mathbf{x}'_j}}{\hat{\sigma}_{\mathbf{x}'_j}} \leq \quad 1, \\ -1 & \text{if } \qquad \dfrac{K(.,\mathbf{x}'_j) - \hat{\mu}_{\mathbf{x}'_j}}{\hat{\sigma}_{\mathbf{x}'_j}} \leq -1, \\ 1 & \text{if } \quad 1 \leq \dfrac{K(.,\mathbf{x}'_j) - \hat{\mu}_{\mathbf{x}'_j}}{\hat{\sigma}_{\mathbf{x}'_j}} \qquad , \end{cases} \quad (SSDASF_{opt})$$

where $\hat{\mu}_{\mathbf{x}'_j}$ is the empirical mean of similarities to $\mathbf{x}'_j$ over ST,

$$\forall \mathbf{x}'_j \in R,\ \hat{\mu}_{\mathbf{x}'_j} = \frac{1}{|ST|} \sum_{\mathbf{x} \in ST} K(\mathbf{x}, \mathbf{x}'_j),$$

and $\hat{\sigma}_{\mathbf{x}'_j}$ is the empirical unbiased estimate of the standard deviation associated with $\hat{\mu}_{\mathbf{x}'_j}$,

$$\forall \mathbf{x}'_j \in R,\ \hat{\sigma}_{\mathbf{x}'_j} = \sqrt{\frac{1}{|ST|-1}\sum_{\mathbf{x}\in ST}\left(K(\mathbf{x},\mathbf{x}'_j) - \hat{\mu}_{\mathbf{x}'_j}\right)^2}.$$

By construction, the similarity K_{ST} is then non-symmetric and non-PSD.

For all experiments, we take as similarity function K a Gaussian kernel,

$$K(\mathbf{x},\mathbf{x}') = \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}'\|_2^2}{D^2}\right). \qquad (SSDASF_{opt})$$

However, depending on the samples, the non-symmetric, non-PSD similarity K_{ST} does not always offer better (ϵ,γ,τ)-good guarantees than the Gaussian kernel. In the following, we only indicate the similarity which leads to the best results. Those obtained with K_{ST} are pointed out with a*, as we will see they correspond generally to harder tasks.

7.2 General experimental setup

We compare our algorithm DASF with a classical SVM learned only from the source domain, the semi-supervised Transductive SVM [43] (TSVM) and the DA method DASVM [14]. We take a classical Gaussian kernel ($SSDASF_{opt}$) for these three methods to facilitate the comparison. We use the SVM-light library [31] with parameters tuned by cross-validation on the source data for SVM and TSVM. DASVM is implemented with the LibSVM library [16]. The parameters of DASVM and DASF are tuned according to a grid search by reverse validation. We also measure the behavior of a SF-classifier trained only from the source domain. For DASF and SF, the landmarks are taken from the labeled source sample. Following Equation (1), we assess an estimation of the $\mathcal{H}\Delta\mathcal{H}$-distance $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}$ between the two marginal distributions by learning a SF-classifier to separate source from target samples: A small value—near 0—indicates close distributions while a larger value—near 2—indicates a hard DA task. We also observe the influence of the hyperparameters λ (with fixed β) and β (with fixed λ) on our method DASF. The tested values for these parameters are 0, .01, 0.1, .25, .5, .75, and 1.

Finally, we observe the behavior of the algorithm SSDASF when we combine labeled target points and labeled source points in the learning sample. In other words, we measure the ability of our method to learn a low-error classifier when a part of the learning sample is drawn according to the target domain. For that purpose, each DA task is repeated 9 times by using the semi-supervised extension SSDASF with the addition of 9 random sets of 2, 4, 8, 10, 12, 14, 16, 18, and 20 labeled target examples in the learning sample. For these cases, we compare results with a SF-classifier learned only from the considered target labeled sample. Additionally, we regard the impact of the parameter κ of Problem ($SSDASF_{opt}$). In this case, we fix λ, β and the number of labeled target data at 10. The tested κ are 0, .01, 0.1, .25, .5, .75, .80, .85, .90, .95, .99, and 1. Note that, in this context, we add target (both labeled and unlabeled) landmarks in R, while for DASF, R contains only the learning source examples.[9] Moreover, in this study, we have not reported the observation of λ and β of Problem ($SSDASF_{opt}$), since the behavior of SSDASF for these parameters is the same than for DASF.

Simultaneously, we compute the average time costs of each algorithm with fixed parameters. Actually, the baseline is the one of learning a SF-classifier. Then, we report the duration

[9] This point is discussed in the conclusion.

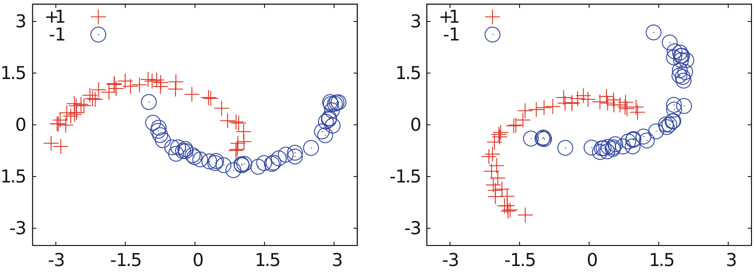


Fig. 6 Toy problem: on the *left*: a source sample. on the *right*: a target sample with a 50° rotation

of the other methods as the ratio of this baseline.[10] We also consider the execution time of the first iteration of our approaches (reported as ${}_{it_1}$).

Recall that considering all the possible pairs is completely intractable and that our iterative method provides a way to tackle this problem. Within these experiments, we will show that our iterative procedure leads to a very reasonable and competitive additional cost.

7.3 Synthetic toy problem

Setup As the source domain we consider a classical binary problem with two inter-twinning moons, each class corresponding to one moon (Fig. 6). We then considerate 8 different target domains by rotating anticlockwise the source domain according to 8 angles. The higher the angle, the more difficult the problem becomes. For each domain, we generate 300 instances (150 of each class). Moreover, to assess the generalization ability of our approach, we evaluate each algorithm on an independent test set of 1500 points drawn from the target domain (not provided to the algorithm). Each DA problem is repeated 10 times and the results will come next.

Choice of the "best" similarity function Before presenting the results and in order to evaluate if K_{ST} is a better similarity on the target domain, we propose to empirically study the (ϵ,γ)-guarantees on the target sample according to Definition 1. For that purpose, given $R=\{\mathbf{x}'_j\}_{j=1}^{d_u}$, we estimate ϵ as a function of γ from a labeled target sample $\{\mathbf{x}_{i'}, y_{i'}\}_{i'=1}^{d_t}$ (with true labels but only for this evaluation). Indeed, for a given γ, ϵ is the proportion of examples $\mathbf{x}_{i'}$ verifying $\sum_{\mathbf{x}'_j\in R} y_i y_{i'} K(\mathbf{x}_{i'}, \mathbf{x}'_j)/d_u < \gamma$. For each similarity function, we compute ϵ according to 20 values of γ between 0 and 1. We then obtain a curve representing ϵ as a function of γ. By considering each DA problem (each rotation angle), we obtain two curves and the best similarity function is the one with a lower area under the curve, meaning a lower error in average. Figure 7 shows the goodness guarantees of the similarity functions over each problem. We observe for hardest problems ($\geq 50°$) an improvement of the goodness with the normalized similarity K_{ST}. For easier tasks, this improvement is not significant, justifying the fact that the similarity K can be better. Our normalized similarity thus seems relevant only for hard DA problems.

Note that the ϵ rate is relatively high because we consider only landmarks from the source sample to study our adaptation capability.

Results The average accuracy of each method is reported in Table 2. We also indicate the average number of support vectors (SV) used by SVM, TSVM, and DASVM, the number of

[10] For example, a cost of 0.5 means that the algorithms need a running time half as long and a cost of 2 means a duration of twice as long.

Table 2 Toy problem: average accuracy over the 8 toy problems

Rotation angle	20°	30°	40°	50°	60°*	70°*	80°*	90°*
SVM	89.68 ± 0.78	75.99 ± 0.92	68.84 ± 0.85	60.00 ± 1.08	47.18 ± 2.82	26.12 ± 3.12	19.22 ± 0.28	17.2 ± 0.37
SV	18 ± 0.99							
SF	92.4 ± 3.13	81.81 ± 4.62	72.55 ± 7.60	57.85 ± 4.81	43.93 ± 4.46	39.2 ± 9.64	35.93 ± 10.93	36.73 ± 10.17
Land.	24 ± 1.72				22 ± 3.57	20 ± 2.06	20 ± 2.82	20 ± 1.51
TSVM	**100** ± 0.00	78.98 ± 2.31	74.66 ± 2.17	70.91 ± 0.88	64.72 ± 9.10	21.28 ± 1.26	18.92 ± 1.10	17.49 ± 1.12
SV	28 ± 1.92	37 ± 3.77	37 ± 2.66	37 ± 1.50	38 ± 2.67	35 ± 2.93	37 ± 2.10	36 ± 1.69
DASVM	100 ± 0	78.41 ± 4.56	71.63 ± 4.16	66.59 ± 4.01	61.57 ± 4.15	25.34 ± 3.28	21.07 ± 2.33	18.06 ± 2.66
SV	20 ± 3.13	20 ± 4.42	26 ± 6.80	28 ± 2.81	29 ± 3.62	34 ± 7.58	38 ± 6.20	23 ± 4.95
DASF	99.80 ± 0.40	**99.55** ± 1.19	**91.03** ± 3.30	**81.27** ± 4.36	**65.23** ± 6.36	**61.95** ± 4.88	**60.91** ± 2.24	**59.75** ± 2.11
Land.	**10** ± 2.32	**10** ± 1.59	**9** ± 2.21	**8** ± 3.27	**4** ± 0.99	**4** ± 2.16	**4** ± 1.84	**3** ± 1.06
$\hat{d}_{\mathcal{H}\Delta\mathcal{H}}$ in ϕ_0^R	0.58 ± 0.04	1.16 ± 0.04	1.31 ± 0.04	1.34 ± 0.04	1.34 ± 0.03	1.32 ± 0.03	1.33 ± 0.03	1.31 ± 0.05
$\hat{d}_{\mathcal{H}\Delta\mathcal{H}}$ in ϕ_{final}^R	0.33 ± 0.12	0.66 ± 0.11	0.82 ± 0.13	0.85 ± 0.11	0.39 ± 0.15	0.40 ± 0.05	0.49 ± 0.12	0.45 ± 0.09

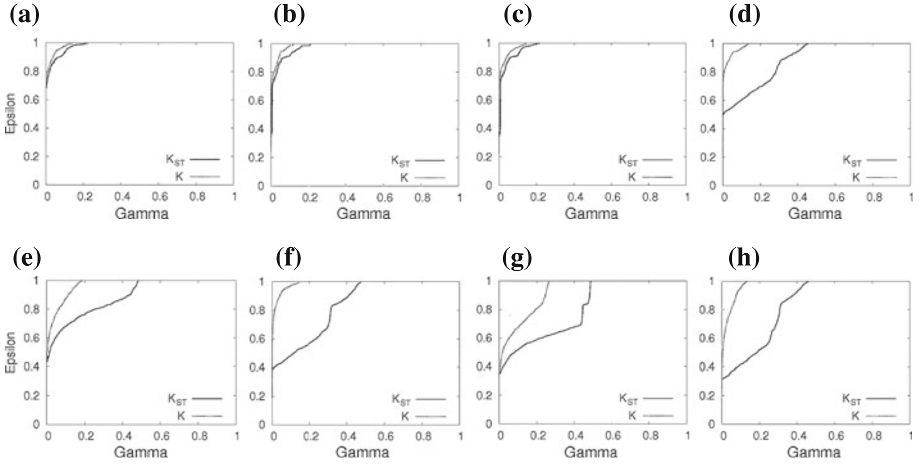


Fig. 7 Toy problem: Goodness of the similarities over the target samples: ϵ (Epsilon), as a function of γ (Gamma). **a** For a 20° task. **b** For a 30° task. **c** For a 40° task. **d** For a 50° task. **e** For a 60° task. **f** For a 70° task. **g** For a 80° task. **h** For a 90° task.

landmarks (LAND.) selected by SF and DASF and an estimation of $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(D_S, D_T)$ between the marginal distributions in the initial ϕ_0^R-space and the final ϕ_{final}^R-space. We can make the following remarks.

- In average, DASF outperforms the other methods. It is significantly better for every problem with an angle greater than 20°. While the accuracy of TSVM and DASVM falls down from 60°, DASF still remains competitive even when the difficulty increases. As we have shown on Fig. 7, we have the confirmation that the normalized similarity (*) is preferred in these cases.
- The number of landmarks (LAND.) is significantly lower than the number of support vectors SV, which confirms that DASF produces very sparse models with good performances. The gain ratio is between 3 to 12. The DASF-classifiers are also sparser than the SF-ones which use a L1-regularization too. Finally, they tend to be sparser for difficult problems as suggested by Lemma 2.
- The empirical $\mathcal{H}\Delta\mathcal{H}$-distance between the domains is lower at the last iteration—between 1 and 9—showing our iterative process is effectively able to quickly move closer the distributions. As evoked before, DASF tends to build a small projection space for hard tasks, probably to have sufficiently close domains, but it may imply a loss of expressiveness.

Figure 8 shows two DASF runs on two DA problems. For both cases, the empirical $\mathcal{H}\Delta\mathcal{H}$-distance decreases significantly in comparison with the first iteration. The algorithm stops when the joint error reaches a minimum after decreasing continuously. Note that the final projection space is not always the one with the lowest distance. This is because we need to find a good compromise between the minimization of the $\mathcal{H}\Delta\mathcal{H}$-distance and the one of the source error. Thanks to the iterative procedure, DASF is then able to slightly auto-correct the projection space when it allows a better adaptation. For the 30° example, DASF finds a null error classifier on the target test sample. For the more difficult 50° example, DASF performs better than the SF-classifier learned only on the source data. Note that the source error increases, which is expected since we aim at being performing on the target domain.

DASF: Influence of the hyperparameters λ and β We aim at observing how the correct classification rate evolves according to different values of the hyperparameters of our global

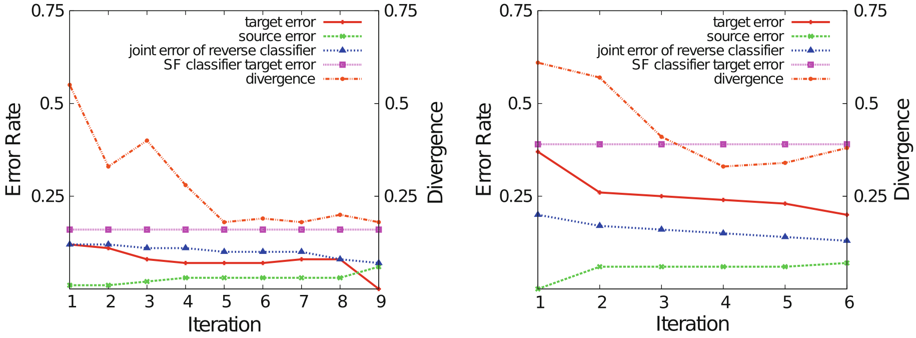


Fig. 8 Toy problem: two DASF runs. On the *left* for a 30° rotation, on the *right* for a 50° rotation. On the *left* y axis is the error rate, on the *right* y axis is the divergence measure. We provide the error rates of the classifiers h_l built at each iteration on the source and target test samples, the divergence $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}$, the reverse classifier joint error, the error on the target test sample of a SF-classifier learning without DA as a baseline

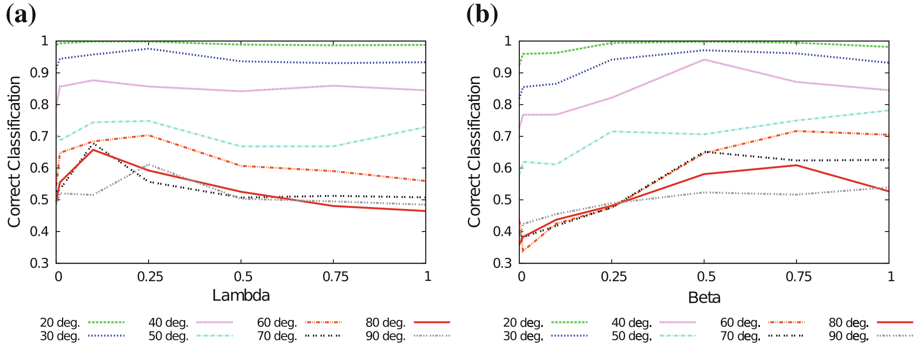


Fig. 9 Toy problem: the correct classification rate according to λ and β in Problem ($DASF_{opt}$) of DASF. **a** The average rate according to λ with $\beta = 0.75$. **b** The average rate according to β with $\lambda = 0.15$

Problem ($DASF_{opt}$) for DASF. Experiments reported on Fig. 9a, respectively, Fig. 9b, correspond to the average correct classification rate for each rotation angle according to $\lambda \in [0; 1]$ (with the bestβ), respectively, $\beta \in [0; 1]$ (with the best λ). Moreover, in Table 3, we indicate the average number of reasonable points (i.e., the sparsity) associated with the learned models. We can make the following remarks:

- Concerning λ, on Fig. 9a we note that for the 4 easiest tasks, this parameter does not influence a lot the results: The gain is between 0 and 0.1. However, for the 4 hardest tasks, chosen a relevant λ shows better results: The gain is between 0.1 and 0.2. For all problems, the best value seems to stand between 0.1 and 0.25. From Table 3(a), we note that the sparsity of the models does not really depend on the value of λ, indeed this sparsity is also influenced by β and it is the combination of the both that leads to sparser models.
- On Fig. 9b, we remark that the models are more sensitive to β than λ. In fact, when β tends to the best value, the increasing of the correct classification rate is more significant than for λ: The gain is between 0.05 and 0.35. Like in the observation of λ, the hardest tasks are more sensitive to β. The relevant value of β seems to stand between 0.5 and 1. Moreover, as expected by Lemma 2, the sparsity increases with the value of β associated with the best model (see Table 3(b)) and with the difficulty of the task.

Table 3 Toy problem: the average landmark number (i.e., the sparsity) according to λ and β from DASF

Rotation angle	20°	30°	40°	50°	60°	70°	80°	90°
(a) The sparsity according λ with $\beta = 0.75$								
$\lambda = 0$	16	12	13	8	12	6	7	8
$\lambda = 0.01$	15	10	9	6	5	5	8	11
$\lambda = 0.1$	**10**	**9**	**8**	**6**	3	**3**	**2**	6
$\lambda = 0.25$	15	11	12	5	**4**	5	6	**3**
$\lambda = 0.5$	18	17	14	10	6	6	9	4
$\lambda = 0.75$	13	15	13	10	8	9	8	8
$\lambda = 1$	15	21	16	9	12	12	8	7
(b) The sparsity according to β with $\lambda = 0.15$								
$\beta = 0$	24	24	24	24	22	20	20	20
$\beta = 0.01$	22	18	20	20	4	6	6	11
$\beta = 0.1$	16	17	17	19	3	4	5	9
$\beta = 0.25$	13	11	13	12	3	3	5	7
$\beta = 0.5$	**11**	**10**	**8**	11	3	**3**	3	7
$\beta = 0.75$	12	16	11	11	**3**	4	**3**	6
$\beta = 1$	14	11	8	**6**	3	5	4	**5**

Bold values are associated with the best model

SSDASF: Influence of combining source and target labeled learning sample We consider here SSDASF, the semi-supervised extension of DASF combining source and target labeled samples by solving ($SSDASF_{opt}$) with the setup of Algorithm 2.

Firstly, on Fig. 10a, we observe the average correct classification rates by adding target labeled samples of different sizes. The results show that the classifier's performance increases with the number of labeled target examples, which is an expected behavior. The more difficult the problem, the more significant the increase. However, for the hardest tasks ($\geq 70°$), we are not able to find an efficient classifier in comparison with a SF-classifier only learned from the labeled target sample (without DA). It is coherent with the analysis of the semi-supervised generalization bound since for the hardest tasks, the models need more target labels when the domains are far, and sometimes it is more reliable to focus only on target data.

Secondly, on Fig. 10b, we observe the behavior of SSDASF (using 10 target labels) according to the hyperparameter κ, which weights the importance of the labeled target data in Problem ($SSDASF_{opt}$). We fix $\lambda = 0.15$ and $\beta = 0.75$. As expected, this method needs a high κ value between 0.9 and 0.99: The gain is between 0.1 and 0.5, and again the impact is higher for the hardest problems. From Table 4, κ directly influences the sparsity of the models for the easiest tasks. However, for the hardest ones, that is, when B_R tends to be high, κ has a lower impact. Finally, as expected from Lemma 3, the use of target labeled data leads to less sparse models.

Computational costs We report now in Table 5 the average execution time of each algorithm with fixed parameters. We take the SF-classifier learning as the baseline, which with SVM is the fastest. Firstly, we can observe that the first iteration of DASF (DASF$_{it_1}$) needs the same time as the usual SF algorithm. The additional cost due to the iterations of DASF is reasonable since for at most 10 iterations, it is between 4 and 8 times longer than DASF$_{it_1}$. Secondly, our method DASF is faster than DASVM. It takes between almost one-third as

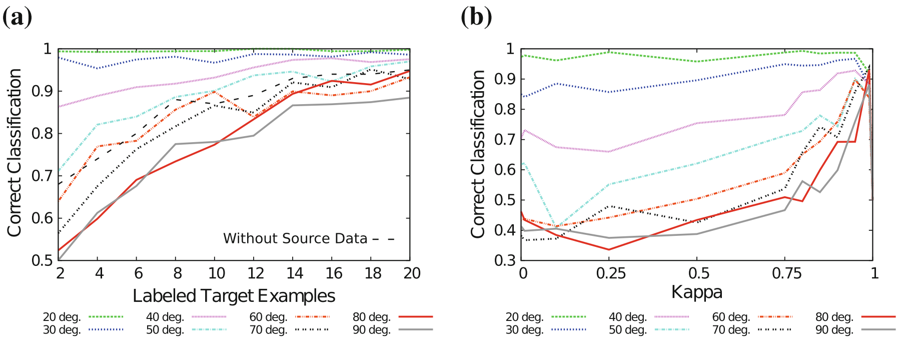


Fig. 10 Toy problem: the average correct classification rates obtained by combining source and target labeled samples in Problem ($SSDASF_{opt}$), that is, with SSDASF. **a** The average rates for each rotation angle according to the quantity of labeled target examples in the learning sample. The result obtained without source data (i.e., considering only target labels) is indicated with the *dashed black line with big dashes*. **b** The average rates according to κ with 10 target labeled examples, $\lambda = 0.15$ and $\beta = 0.75$

Table 4 Toy problem: the average number of landmarks selected (i.e., the sparsity) according to κ, with $\lambda = 0.15$ and $\beta = 0.75$ from SSDASF

Rotation angle	20°	30°	40°	50°	60°	70°	80°	90°
$\kappa = 0$	24	22	16	18	11	8	17	8
$\kappa = 0.01$	18	19	11	17	15	7	15	6
$\kappa = 0.1$	22	18	13	10	11	9	4	11
$\kappa = 0.25$	21	19	14	17	10	17	6	11
$\kappa = 0.5$	19	18	19	11	12	8	13	10
$\kappa = 0.75$	19	17	16	20	9	14	8	14
$\kappa = 0.8$	**22**	20	10	16	12	14	9	16
$\kappa = 0.85$	24	18	24	18	7	14	13	11
$\kappa = 0.9$	24	21	18	23	14	12	9	10
$\kappa = 0.95$	21	**23**	**20**	**14**	**11**	12	16	14
$\kappa = 0.99$	7	7	7	7	12	**9**	**11**	**7**

long. Nevertheless, TSVM is quicker than DASF, probably due to the cost of computing the pairs for DASF. Finally, we observe for the three adaptive methods—TSVM, DASVM, and DASF—that the lowest costs are obtained for 40° to 70° rotations. We have not reported the costs for SSDASF, since the execution times of SSDASF and DASF for these easy tasks are almost the same.

7.4 Image classification

Setup In this section, we experiment our approach on PascalVOC 2007 [23] and TrecVid 2007 [41] corpora. The goal is to identify visual objects and scenes in images and videos. TrecVid corpus is constituted of images extracted from videos and can be seen as an image corpus. Visual features used for those experiments are based on the prediction scores of 15 "intermediate" visual concepts (Animal, Building, Car, Cartoon, Explosion- Fire, Flag- US, Greenery, Maps, Road, Sea, Skin_face, Sky, Snow, Sports, Studio_setting), which have been successfully used in previous TrecVid evaluations. Each of those intermediate concepts is

Table 5 Toy problem: the average computational costs of each method measured as a ratio of the baseline SF

Rotation angle	20°	30°	40°	50°	60°*	70°*	80°*	90°*	Average
SVM	1	1	1	1	1	1	1	1	1.0
SF	1	1	1	1	1	1	1	1	1.0
TSVM	3	3	1	1	1	1	2	3	1.8
DASVM	29	14	13	8	8	19	26	28	18.1
DASF$_{it_1}$	1	1	1	1	1	1	1	1	1.0
DASF	8	7	6	6	4	6	7	6	6.2

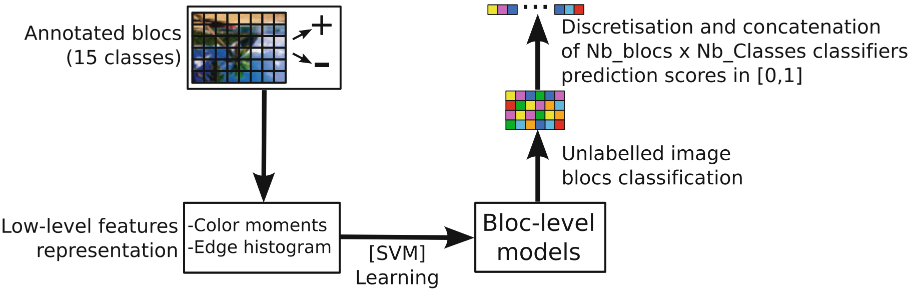


Fig. 11 Image classification: idea behind the visual features

detected using SVM-classifiers from color moments and edge orientations on 260 blocs of 32 × 32 pixels (data dimension is 3900) according to [4] (Fig. 11 is an illustration).

We performed two experiments. Section 7.4.1 deals with the first one where the objective is to evaluate the DA capability of our algorithm on close domains. The Sect. 7.4.2 presents the second experiment, which stands in an usual DA setup with potentially very different domains.

7.4.1 Adaptation capability when the label ratio is different between source and target sample

Setup The PascalVOC benchmark is constituted of 5,000 training images, 5,000 test images, and a list of 20 concepts to identify. Train and test sets are in fact relatively close ($\hat{d}_{\mathcal{H}\Delta\mathcal{H}} \simeq 0.05$) and a DA step is not necessary. We rather propose to evaluate the DA capability of our algorithm when the ratio $+/-$ is different between the source and target samples, leading to a harder DA task. Our objective is not to provide a solution in such a situation (specific methods already exist like [40]), but rather to evaluate if our method can avoid negative transfer and improve the accuracy over the test set. Since the two domains are close, we only evaluate our unsupervised approach DASF (adding labeled target examples will actually correspond to adding more labeled source instances).

In general, the ratio between positive and negative examples (ratio $+/-$) is less than 10 % in this dataset. For each concept, we generated a source sample constituted of all the training positive examples and negative examples independently drawn such that the ratio $+/-$ is $\frac{1}{3}/\frac{2}{3}$. We keep the original test set as the target sample. We applied the five methods previously described for learning a binary classifier for each concept. Due to the relative small

Table 6 Image classification: the F-measure results for each concept (Conc.) on the PascalVOC test target domains according to the F-measure

Conc.	Bird	Boat	Bottle*	Bus	Car	Cat	Chair	Cycle	Cow	Diningtable
SVM	0.18	0.29	0.01	0.16	0.28	**0.23**	0.24	0.10	0.15	0.15
SV	867	351	587	476	1096	882	1195	392	681	534
SF	0.18	0.27	0.11	0.12	0.34	0.20	0.21	0.10	0.11	0.10
Land.	237	203	233	212	185	**178**	241	139	239	**253**
TSVM	0.14	0.14	0.11	0.16	0.37	0.14	0.22	0.13	0.12	0.13
SV	814	704	718	445	631	779	864	390	888	515
DASVM	0.16	0.22	0.11	0.14	0.37	0.20	0.23	0.14	0.11	0.15
SV	922	223	295	421	866	1011	1418	706	335	536
DASF	**0.20**	**0.32**	**0.12**	**0.17**	**0.38**	**0.23**	**0.26**	**0.16**	**0.16**	**0.16**
Land.	**50**	**184**	**78**	**94**	**51**	378	**229**	**192**	**203**	372

Conc.	Dog*	Horse	Monitor	Motorbike	Person*	Plane	Plant	Sheep	Sofa	Train	Avg.
SVM	0.24	0.31	**0.16**	0.17	0.56	0.34	0.12	0.16	0.16	0.36	0.22
SV	436	761	698	670	951	428	428	261	631	510	642
SF	0.18	0.24	0.12	0.17	0.46	0.34	0.13	0.12	0.13	0.20	0.19
Land.	200	**247**	**203**	243	226	**178**	**236**	**128**	224	202	210
TSVM	0.22	0.17	0.12	0.12	0.44	0.18	0.10	0.12	0.15	0.19	0.17
SV	704	828	861	861	1111	585	406	474	866	652	705
DASVM	0.22	0.23	0.12	0.14	0.55	0.30	0.12	0.13	0.17	0.28	0.20
SV	**180**	802	668	841	303	356	1434	246	486	407	622
DASF	**0.25**	**0.32**	**0.16**	**0.18**	**0.58**	**0.35**	**0.15**	**0.20**	**0.18**	**0.42**	**0.25**
Land.	391	384	287	**239**	**6**	181	293	153	**167**	**75**	**200**

Avg. is the average result

ratio +/− in the target sample, we evaluate the performances according to the well-known F-measure defined by $\frac{2.Precision.Recall}{Precision+Recall}$.

Results The results are reported in Table 6. First, TSVM and DASVM perform badly, probably because of the difference between target and source ratios +/−, which cannot be estimated due to the lack of information on the target sample. SVM performs often better than the two previous ones that can be explained by the similarity between the train and test data. DASF has the best behavior on average. It always improves the results of a SF-classifier, avoiding negative transfer, and is the best for 18 concepts. Moreover, it always outputs significantly sparser models. As an illustration, we give in Fig. 12 the landmarks selected for the concept PERSON.

Computational costs The average execution time of each algorithm (with fixed parameters) is reported in Table 7. We recall that the SF-classifier learning is the baseline. For this real corpus, DASVM is significantly more costly. Unlike the toy problem, SVM is on average longer than the baseline and DASF quicker than TSVM. This is probably a direct consequence of the use of the $L1$-norm regularization. Indeed, the size of the set of landmarks is lower than the quantity of possible support vectors. Again, an interesting point is that the

Fig. 12 Image classification: the 6 landmarks selected for the concept PERSON, the first three images are positive and the last three are negative (PascalVOC)

Table 7 Image classification: the average computational costs of each method measured as a ratio of the baseline SF

CONC.	Bird	Boat	Bottle*	Bus	Car	Cat	Chair	Cycle	Cow	Diningtable
SVM	5.5	0.8	2.1	2	1.3	9	3.4	8.8	0.3	1.9
SF	1	1	1	1	1	1	1	1	1	1
TSVM	18.2	4.1	8	6	3.3	11.4	8.7	28.6	2.8	5.7
DASVM	4,254	1,440	3,870	4,860	1,470	3,428	2,674	2,828	900	1,300
DASF$_{it_1}$	1	1	1	1	1	1	1	1	1	1
DASF	4.9	6	6	9	7.3	5	3.8	6.8	2.7	2.1

CONC.	Dog*	Horse	Monitor	Motorbike	Person*	Plane	Plant	Sheep	Sofa	Train	AVG.
SVM	1.4	3.7	4.2	5	2.1	2.9	0.5	1.4	2	1.4	2.2
SF	1	1	1	1	1	1	1	1	1	1	1
TSVM	1.9	7.4	12.2	5.9	4.5	10	0.7	4.4	15.9	10.1	8.49
DASVM	340	3,553	4,230	3,060	675	1,710	3,900	1,836	1,912	1,550	2,489
DASF$_{it_1}$	1	1	1	1	1	1	1	1	1	1	1
DASF	6.1	5.8	8.8	5.5	5.6	5	3.7	3.6	5.7	5.5	5.4

cost of the first iteration of our method (DASF$_{it_1}$) is the same as the cost of a SF learning. Moreover, the additional cost due to the iterations of DASF is again very reasonable: it is of a factor 5.4 on average and significantly lower than DASVM and TSVM.

7.4.2 Adaptation from PascalVOC 2007 to TrecVid 2007

Setup In the last experiment, we select the 6 common concepts between TrecVid 2007 and PascalVOC 2007. For each concept, we keep our previous PascalVOC train set as the source domain and take, as the target domain, a TrecVid set of examples with the same ratio +/− as the train set. $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}$ is about 1.4, justifying the high difference between the two corpora and thus a potentially hard DA task.

Results The results evaluated with the F-measure are reported in Table 8. DASF obtains the best results on average and outputs again significantly sparser models. Finally, for those hard tasks, the normalized similarity is always preferred (*), showing that DASF is effectively able to deal with non-symmetric non-PSD good similarities. K_{ST} has the interest of incorporating some target information, which seems useful for hard DA tasks.

DASF: Influence of the hyperparameters λ and β In these experiments, we observe on Fig. 13 the impacts on the F-measure of λ (with the best β) and β (with the best λ). We make the following remarks:

Table 8 Image classification: the F-measure results obtained for each concept on the TrecVid target domains according to the F-measure

Concept	Boat*	Bus*	Car*	Monitor*	Person*	Plane*	Average
SVM	0.56	0.25	0.43	0.19	0.52	0.32	0.38
SV	351	476	1,096	698	951	428	667
SF	0.49	0.46	0.50	0.34	0.45	0.54	0.46
Land.	214	224	**176**	246	226	178	211
TSVM	0.56	0.48	0.52	0.37	0.46	0.61	0.50
SV	498	535	631	741	1024	259	615
DASVM	0.52	0.46	**0.55**	0.30	0.54	0.52	0.48
SV	202	222	627	523	274	450	383
DASF	**0.57**	**0.49**	**0.55**	**0.42**	**0.57**	**0.66**	**0.54**
Land.	**120**	**130**	254	**151**	**19**	**7**	**113**

Avg is the average result

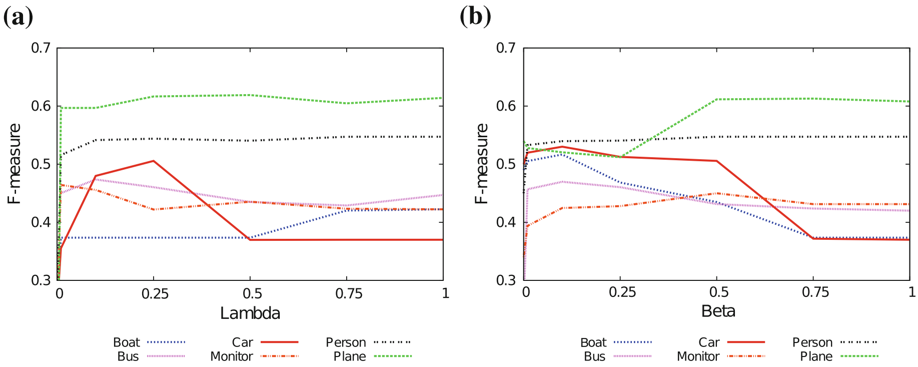


Fig. 13 Image classification: the results for each concept according to λ and β in Problem ($DASF_{opt}$), with DASF. **a** The F-measure according to λ with β fixed. **b** The F-measure according to β with λ = 0.15

- From Fig. 13a, the parameter λ shows a relative influence except for the concept Car where the value leading to the best classifier is near 0.25 (for which the gain is at least 0.15). For the others, the gain is lower than 0.1. The best λ depends on the considered problem but must be greater than 0, indicating that the corresponding regularization is necessary.
- From Fig. 13b, β clearly shows a higher impact: For Boat, Car and Plane the gain is between 0.1 and 0.15. Except for Person where the gain is *quasi* null, the choice of a relevant β can imply an improvement of at least 0.05 and even more. The value associated with the best classifier is greater than 0 and belongs to [0.01; 0.25] except for the concept Plane which prefers a β greater than 0.5.

Like for the toy problems, the parameter β has a higher impact. Thus, for lightening a bit the search of relevant parameters, one can focus more precisely on β than on λ.

SSDASF: Influence of combining source and target labeled learning sample On Fig. 14a, we can observe the average results for each concept by running SSDASF on a combination

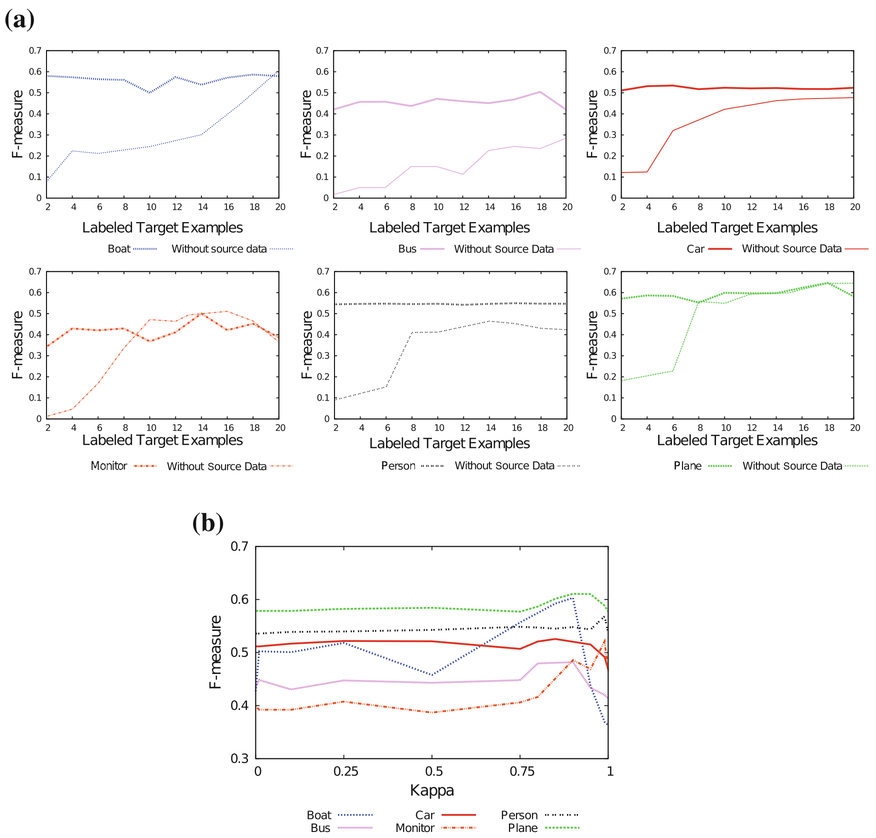


Fig. 14 Image classification: the F-measure obtained for each concept by combining source and target labeled samples in Problem ($SSDASF_{opt}$), that is, with SSDASF. **a** The F-measure according to the quantity of labeled target examples in the learning sample. “Without Source Data” corresponds to the result obtain with a SF-classifier learned only from the target labeled sample. **b** The F-measure according to κ with 10 labeled target examples, $\lambda = 0.15$ and β fixed

of source and target labeled learning samples. These results improve those obtained without target labels. We can remark that with less than 8 target labeled examples, our extended approach SSDASF always improves the results in comparison with a SF-classifier learned only from the target labeled data. Some concepts may even need more than 20 target examples that shows that the addition of a few target labels can be very useful.

In Fig. (14b) are reported results for the different values of κ (with 10 target labels and the best λ and β). Then, we clearly see that a relevant value for κ stands between 0.9 and 0.99 showing that the target labels give an important—additional—information during the learning process for these hard DA tasks.

Finally, it appears that, for some concepts, the used visual features may be not very expressive, which can explain the difficulty to obtain better results. In fact, in many image processing or multimedia issues, data are often represented with multimodal or multiview features in order to have a higher level of expressiveness. Taking into account such multimodal features would lead to further investigation, out of the scope of this paper. But this might be clearly a promising perspective.

Table 9 Image classification: the average computational costs of each method measured as a ratio of the baseline SF

CONCEPT	Boat*	Bus*	Car*	Monitor*	Person*	Plane*	AVERAGE
SVM	0.8	2	1.3	4.2	2.1	2.9	*2.2*
SF	1	1	1	1	1	1	*1*
TSVM	5	7	3.6	13.2	5.5	11	*7.5*
DASVM	3,870	720	2,370	2,790	300	540	*1,765*
DASF$_{it_1}$	1	1	1	1	1	1	*1*
DASF	9.5	10	8	23	3.4	10.5	*10.7*
SSDASF$_{it_1}$	4.4	2	0.4	1	0.6	1.5	*1.6*
SSDASF	29.8	14.4	6.6	29	3.4	13	*16*

Computational costs Table 9 corresponds to the average computational costs (with fixed parameters) reported as a ratio of the baseline (SF). The only difference with the previous image classification task (*c.f.* Sect. 7.4.1) is that TSVM is here quicker than DASF. This may be due to the difficulty of the task. In fact, in Sect. 7.4.1, the distance between the marginal distributions is low, whereas in this experiment the distance is large. This imply a harder construction of the set of pairs: When the points are far from each other, the minimization of the objective function of Problem (7) needs more time.

Lastly, SSDASF (and its first iteration) is about 1.5 more longer than the unsupervised DASF. The Problem ($SSDASF_{opt}$) is actually constructed by adding constraints to Problem ($DASF_{opt}$). Then, it explains why it may need more time for being solved. For both DASF and SSDASF, the overhead cost due to the iterations is again relatively reasonable with a factor 10 on average, showing that the iterative approach is still competitive in terms of computational cost.

8 Conclusion and perspectives

In this paper, we have proposed a novel domain adaptation approach that makes use of the framework of Balcan et al. [6,7] allowing one to deal with similarity functions potentially non-PSD and non-symmetric. Our method relies on a regularization term that helps to build a projection space—made of similarities to landmark points—by selecting the landmarks that are both close to the source and target examples. We have also proposed an effective iterative procedure in order to lighten the search of the projection space by a reweighting of the similarities. The linear formulation of the method enables the proposed algorithm to output sparse models, even when the DA task is hard. We have also studied the generalization ability of our method according to the framework of robustness, which allows us to take into account our regularizers. Moreover, we have extended our method for allowing the use of some few target labels. We have shown experimentally good adaptation capabilities on various tasks. Furthermore, our method always outputs sparser models, which is clearly an advantage for a large-scale application perspective. Additionally, our results show that a similarity renormalized according to a DA objective in a non-PSD and non-symmetric way enables us to infer better models for difficult domain adaptation problems.

Finally, we present and discuss several perspectives.

Designing non-PSD similarity function by metric learning From our experimental evaluation (Sect. 7.1), it appears that the use of non-PSD, non-symmetric functions can be useful for solving domain adaptation tasks. So far, we have only proposed a simple heuristic for designing such similarities. It is nevertheless an important issue to be able to automatically design relevant good similarities. We think that a possible direction is to investigate some metric learning approaches for domain adaptation. At the moment, few methods exist [15,26,33,50] and they mainly focus on PSD similarities. A possible way could be to combine such approaches with non-PSD similarity learning like the work of [8].

Building a projection space of landmark points According to the theory of [6,7], a low-error linear classifier can be learned in the explicit ϕ^R-space induced by the mapping function ϕ^R, defined by the landmarks (see Eq. (2)). In fact, according to Theorem 2, the process needs enough different and representative landmarks for learning a good classifier. On the other hand, using a lot of landmarks implies a high-dimensional projection space and thus to deal with a more complex optimization problem. One possible perspective is to use a preprocessing step for selecting a limited set of relevant landmark points (such as clustering [34] for example) in order to deal with a lower dimensional space. An important issue in this context is to select the ratio of source and target instances to use in the projection space. Indeed, in our experiments, we have noticed that in the unsupervised setting, the addition of target instances in the space of landmarks does not improve the results while the addition of target landmarks appears necessary in the semi-supervised approach. Moreover, this step is clearly related with the metric learning perspective evoked above since the projection space also depends on the similarity and a good similarity could compensate the lack of dimension.

Relationships with other DA frameworks The two previous perspectives are of high importance for precisely modeling the domain adaptation problem. For example, if the source and target domains are not very different, we have to slightly modify a relevant projection space for the source domain. This can be done by looking for some few new relevant landmark points. Some outlier detection can eventually help to find the part of the density that requires more attention. However, if the two domains are very different, we then must modify drastically the projection space. Another interesting and important perspective is also to investigate the link with reweighting methods for domain adaptation like [30,36,42] and in particular their relationships with our iterative procedure for reweighting similarities. Another point concerns our generalization bound: The divergence measure $d_{\mathcal{H}\Delta\mathcal{H}}$ used is not directly linked with the framework of robustness. In their extended work, [46] mention in their perspectives that the sum of the absolute values of the deviations between the expectations of source and target examples in each part of the instance space cover can be used to bound the domain adaptation error. It may lead to better convergence bound but deserves more investigation in our case.

Combining multiple source data To deal efficiently with images or video corpora, it is necessary to use multimodal or multiview representations allowing one to combine multiple features taking into account various information (such as colors, textures, spatiotemporal descriptors, …). A perspective is then to consider an adaptation of some multisource frameworks presented in [7,9,17,22,35]. Another standpoint could be to study some multi-tasks approaches [2,24,47].

Acknowledgments The authors are very grateful to Marc Sebban for his many useful remarks that were the basis of a significant improvement of the paper. We would like to acknowledge support for this project from the French project VideoSense ANR-09-CORD-026 of the ANR and the IST Program of the European Community, under the PASCAL2 Network of Excellence, IST-2007-216886.

Appendix

Proof of Theorem 8

In order to prove this theorem, we first need two technical lemmas from [9] able to link $\mathrm{err}_T(h)$, $\mathrm{err}_\kappa(h)$ and $e\hat{r}r_\kappa(h)$.

Lemma 4 [9] *Let h be an hypothesis in a class $\mathcal{H}$, then*

$$|\,\mathrm{err}_\kappa(h) - \mathrm{err}_T(h)| \leq (1-\kappa)\left(\frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(D_S, D_T) + \nu\right).$$

Lemma 5 [9] *Let h be a fixed hypothesis, if a random sample of size d_l is generated by drawing i.i.d. θd_l points from P_T and $(1-\theta)d_l$ points from P_S, then for any $\delta > 0$, with probability at least $1-\delta$ (over the choice of the samples),*

$$|e\hat{r}r_\kappa(h) - \mathrm{err}_\kappa(h)| < \sqrt{\frac{\kappa^2}{\theta} + \frac{(1-\kappa)^2}{1-\theta}}\sqrt{\frac{\log\frac{2}{\delta}}{2d_l}}.$$

Then, in order to take into account our regularizers, we use again the framework of robustness. For readability reasons, we recall the generalization bound of [45] for robust algorithms introduced previously in Sect. 4.2.2.

Theorem 4 ([45]) *If a learning sample $LS = \{(\mathbf{x}_i, y_i)\}_{i=1}^{d_l}$ is drawn i.i.d. from a distribution P and if an algorithm $\mathcal{A}$ is $(M, \epsilon(LS))$ robust, then for any $\delta > 0$, with probability at least $1-\delta$,*

$$\mathrm{err}_P(\mathcal{A}_{LS}) \;\leq \hat{\mathrm{err}}_P(\mathcal{A}_{LS}) + \epsilon(LS) + L^{UP}\sqrt{\frac{2M\ln 2 + 2\ln\frac{1}{\delta}}{d_l}},$$

where $\mathrm{err}_P(\mathcal{A}_{LS})$ and $\hat{\mathrm{err}}_P(\mathcal{A}_{LS})$ are respectively the generalization and the empirical errors over P of the model $\mathcal{A}_{LS}$ learned from LS, $L(\cdot,\cdot)$ being upper bounded by L^{UP}.

By combining source and target labels, this theorem cannot be used directly on err_κ since the learning sample $LS = (LS_S, LS_T)$ contains examples coming from two different distributions. However, by definition of err_κ and $\hat{\mathrm{err}}_\kappa$, the corresponding error on the source domain and on the target one are evaluated independently on each domain. A solution is then to apply the robustness theorem on each domain error and then to consider the convex combination of the two bounds with respect to κ.

Assuming a normalization such that $L^{UP} = 1$ in our case, we can now prove Theorem 8.

Theorem 8 *Let $\theta \in [0,1]$, $\kappa \in [0,1]$, and LS be a labeled learning sample of size d_l constituted of θd_l instances i.i.d. from target distribution P_T and $(1-\theta)d_l$ examples i.i.d. from source distribution P_S. Let $\eta', \eta > 0$ with $M = \max(M_\eta, M_{\eta'})$ a covering number for X, $\beta > 0$, $\lambda > 0$ and $B_R > 0$. For all $h \in \mathcal{H}$ minimizing the empirical error by Problem ($SSDASF_{opt}$), if $h^* = \mathrm{argmin}_{h'\in\mathcal{H}}\{\mathrm{err}_T(h')/e\hat{r}r_\kappa(h) \leq e\hat{r}r_\kappa(h')\}$, then with probability at least $1-\delta$,*

$$\mathrm{err}_T(h) \leq \mathrm{err}_T(h^*) + \sqrt{\frac{\kappa^2}{\theta} + \frac{(1-\kappa)^2}{1-\theta}}\sqrt{\frac{\ln\frac{4}{\delta}}{2d_l}} + \frac{\kappa(N_{\eta'}^T - N_\eta^S) + N_\eta^S}{(1-\kappa)\beta B_R + \lambda}$$
$$+\sqrt{\frac{4M\ln 2 + 2\ln\frac{4}{\delta}}{d_l}}\left(\frac{\kappa}{\sqrt{\theta}} + \frac{1-\kappa}{\sqrt{1-\theta}}\right) + 2(1-\kappa)\left(\frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(D_S, D_T) + \nu\right),$$

where $B_R = \min_{\mathbf{x}'_j \in R} \left\{ \max_{(\mathbf{x}_s,\mathbf{x}_t)\in \mathcal{C}_{ST}} |K(\mathbf{x}_s, \mathbf{x}'_j) - K(\mathbf{x}_t, \mathbf{x}'_j)| \right\}$ *and*

$$N_\eta^S = \max_{\substack{\mathbf{x}_a,\mathbf{x}_b \sim D_S \\ \rho(\mathbf{x}_a,\mathbf{x}_b)\leq \eta}} \|{}^t\phi^R(\mathbf{x}_a) - {}^t\phi^R(\mathbf{x}_b)\|_\infty,\ N_{\eta'}^T = \max_{\substack{\mathbf{x}_a,\mathbf{x}_b \sim D_T \\ \rho(\mathbf{x}_a,\mathbf{x}_b)\leq \eta'}} \|{}^t\phi^R(\mathbf{x}_a) - {}^t\phi^R(\mathbf{x}_b)\|_\infty.$$

Proof First, using the principle of Theorem 5, by applying Theorem 4 on the target domain and on the source domain, with Lemma 3 and eventually two different covers of X, we have respectively:

with probability $1-\frac{\delta}{4}$ and $N_{\eta'}^T = \max_{\substack{\mathbf{x}_a,\mathbf{x}_b \sim D_T \\ \rho(\mathbf{x}_a,\mathbf{x}_b)\leq \eta'}} \|{}^t\phi^R(\mathbf{x}_a) - {}^t\phi^R(\mathbf{x}_b)\|_\infty$,

$$\kappa\ e\hat{r}r_T(h) \leq \kappa \left(\mathrm{err}_T(h) + \frac{N_{\eta'}^T}{(1-\kappa)\beta B_R + \lambda} + \sqrt{\frac{4M_{\eta'}\ln 2 + 2\ln\frac{4}{\delta}}{\theta d_l}} \right),$$

with probability $1-\frac{\delta}{4}$ and $N_{\eta}^S = \max_{\substack{\mathbf{x}_a,\mathbf{x}_b \sim D_S \\ \rho(\mathbf{x}_a,\mathbf{x}_b)\leq \eta}} \|{}^t\phi^R(\mathbf{x}_a) - {}^t\phi^R(\mathbf{x}_b)\|_\infty$,

$$(1-\kappa)e\hat{r}r_S(h) \leq (1-\kappa) \left(\mathrm{err}_S(h) + \frac{N_{\eta}^S}{(1-\kappa)\beta B_R + \lambda} + \sqrt{\frac{4M_{\eta}\ln 2 + 2\ln\frac{4}{\delta}}{(1-\theta) d_l}} \right).$$

Let h be the hypothesis minimizing $e\hat{r}r_\kappa(h)$, $h^* = \mathrm{argmin}_{h'\in\mathcal{H}}\{\mathrm{err}_T(h')/e\hat{r}r_\kappa(h) \leq e\hat{r}r_\kappa(h')\}$ and let $A = \frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(D_S, D_T) + \nu$, then

$$\begin{aligned}
\mathrm{err}_T(h) &\leq \mathrm{err}_\kappa(h) + (1-\kappa)A, \quad \text{from Lemma 4,}\\
&\leq e\hat{r}r_\kappa(h) + \frac{\kappa(N_{\eta'}^T - N_\eta^S) + N_\eta^S}{(1-\kappa)\beta B_R + \lambda} + \sqrt{\frac{4M\ln 2 + 2\ln\frac{4}{\delta}}{d_l}} \left(\frac{\kappa}{\sqrt{\theta}} + \frac{1-\kappa}{\sqrt{1-\theta}} \right)\\
&\quad + (1-\kappa)A, \text{ by Theorem 4 on the two domain errors with } M = \max(M_{\eta'}, M_\eta),\\
&\leq e\hat{r}r_\kappa(h^*) + \frac{\kappa(N_{\eta'}^T - N_\eta^S) + N_\eta^S}{(1-\kappa)\beta B_R + \lambda} + \sqrt{\frac{4M\ln 2 + 2\ln\frac{4}{\delta}}{d_l}} \left(\frac{\kappa}{\sqrt{\theta}} + \frac{1-\kappa}{\sqrt{1-\theta}} \right)\\
&\quad + (1-\kappa)A, \quad \text{since } e\hat{r}r_\kappa(h) \leq e\hat{r}r_\kappa(h^*),\\
&\leq \mathrm{err}_\kappa(h^*) + \sqrt{\frac{\kappa^2}{\theta} + \frac{(1-\kappa)^2}{1-\theta}} \sqrt{\frac{\ln\frac{4}{\delta}}{2d_l}} + \frac{\kappa(N_{\eta'}^T - N_\eta^S) + N_\eta^S}{(1-\kappa)\beta B_R + \lambda}\\
&\quad + \sqrt{\frac{4M\ln 2 + 2\ln\frac{4}{\delta}}{d_l}} \left(\frac{\kappa}{\sqrt{\theta}} + \frac{1-\kappa}{\sqrt{1-\theta}} \right) + (1-\kappa)A, \quad \text{by Lemma 5,}\\
&\leq \mathrm{err}_T(h^*) + \sqrt{\frac{\kappa^2}{\theta} + \frac{(1-\kappa)^2}{1-\theta}} \sqrt{\frac{\ln\frac{4}{\delta}}{2d_l}} + \frac{\kappa(N_{\eta'}^T - N_\eta^S) + N_\eta^S}{(1-\kappa)\beta B_R + \lambda}\\
&\quad + \sqrt{\frac{4M\ln 2 + 2\ln\frac{4}{\delta}}{d_l}} \left(\frac{\kappa}{\sqrt{\theta}} + \frac{1-\kappa}{\sqrt{1-\theta}} \right) + 2(1-\kappa)A, \quad \text{by Lemma 4.}
\end{aligned}$$

□

References

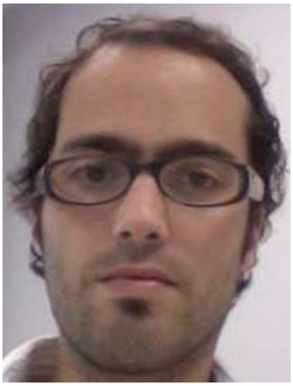
1. Abbasnejad M, Ramachandram D, Mandava R (2012) A survey of the state of the art in learning the kernels. Knowl Inf Syst 31(2):193–221. doi:10.1007/s10115-011-0404-6
2. Ando R, Zhang T (2005) A framework for learning predictive structures from multiple tasks and unlabeled data. J Mach Learn Res 6:1817–1853
3. Ayache S, Quénot G (2008) Video corpus annotation using active learning. In: Proceedings of the 30th European conference on information retrieval research (ECIR), vol 4956 of LNCS. Springer, pp 187–198
4. Ayache S, Quénot G, Gensel J (2007) Image and video indexing using networks of operators. J Image Video Process 1:1–113
5. Bahadori MT, Liu Y, Zhang D (2011) Learning with minimum supervision: a general framework for transductive transfer learning. In: Proceedings of the 11th IEEE international conference on data mining (ICDM), pp 61–70
6. Balcan M, Blum A, Srebro N (2008a) Improved guarantees for learning via similarity functions. In: Proceedings of the annual conference on computational learning theory (COLT), pp 287–298
7. Balcan M, Blum A, Srebro N (2008) A theory of learning with similarity functions. Mach Learn J 72(1–2):89–112
8. Bellet A, Habrard A, Sebban M (2011) Learning good edit similarities with generalization guarantees. In: Proceedings of European conference on machine learning and principles of data mining and knowledge discovery (ECML/PKDD), vol 6911 of LNCS, pp 188–203
9. Ben-David S, Blitzer J, Crammer K, Kulesza A, Pereira F, Vaughan J (2010) A theory of learning from different domains. Mach Learn J 79(1–2):151–175
10. Ben-David S, Blitzer J, Crammer K, Pereira F (2007) Analysis of representations for domain adaptation. In: Proceedings of advances in neural information processing systems (NIPS), pp 137–144
11. Ben-David S, Lu T, Luu T, Pal D (2010) Impossibility theorems for domain adaptation. JMLR W&CP 9:129–136
12. Bergamo A, Torresani L (2010) Exploiting weakly-labeled web images to improve object classification: a domain adaptation approach. In: Proceedings of advances in neural information processing systems (NIPS)
13. Blitzer J, Foster D, Kakade S (2011) Domain adaptation with coupled subspaces. In: Proceedings of AISTATS
14. Bruzzone L, Marconcini M (2010) Domain adaptation problems: a DASVM classification technique and a circular validation strategy. IEEE Trans Pattern Anal Mach Intell 32(5):770–787
15. Cao B, Ni X, Sun J-T, Wang G, Yang Q (2011) Distance metric learning under covariate shift. In: Proceedings of international joint conference on artificial intelligence (IJCAI), pp 1204–1210
16. Chang C-C, Lin C-J (2001) LIBSVM: a library for support vector machines. http://www.csie.ntu.edu.tw/~cjlin/libsvm
17. Chattopadhyay R, Ye J, Panchanathan S, Fan W, Davidson I (2011) Multi-source domain adaptation and its application to early detection of fatigue. In: Proceedings of the 17th ACM SIGKDD international conference on knowledge discovery and data mining (KDD). ACM, pp 717–725
18. Chen M, Weinberger K, Blitzer J (2011) Co-training for domain adaptation. In: Proceedings of advances in neural information processing systems (NIPS)
19. Cortes C, Mohri M (2011) Domain adaptation in regression. In: Proceedings of international conference on algorithmic learning theory (ALT), vol 6925 of LNCS, pp 308–323
20. Daumé H III (2007) Frustratingly easy domain adaptation. In: Proceedings of the association for computational linguistics (ACL)
21. Daumé H III, Kumar A, Saha A (2010) Co-regularization based semi-supervised domain adaptation. In: Proceedings of advances in neural information processing systems (NIPS)
22. Duan L, Tsang I, Xu D, Chua T (2009) Domain adaptation from multiple sources via auxiliary classifiers. In: Proceedings of international conference on machine learning (ICML), p 37
23. Everingham M, Van Gool L, Williams CKI, Winn J, Zisserman A (2007) The PASCAL visual object classes challenge 2007 (VOC2007) results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/
24. Fei H, Huan J (2011) Structured feature selection and task relationship inference for multi-task learning. In: Proceedings of the 11th IEEE international conference on data mining (ICDM). IEEE, pp 171–180
25. Freund R (1991) Polynomial-time algorithms for linear programming based only on primal scaling and projected gradients of a potential function. Math Program 51:203–222
26. Geng B, Tao D, Xu C (2011) DAML: Domain adaptation metric learning. IEEE Trans Image Process (TIP) 20(10):2980–2989

27. Guerra P, Veloso A Jr, WM, Almeida V (2011) From bias to opinion: a transfer-learning approach to real-time sentiment analysis. In: Proceedings of the 17th ACM SIGKDD international conference on knowledge discovery and data mining (KDD). ACM, pp 150–158
28. Huang J, Smola A, Gretton A, Borgwardt K, Schölkopf B (2006) Correcting sample selection bias by unlabeled data. In: Proceedings of advances in neural information processing systems (NIPS), pp 601–608
29. Jiang J (2008) A literature survey on domain adaptation of statistical classifiers. Technical report, Computer Science Department at University of Illinois at Urbana-Champaign. http://sifaka.cs.uiuc.edu/jiang4/domain_adaptation/da_survey.pdf
30. Jiang J, Zhai C (2007) Instance weighting for domain adaptation in nlp. In: Proceedings of the association for computational linguistics (ACL)
31. Joachims T (1999) Transductive inference for text classification using support vector machines. In: Proceedings of international conference on machine learning (ICML), pp 200–209
32. Junejo K, Karim A (2012) Robust personalizable spam filtering via local and global discrimination modeling. Knowl Inf Syst 1–36. doi:10.1007/s10115-012-0477-x
33. Kulis B, Saenko K, Darrell T (2011) What you saw is not what you get: domain adaptation using asymmetric kernel transforms. In: Proceedings of IEEE conference on computer vision and pattern recognition (CVPR 2011), pp 1785–1792
34. Macqueen J (1967) Some methods of classification and analysis of multivariate observations. In: Proceedings of the 5th Berkeley symposium on mathematical statistics and probability, pp 281–297
35. Mansour Y, Mohri M, Rostamizadeh A (2008) Domain adaptation with multiple sources. In: Proceedings of advances in neural information processing systems (NIPS), pp 1041–1048
36. Mansour Y, Mohri M, Rostamizadeh A (2009) Domain adaptation: learning bounds and algorithms. In: Proceedings of annual conference on learning theory (COLT), pp 19–30
37. Pan S, Yang Q (2010) A survey on transfer learning. IEEE Trans Knowl Data Eng 22(10):1345–1359
38. Quionero-Candela J, Sugiyama M, Schwaighofer A, Lawrence N (2009) Dataset shift in machine learning. MIT Press, Cambridge
39. Schweikert G, Widmer C, Schölkopf B, Rätsch G (2008) An empirical analysis of domain adaptation algorithms for genomic sequence analysis. In: Proceedings of advances in neural information processing systems (NIPS), pp 1433–1440
40. Seah C, Tsang I, Ong Y, Lee K (2010) Predictive distribution matching svm for multi-domain learning. In: Proceedings of European conference on machine learning and principles of data mining and knowledge discovery (ECML/PKDD), vol 6321 of LNCS. Springer, pp 231–247
41. Smeaton A, Over P, Kraaij W (2009) High-level feature detection from video in TRECVid: a 5-year retrospective of achievements. In: Multimedia content analysis, theory and applications. Springer, pp 151–174
42. Sugiyama M, Nakajima S, Kashima H, von Bünau P, Kawanabe M (2007) Direct importance estimation with model selection and its application to covariate shift adaptation. In: Proceedings of advances in neural information processing systems (NIPS)
43. Vapnik V (1998) Statistical learning theory. Springer, Berlin
44. Wang B, Tang J, Fan W, Chen S, Tan C, Yang Z (2012) Query-dependent cross-domain ranking in heterogeneous network. Knowl Inf Syst 1–37. doi:10.1007/s10115-011-0472-7
45. Xu H, Mannor S (2010) Robustness and generalization. In: Proceedings of annual conference on computational theory (COLT), pp 503–515
46. Xu H, Mannor S (2012) Robustness and generalization. Mach Learn J 86(3):391–423
47. Xu Z, Kersting K (2011) Multi-task learning with task relations. In: Proceedings of the 11th IEEE international conference on data mining (ICDM). IEEE, pp 884–893
48. Xue G-R, Dai W, Yang Q, Yu Y (2008) Topic-bridged plsa for cross-domain text classification. In: Proceedings of international ACM SIGIR conference on research and development in information retrieval, pp 627–634
49. Ye Y (1991) 'An O(n^3L) potential reduction algorithm for linear programming'. Math Program 50: 239–258
50. Zhang Y, Yeung D-Y (2010) Transfer metric learning by learning task relationships. In: Proceedings of the 16th ACM SIGKDD international conference on knowledge discovery and data mining (KDD). ACM, pp 1199–1208
51. Zhong E, Fan W, Yang Q, Verscheure O, Ren J (2010) Cross validation framework to choose amongst models and datasets for transfer learning. In: Proceedings of European conference on machine learning and principles of data mining and knowledge discovery (ECML/PKDD), vol 6323 of LNCS. Springer, pp 547–562

Author Biographies

Emilie Morvant is currently a Ph.D. student in Machine Learning and Multimedia at the Laboratoire d'Informatique Fondamentale (LIF) of Marseille, France, in the Qarma team. She is under the direction of Amaury Habrard and Stéphane Ayache. She received her bachelor's degree in 2008 in Computer Science from Saint-Etienne University, France, and her master's degree in 2010 in Fundamental Computer Science (Machine Learning and Data Mining speciality) from Aix-Marseille University, France. She is interested in classifiers fusion and domain adaptation methods for multimedia document indexing and also on PAC-Bayes theory. She is a member of the PASCAL2 Network of Excellence.

Amaury Habrard is currently Professor in the Machine Learning group at the Laboratoire Hubert Curien of the University Jean Monnet of Saint-Etienne. He received a Ph.D. in Machine Learning in 2004 from the University of Saint-Etienne. He was an assistant professor at the Laboratoire d'Informatique Fondamentale of the Aix-Marseille University until 2011, where he received an habilitation thesis in 2010. His research interests include statistical learning theory, domain adaptation, metric learning, and multimedia fusion. He is a member of the PASCAL2 Network of Excellence.



Stéphane Ayache is currently an assistant professor in the Machine Learning and Multimedia group (QARMA) at the Laboratoire d'Informatique Fondamentale (LIF) of the University of Aix-Marseille, France. He received a Ph.D. degree in 2007 from the Grenoble Institute of Technology (INPG, France) on "Semantic video indexing by combination of Image, Audio and Text features." His research interests include semantic multimedia indexing, multimedia retrieval, and machine learning models. He is a member of the PASCAL2 Network of Excellence.