Under consideration for publication in Knowledge and Information Systems

A New Boosting Algorithm for Provably Accurate Unsupervised Domain Adaptation

Amaury Habrard, Jean-Philippe Peyrache and Marc Sebban

Laboratoire Hubert Curien, UMR CNRS 5516
18 rue du Professeur Benoit Lauras - 42000 Saint-Etienne Cedex 2 - France

Abstract. *Domain Adaptation* (DA) is a new learning framework dealing with learning problems where the target test data are drawn from a distribution different from the one that has generated the learning source data. In this article, we introduce SLDAB (Self-Labeling Domain Adaptation Boosting), a new DA algorithm that falls both within the theory of DA and the theory of Boosting, allowing us to derive strong theoretical properties. SLDAB stands in the unsupervised DA setting where labeled data are only available in the source domain. To deal with this more complex situation, the strategy of SLDAB consists in jointly minimizing the empirical error on the source domain while limiting the violations of a natural notion of pseudo-margin over the target domain instances. Another contribution of this paper is the definition of a new divergence measure aiming at penalizing models that induce a large discrepancy between the two domains, reducing the production of degenerate models. We provide several theoretical results that justify this strategy. The practical efficiency of our model is assessed on two widely used datasets.

Keywords: Boosting; Domain Adaptation; Transfer Learning

1. Introduction

In the classic machine learning setting, training and test data are supposed to come from the same statistical distribution. However, it is worth noting that this assumption does not hold in many real applications challenging common learning theories such as the PAC model Valiant (1984). To cope with such situations, a

Received Oct 08, 2014
Revised Mar 05, 2015
Accepted Apr 04, 2015

new machine learning framework has been recently studied leading to the emergence of the theory of *domain adaptation* (DA) Ben-David et al. (2010), Mansour et al. (2009). A standard DA problem can be defined as a situation where the learner receives labeled data drawn from a *source* domain (or even from several sources Mansour et al. (2008)) and very few or no labeled points from the *target* distribution. DA arises in a large spectrum of applications, such as in computer vision Martínez (2002), speech processing Leggetter & Woodland (1995), Roark & Bacchiani (2003), natural language processing Blitzer et al. (2007), Chelba & Acero (2006), etc. During the past few years, new fundamental results opened the door for the design of theoretically well-founded DA-algorithms. In this paper, we focus on the scenario where the training set is made of labeled source data and *unlabeled* target instances. To deal with this more complex situation, several solutions have been presented in the literature (see, e.g., surveys Margolis (2011), Quionero-Candela et al. (2009)). Among them, *instance weighting-based methods* are used to deal with covariate shift where the labeling functions are supposed to remain unchanged between the two domains. On the other hand, *feature representation approaches* aim at seeking a domain invariant feature space by either generating latent variables or selecting a relevant subset of the original features. A third class of approaches, called *iterative self-labeling methods*, consists in iteratively inserting target examples, which have been labeled, in the learning set.

In this paper, our objective is to propose a new approach, taking advantage of both feature representation and iterative self-labeling approaches. The idea is to *iteratively* learn a large margin linear hyperplane in a *new projection space*. We present a novel DA algorithm which takes its origin from both the theory of boosting Freund & Schapire (1996) and the theory of DA. Let us remind that boosting (via its well known AdaBoost algorithm) iteratively builds a combination of weak classifiers. At each step, AdaBoost makes use of an update rule which increases (resp. decreases) the weight of those instances misclassified (resp. correctly classified) by previous classifiers. It is worth noting that boosting has already been exploited in DA methods but mainly in supervised situations where the learner receives some labeled target instances. In Dai et al. (2007), TrAdaBoost uses the standard weighting scheme of AdaBoost on the target data, while the weights of the source instances are monotonically decreased according to their margin. A generalization of TrAdaBoost to multiple sources is presented in Yao & Doretto (2010). On the other hand, some boosting-based approaches relax the constraint of having labeled target examples. However, they are proposed in the context of semi-supervised ensemble methods, *i.e.* assuming that the source and the target domains are (sufficiently) similar Bennett et al. (2002), Mallapragada et al. (2009).

In this paper, we present SLDAB (*Self-Labeling Domain Adaptation Boosting*), a boosting-like DA algorithm which both optimizes the *source classification error* and *margin constraints* over the unlabeled target instances. However, unlike state of the art self-labeling DA methods, SLDAB aims at also reducing the divergence between the two distributions in the space of the learned hypotheses. In this context, we introduce the notion of weak DA assumption which takes into account a measure of divergence. This classifier-induced measure is exploited in the update rule so as to penalize hypotheses inducing a large discrepancy. This strategy tends to prevent the algorithm from building degenerate models which would, e.g., perfectly classify the source data while moving the target examples far away from the learned hyperplane (and thus satisfying any

margin constraint). We present a theoretical analysis of SLDAB and derive several theoretical results that, in addition to good experimental results, support our claims.

The rest of this paper is organized as follows: after an overview on related work in DA in Section 2 and an intuition about our work in Section 3; we give notations and definitions in Section 4. SLDAB is presented in Section 5 and theoretically analyzed in Section 6. We then discuss the way to compute the divergence between the source and target domains in Section 7. Finally, we conduct two series of experiments and show practical evidences of the efficiency of SLDAB in Section 8, before discussing about generalization guarantees in Section 9 and concluding in Section 10.

2. Related Work in Domain Adaptation

We consider the binary classification problem in which we have a training set $S \cup T$ with S a set of labeled data (x', y') drawn from a source distribution $\mathcal{S}$ over $Z = X \times Y$, where X is the instance space and $Y = \{-1, +1\}$ is the set of labels and T a set of unlabeled examples x drawn from a target distribution $\mathcal{T}$ over X. We want to learn a classifier with an error rate $\epsilon_{\mathcal{T}}$ as low as possible over the *target distribution*. In this setting, the (unknown) generalization source error of a hypothesis $h \in \mathcal{H}$ is defined as follows:

$$\epsilon_{\mathcal{S}}(h) = \mathbb{E}_{(x',y')\sim\mathcal{S}}[\ell(h, (x', y'))],$$

where $\ell : \mathcal{H} \times X \times \{-1, +1\} \to \mathbb{R}^+$ is a nonnegative loss function, while the (unknown) generalization target error of h is:

$$\epsilon_{\mathcal{T}}(h) = \mathbb{E}_{(x,y)\sim\mathcal{T}}[\ell(h, (x', y'))].$$

2.1. Theoretical Results in DA

In the past few years, DA has been widely studied from a statistical learning theory point of view. Typically, these theoretical results take the form of upper bounds on the generalization target error of h, which has been learned from the source data. To assess the adaptation difficulty of a given problem at hand, Ben-David et al. introduce in Ben-David et al. (2010) the $\mathcal{H}\Delta\mathcal{H}$-divergence, noted $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}, \mathcal{T})$, which is a measure between the source and the target distributions w.r.t. $\mathcal{H}$. To estimate this divergence, the authors suggest to label each source example of S with -1 and each unlabeled example of T as $+1$ and train a classifier to discriminate between source and target data. The $\mathcal{H}\Delta\mathcal{H}$-divergence is immediately computable from the error of the classifier and a term of complexity which depends on the Vapnik-Chervonenkis dimension (VC-dim) of the hypothesis space. Intuitively, if the error is low, we are thus able to easily separate source and target data that means that the two distributions are quite different. On the other hand, the higher the error, the less different $\mathcal{S}$ and $\mathcal{T}$. The authors provide a generalization bound for domain adaptation on $\epsilon_T(h)$ which generalizes the standard bound on $\epsilon_S(h)$ by taking into account the $\mathcal{H}\Delta\mathcal{H}$-divergence between the source and target distributions. More formally:

$$\epsilon_{\mathcal{T}}(h) \leq \epsilon_{\mathcal{S}}(h) + d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}, \mathcal{T}) + \lambda, \tag{1}$$

where λ is the error of the ideal joint hypothesis on the source and target domains (which is supposed to be a negligible term if the adaptation is possible). Equation(1) expresses the idea that to adapt well, the DA algorithm has to learn an hypothesis h which works well on the source data while reducing the divergence between $\mathcal{S}$ and $\mathcal{T}$.

Based on this work, Mansour et al. introduce a new divergence measure in Mansour et al. (2009), called *discrepancy distance*. Its empirical estimate is based on the Rademacher Koltchinskii (2001), Bartlett & Mendelson (2002) complexity (rather than the VC-dim) allowing the use of arbitrary loss functions leading to generalization bounds useful for more general learning problems, such as in classification with SVMs or in regression.

There also exists other theoretical works that have been made in the field of DA, such as Mansour & Schain (2012), which takes advantage of the robustness properties introduced in Xu & Mannor (2010*a*, 2012*a*), together with the notion of λ-shift between two distributions, to derive generalization bounds on the target error.

2.2. Algorithmic Contributions

Following the underlying ideas of Equation(1), many DA algorithms have been proposed in the literature in different settings. We present in the following some of these approaches (for more details, the interested reader may refer to the following surveys Margolis (2011), Pan & Yang (2010), Jiang (2008)).

Instance Weighting When the training examples, drawn in a biased manner, are not representative enough of $\mathcal{T}$, we have to face a problem of *sample selection bias*. *Covariate shift* describes the sample selection bias scenario where the prior distributions $P_{\mathcal{S}}(x)$ and $P_{\mathcal{T}}(x)$ differ but the conditional probabilities are the same, *i.e.* $P_{\mathcal{S}}(y|x) = P_{\mathcal{T}}(y|x)$.

Assuming that a certain mass of the source data can be used for learning the target examples, the domain adaptation can be done by resorting to a reweighting scheme of the data. Some solutions have been proposed (*e.g.*, see Huang et al. (2006), Sugiyama et al. (2008)) that use the ratio of the test and training input densities. Roughly speaking, the idea is to increase the importance of source points located into a region where there is a high density of target points. A learning process is then launched on this reweighted training set, which is supposed to be much closer from the target distribution than the unweighted one. In Dudík et al. (2005), the sample selection bias is corrected by resorting to an entropy maximization, while in Tsuboi et al. (2009) the source data are reweighted while minimizing the Kullback-Leibler-divergence between the source and target distributions. Finally, in Bickel et al. (2007), a classifier is found as the solution of an optimization problem integrating the covariate shift problem.

New feature representations Rather than reweighting the source data, another way to handle DA problems consists in changing the feature representation of the data to better describe shared characteristics of the two domains $\mathcal{S}$ and $\mathcal{T}$. We distinguish two different strategies: the first one assumes that some features are generalizable, and thus they can be shared by both domains, while some others are domain-specific. In this context, a feature selection algorithm which penalizes or removes some features can be used to find the shared low-

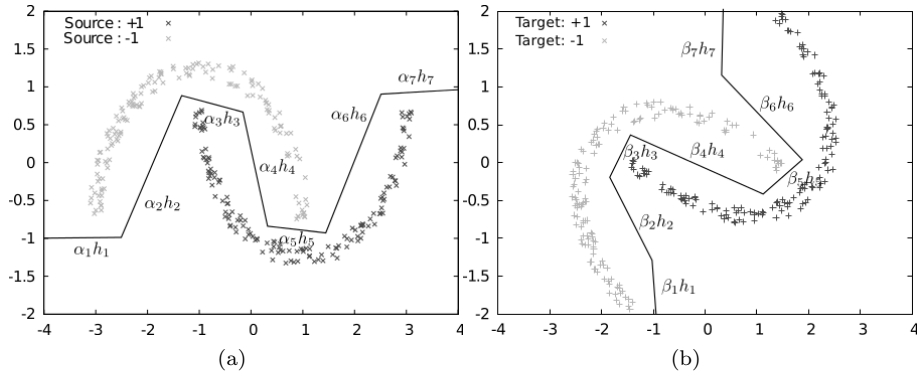
dimensional space. In Satpal & Sarawagi (2007), the authors suggest to train a model maximizing the likelihood over the training sample on a subset of the original features, minimizing the distance between the two distributions. Another approach proposed in Becker et al. (2013) tries to find a task-independent decision boundary in a common space thanks to a non linear mapping of the features in each task. This methods stands rather in a multi-task setting and requires labeled examples in each task, and thus target labels in a two tasks setting.

The second strategy aims at learning new latent features. For example, Florian et al. proposed in Florian et al. (2004) an algorithm that builds a model on the source domain and uses the predicted labels as new features for both source and target data. In Blitzer et al. (2006), the authors introduce the notion of *Structural Correspondence Learning* (SCL). They identify the correspondence between features of the two domains by modeling their correlations. In this case, the principle is to project the examples from both domains into the same low-dimensional space, obtained by a mapping of the original features as done for example with a Principal Component Analysis (PCA). In Ji et al. (2011), the authors use SCL in a Multi-View way. They first choose m bridge features, as does SCL and then learn for each domain a correspondence between these bridge features and the other features. Finally, a Multi-View Principal Component Analysis is applied to learn a low-dimensional space. In Daumé III (2007), a mapping is learned from the original n-dimensional space to a new $3n$-dimensional space, where the first n features are shared by both domains, and the last $n+n$ features are specific to the source and target domains respectively. Note that in this work, the author makes use of a small but labeled set of target data.

Finally, a recent work Morvant et al. (2011, 2012) tries to minimize the $\mathcal{H}$-divergence in order to decrease the generalization target error. The authors propose an approach based on the recent theory of $(\varepsilon, \gamma, \tau)$-good similarity functions Balcan & Blum (2006), Balcan et al. (2008), which aims to project examples in a new space where each dimension represents the similarity to a learning example.

Iterative Self-labeling In order to take advantage of the available unlabeled target data during the learning process, *iterative self-labeling* methods use and label some of them with an hypothesis built from source examples. Such target data are usually called semi-labeled examples. Then, a new classifier is trained taking into account these semi-labeled data, thus considering information from both domains. The main difficulty is to define the way of choosing the considered target points (and the appropriate proportion). Intuitively, a good strategy would retain the semi-labeled target data for which we have the greater confidence in their classification. In Pérez & Sánchez-Montañés (2007), Perez et al. introduce a two-step algorithm: first, a statistical model is learned from the source domain. Then, using an extension of the EM algorithm, they estimate how these source parameters change in the target set. The assumption is that large changes are less likely to occur than small ones. Finally, they use this re-estimated model to learn a classifier for the target domain. Other works have been proposed, based on co-training or self-training methods Blum & Mitchell (1998).

The most famous self-labeling algorithm is certainly DASVM, introduced in Bruzzone & Marconcini (2010), and which was shown to be very competitive. The idea is to iteratively incorporate target samples from T into the learning set to progressively modify an SVM classifier h. Despite good practical results,



(a) (b)

Fig. 1. Illustration of the intuition behind SLDAB on the Moons database where the target data are generated after a rotation of the source examples. Source and target examples are projected in the same N-dimensional space, where the same N weak hypotheses are combined to get the final classifier. The only difference is about the weights applied in the combination. For the source domain S (on the left), the α's parameters are optimized w.r.t. the classification error on S, while for the target distribution T (on the right), the β's parameters are learned w.r.t. the ability of the hypothesis to maximize the margins.

DASVM is not theoretically founded. In a recent work, Habrard et al. (2013) has been proposed a theoretical analysis on the necessary conditions ensuring the good behaviour of a self-labeling process. An algorithm is also introduced, using the $(\varepsilon, \gamma, \tau)$-good similarity functions Balcan & Blum (2006), Balcan et al. (2008), and specifically designed for an application on structured data.

In this paper, we try to take advantage of two categories of approaches in the same time, by introducing an algorithm, SLDAB, which follows an iterative procedure by learning weak classifiers with the help of target examples and finally combines all theses weak hypotheses to obtain a hyperplane in a new space in which are projected both source and target data. We give an intuition on our algorithm in the next section.

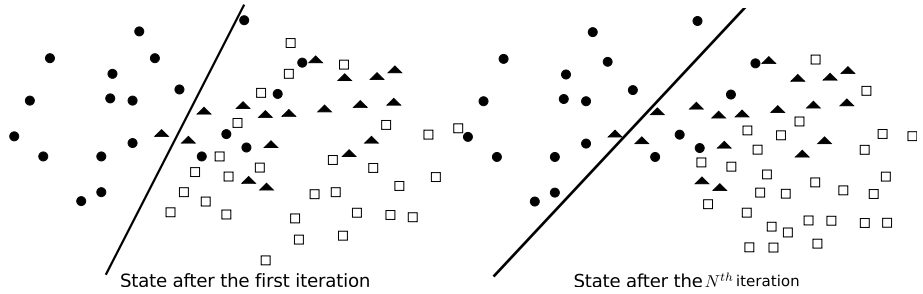
3. Intuition behind SLDAB

Let us recall that boosting aims to iteratively learn weak classifiers h_n (typically, stumps) and to make a linear combinations of all the classifiers regarding their relevance. In the classic supervised classification setting, this relevance depends on the ability of h_n to correctly label source examples from the training set S drawn according to the current distribution D_n^S. As a recall, the final classifier F_S^N, after N iterations, is defined as follows:

$$F_S^N = \sum_{n=1}^{N} \alpha_n h_n(\mathbf{x}),$$

where $\alpha_n = \frac{1}{2} \ln \frac{1-\hat{\epsilon}_n(h_n)}{\hat{\epsilon}_n(h_n)}$, and $\hat{\epsilon}_n = \mathbb{E}_{(x,y)\sim D_n^S}[\ell(h,(x,y))]$.

In a geometric way, F_S^N takes the form of an optimized hyperplane in the space corresponding to the outputs of the classifiers h_n.



(a) (b)

Fig. 2. Illustration of the situation caused by degenerate hypotheses. Figure (a) describes the situation after the first iteration. If no divergence measure is taken into account during the process, we might face a situation illustrated by Figure (b), where the conditions are fulfilled (low source classification error and important target margins), but in which the divergence between the source and target distributions is higher and higher.

In the DA setting, the adaptation of boosting is not straightforward. Indeed, the learning set is made not only of a subset S of labeled source examples, but it also contains a subset T made of unlabeled target samples. In this paper, we aim to **use the same weak learners** to minimize both the empirical error on S while maximizing the margins on T. This strategy consists in projecting both source and target examples in the same N-dimensional space (N still being the number of iterations of the algorithm), while reducing the divergence between the two domains in this new space.

If the learned hypotheses $h_1, \ldots, h_n, \ldots, h_N$ are the same for S and T, note that we aim to optimize different weighting coefficients depending on the domain the example belongs to. Indeed, if it seems natural (and theoretically founded) to keep minimizing the empirical classification error on the labelled source examples, a different strategy has to be applied on the target data for which we do not have access to the labels. We claim that the quality of an hypothesis h_n on the examples from T has to depend on its ability to minimize the proportion of margin violations of the target examples. Therefore, our objective is to optimize a second linear combination $F_T^N = \sum_{n=1}^{N} \beta_n h_n(\mathbf{x})$, where β_n depends both on the example margin and the divergence between S and T induced by h_n.

In this context, SLDAB aims to jointly learn two separator hyperplanes in the same N-dimensional space, common to source and target examples. The geometrical orientation of F_S^N and F_T^N will depend on their ability to minimize respectively the empirical classification error on S and the proportion of margin violations on T. Figure 1 illustrates this idea.

As previously stated, minimizing the classification error on S and maximizing the margins on T is not sufficient. Indeed, designing an algorithm only dedicated to optimize these two criteria may lead to degenerate situations as illustrated in Figure 2. We can see that along the iterations the algorithm tends to increase the divergence between S and T by moving away source and target data, while minimizing source classification errors and target margin violations. This justifies the need of taking into account a divergence measure, **depending on the specific hypothesis** h_n, to prevent us from getting degenerate situations. In

the next four sections, we consider a generic definition of the divergence. We specifically focus on this divergence in Section 7.

4. Definitions and Notations

Let us recall that we dispose of a set S of labeled data (x', y') drawn from a source distribution $\mathcal{S}$ over $Z = X \times Y$, where X is the instance space and $Y = \{-1, +1\}$ is the set of labels, together with a set T of unlabeled examples x drawn from a target distribution $\mathcal{T}$ over X. Let $\mathcal{H}$ be a class of hypotheses and $h_n \in \mathcal{H} : X \to [-1, +1]$ a hypothesis learned from S and T and their associated empirical distribution D_n^S and D_n^T.
We denote by $g_n \in [0, 1]$ a measure of divergence induced by h_n between S and T. Our objective is to take into account g_n in our new boosting scheme so as to penalize hypotheses that do not allow the reduction of the divergence between S and T. To do so, we consider the function $f_{DA} : [-1, +1] \to [-1, +1]$ such that $f_{DA}(h_n(x)) = |h_n(x)| - \lambda g_n$, where $\lambda \in [0, 1]$. $f_{DA}(h_n(x))$ expresses the ability of h_n to not only induce large margins (a large value for $|h_n(x)|$), but also to reduce the divergence between S and T (a small value for g_n). λ plays the role of a trade-off parameter between the importance of the margin and the divergence.

Let $T_n^- = \{x \in T | f_{DA}(h_n(x)) \leq \gamma\}$. If $x \in T_n^- \Leftrightarrow |h_n(x)| \leq \gamma + \lambda g_n$. Therefore, T_n^- corresponds to the set of target points that either violate the margin condition (indeed, if $|h_n(x)| \leq \gamma \Rightarrow |h_n(x)| \leq \gamma + \lambda g_n$) or do not satisfy sufficiently that margin to compensate a large divergence between S and T (i.e. $|h_n(x)| > \gamma$ but $|h_n(x)| \leq \gamma + \lambda g_n$). In the same way, we define $T_n^+ = \{x \in T | f_{DA}(h_n(x)) > \gamma\}$ such that $T = T_n^- \cup T_n^+$. Finally, from T_n^- and T_n^+, we define $W_n^+ = \sum_{x \in T_n^+} D_n^T(x)$ and $W_n^- = \sum_{x \in T_n^-} D_n^T(x)$ such that $W_n^+ + W_n^- = 1$.

Let us remind that the weak assumption presented in Freund & Schapire (1996) states that a classifier h_n is a weak hypothesis over S if it performs at least a little bit better than random guessing, that is $\hat{\epsilon}_n < \frac{1}{2}$, where $\hat{\epsilon}_n$ is the empirical error of h_n over S w.r.t. D_n^S. In this paper, we extend this weak assumption to the DA setting.

Definition 1 (Weak DA Learner). A classifier h_n learned at iteration n from a labeled source set S drawn from $\mathcal{S}$ and an unlabeled target set T drawn from $\mathcal{T}$ is a weak DA learner for T if $\forall \gamma \leq 1$:

1. h_n is a weak learner for S, i.e. $\hat{\epsilon}_n < \frac{1}{2}$.
2. $\hat{L}_n = \mathbb{E}_{x \sim D_n^T}[|f_{DA}(h_n(x))| \leq \gamma] = W_n^- < \frac{\gamma}{\gamma + max(\gamma, \lambda g_n)}$.

Condition 1 means that to adapt from $\mathcal{S}$ to $\mathcal{T}$ using a boosting scheme, h_n must learn something new at each iteration about the source labeling function. Condition 2 takes into account not only the ability of h_n to satisfy the margin γ but also its capacity to reduce the divergence between S and T. From Definition 1, it turns out that:

1. if $max(\gamma, \lambda g_n) = \gamma$, then $\frac{\gamma}{\gamma + max(\gamma, \lambda g_n)} = \frac{1}{2}$ and Condition 2 looks like the weak assumption over the source, except the fact that $\hat{L}_n < \frac{1}{2}$ expresses a margin condition while $\hat{\epsilon}_n < \frac{1}{2}$ considers a classification constraint. Note that if this is true for any hypothesis h_n, it means that the divergence between the

Algorithm 1 SLDAB

Input: a set S of labeled data and a set T of unlabeled data, a number of iterations N, a margin $\gamma \in [0,1]$, a trade-off parameter $\lambda \in [0,1]$, $l = |S|$, $m = |T|$.
Output: two source and target classifiers H_N^S and H_N^T.
Initialization: $\forall (x',y') \in S, D_1^S(x') = \frac{1}{l}$, $\forall x \in T, D_1^T(x) = \frac{1}{m}$.
for $n = 1$ **to** N **do**
Learn a weak DA hypothesis h_n by solving Problem (2).
Compute the divergence value g_n (see Section 7 for details).
$\alpha_n = \frac{1}{2} \ln \frac{1-\hat{\epsilon}_n}{\hat{\epsilon}_n}$ and $\beta_n = \frac{1}{\gamma + \max(\gamma, \lambda g_n)} \ln \frac{\gamma W_n^+}{\max(\gamma, \lambda g_n) W_n^-}$
$\forall (x',y') \in S$, $D_{n+1}^S(x') = D_n^S(x').\frac{e^{-\alpha_n sgn(h_n(x')).y'}}{Z'_n}$.
$\forall x \in T$, $D_{n+1}^T(x) = D_n^T(x).\frac{e^{-\beta_n f_{DA}(h_n(x)).y^n}}{Z_n}$,
where $y^n = sgn(f_{DA}(h_n(x)))$ if $|f_{DA}(h_n(x))| > \gamma$,
$y^n = -sgn(f_{DA}(h_n(x)))$ otherwise,
and Z'_n and Z_n are normalization coefficients.
end for

$$\forall (x',y') \in S, F_N^S(x') = \sum_{n=1}^{N} \alpha_n sgn(h_n(x')),$$

$$\forall x \in T, F_N^T(x) = \sum_{n=1}^{N} \beta_n sgn(h_n(x)).$$

Final source and target classifiers: $H_N^S(x') = sgn(F_N^S(x'))$ and $H_N^T(x) = sgn(F_N^T(x))$.

source and target distributions is rather small, and thus the underlying task looks more like a semi-supervised problem.

2. if $max(\gamma, \lambda g_n) = \lambda g_n$, then the constraint imposed by Condition 2 is stronger (that is $\hat{L}_n < \frac{\gamma}{\gamma + max(\gamma, \lambda g_n)} < \frac{1}{2}$) in order to compensate a large divergence between S and T. In this case, the underlying task requires a domain adaptation process in the weighting scheme.

In the following, we make use of this weak DA assumption to design a new boosting-based DA algorithm, called SLDAB.

5. SLDAB Algorithm

The pseudo-code of SLDAB is presented in Algorithm 1. Starting from uniform distributions over S and T, it iteratively learns a new hypothesis h_n that verifies the weak DA assumption of Definition 1. Note that this task is not trivial. Indeed, while learning a stump (i.e. a one-level decision tree) is sufficient to satisfy the weak assumption of AdaBoost, finding an hypothesis fulfilling Condition 1 on the source and Condition 2 on the target is more complex. To overcome this problem, we present in the following a simple strategy which tends to induce hypotheses that satisfy the weak DA assumption.

First, we generate $\frac{k}{2}$ stumps that satisfy Condition 1 over the source and $\frac{k}{2}$ that fulfill Condition 2 over the target. Then, we seek a convex combination

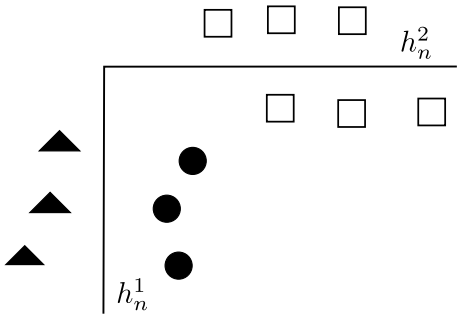


Fig. 3. Illustration of the stumps combination. A single stump would not be sufficient to satisfy the two conditions of Definition 1. This explains why we propose to combine several of them, as illustrated in this figure, where the combination of h_n^1 and h_n^2 allows us to obtain a weak learner for Domain Adaptation, correctly classifying most of the source examples and achieving in the same time large margins, with respect to γ and λg_n, on the target domain.

$h_n = \sum_k \kappa_k h_n^k$ of the k stumps that satisfies simultaneously the two conditions of Definition 1. To do so, we propose to solve the following convex optimization problem:

$$\underset{\kappa}{\operatorname{argmin}} \sum_{z=(x,y)\in S} D_n^S(x) \left[-y \sum_k \kappa_k h_n^k(x) \right]_+ + \sum_{x\in T} D_n^T(x) \left[1 - \left(\sum_k \kappa_k f_{DA}(h_n^k(x)) \right) \right]_+ \quad (2)$$

where $[1-x]_+ = max(0, 1-x)$ is the hinge loss. Solving this optimization problem tends to fulfill Definition 1. Indeed, minimizing the first term of Equation(2) tends to reduce the empirical risk over the source data, while minimizing the second term tends to decrease the number of margin violations over the target data. Using the combination of several stumps thus allows us to satisfy the two conditions, as illustrated in Figure 3.

Note that in order to generate a simple weak DA learner, we start the process with $k = 2$. If the optimized combination does not satisfy the weak DA assumption, we draw a new set of k stumps. If the weak DA assumption is not satisfied after several tries, we increase the dimension of the induced hypothesis h_n. If despite the increase of k (limited to a given value), no hypothesis is able to fulfill the DA weak learner conditions, the adaptation is stopped. The pseudo-code of the algorithm seeking for weak DA hypotheses is described in Algorithm 2.

Once h_n has been learned, the weights of the labeled and unlabeled data are modified according to two different update rules. Those of source examples are updated using the same strategy as that of AdaBoost. Regarding the target examples, their weights are changed according to their location in the space. If a target example x does not satisfy the condition $f_{DA}(h_n(x)) > \gamma$, a pseudo-class $y^n = -sgn(f_{DA}(h_n(x)))$ is assigned to x that simulates a misclassification. Note that such a decision has a geometrical interpretation: it means that we exponentially increase the weights of the points located in an extended margin band of width $\gamma + \lambda g_n$. If x is outside this band, a pseudo-class $y^n = sgn(f_{DA}(h_n(x)))$ is assigned leading to an exponential decrease of $D_n^T(x)$ at the next iteration.

Algorithm 2 Conception of SLDAB weak learner

Input: a set S of size l of source labeled data, a set T of size m of target unlabeled data, a number k of stumps to combine, a constant K_{MAX}, a constant I_{MAX}.
Output: a stumps combination.
$k = 2$
while $k < K_{\text{MAX}}$ **do**
  for $i = 0$ **to** I_{MAX} **do**
    for $j = 0$ **to** k **do**
      if j is odd **then**
        Generate a stump fulfilling Condition 1 of Definition 1.
      else
        Generate a stump satisfying Condition 2 of Definition 1.
      end if
    end for
    Solve Problem 2.
    if The learned classifier fulfills the two conditions of Definition 1 **then**
      Return the classifier.
    end if
  end for
end while
Interrupt iterative process.

6. Theoretical Analysis

In this section, we present a theoretical analysis of our approach. First, we derive a generalization bound on the loss empirically minimized in Algorithm 2. This bound is deduced from the proof of the algorithmic robustness of this algorithm. Then, we focus on theoretical guarantees of SLDAB. Recall that the goodness of a hypothesis h_n is measured by its ability not only to correctly classify the source examples but also to classify the unlabeled target data with a large margin w.r.t. the classifier-induced divergence g_n. Provided that the weak DA constraints of Definition 1 are satisfied, the standard results of AdaBoost still hold on $\mathcal{S}$ (because of Condition 1). In the following, we show that the loss $\hat{L}_{H_N^T}$, which represents after N iterations the proportion of margin violations over T (w.r.t. the successive divergences g_n), also decreases with N.

6.1. Consistency Guarantees of Algorithm 2

We present a theoretical analysis of Algorithm 2 which aims at learning a convex combination of stumps. Let us remind that this combination is the solution of the following minimization problem:

$$\min_{\kappa} \sum_{z=(x,y)\in S} D_n^S(x') \left[-y \sum_k \kappa_k h_n^k(x)\right]_+ + \sum_{x\in T} D_n^T(x) \left[1 - \sum_k \kappa_k \left(f_{DA}[h_n^k(x)] - \gamma\right)\right]_+$$
$$= \min_{\kappa} \sum_{z=(x,y)\in S} D_n^S(x)\ell_1(\kappa, z) + \sum_{x\in T} D_n^T(x)\ell_2(\kappa, x) \quad (3)$$

where:

- $\ell_1(\kappa, z = (x, y)) = \left[-y \sum_k \kappa_k h_n^k(x)\right]_+$
- and $\ell_2(\kappa, x) = \left[1 - \sum_k \kappa_k \left(f_{DA}[h_n^k(x)] - \gamma\right)\right]_+$.

The objective of this theoretical analysis is to derive a generalization bound on the true loss $\mathcal{R}^{\ell_1,\ell_2} = \mathbb{E}_{z\in Z}\ell_1(\kappa, z) + \mathbb{E}_{x\in X}\ell_2(\kappa, x)$ according to the empirical loss $\hat{\mathcal{R}}^{\ell_1,\ell_2} = \min_{\kappa} \sum_{z=(x,y)\in S} D_n^S(x)\ell_1(\kappa, z) + \sum_{x\in T} D_n^T(x)\ell_2(\kappa, x)$, minimized by Algorithm 2 w.r.t. the training source examples S and the training target examples T. Said differently, we aim at analyzing the convergence (in terms of the number of training examples, that is $|S| + |T|$ - where $|.|$ is the cardinality of the set) of the predicted combinaison of stumps as an estimate of the true combination. This can be done into two steps: (i) Prove the algorithmic robustness of Problem 3; (ii) use this robustness property to derive a generalization bound on $\mathcal{R}^{\ell_1,\ell_2}$.

In the following, we propose a new definition of the algorithmic robustness as an extension to the domain adaptation framework of that of introduced in Xu & Mannor (2010*b*).

Definition 2 (Algorithmic Robustness). Let $\mathcal{A}(\kappa, \ell_1, \ell_2)$ be an algorithm of first argument κ trained from $|S|$ labeled source examples $\in S$ w.r.t. to a loss function ℓ_1 and $|T|$ unlabeled target examples $\in T$ w.r.t. to a loss functions ℓ_2 . $\mathcal{A}(\kappa, \ell_1, \ell_2)$ is $(M_1, M_2, \epsilon(\cdot))$-robust, for $M_1, M_2 \in \mathbb{N}$ and $\epsilon(\cdot,\cdot) : (Z^{|S|} \times X^{|T|}) \to \mathbb{R}$, if $Z = \mathcal{X} \times \mathcal{Y}$ (resp. $\mathcal{X}$) can be partitioned into M_1 (resp. M_2) disjoint sets, denoted by $\{C_i\}_{i=1}^{M_1}$ (resp. $\{D_j\}_{j=1}^{M_2}$), such that the following holds for all $S \in Z^{|S|}$ and all $T \in X^{|T|}$:

$$\forall z \in S, \forall z' \in Z, \forall x \in T, \forall x' \in X, \forall i \in [M_1], \forall j \in [M_2] :$$

$$\text{if } z, z' \in C_i, \text{if } x, x' \in D_j, \text{then } |\ell_1(\kappa, z) + \ell_2(\kappa, x) - \ell_1(\kappa, z') - \ell_2(\kappa, x')| \leq \epsilon(S, T).$$

Roughly speaking, an algorithm is robust if for any source **test** example $z' \in Z$ (resp. target **test** example $x' \in X$) falling in the same subset as a **training** example $z \in S$ (resp. $x \in T$), the gap between the losses ℓ_1 (resp. ℓ_2) associated with z and z' (resp. x and x') is bounded. In other words, robustness characterizes the capability of an algorithm to **perform similarly on close train and test instances**. The closeness of the instances is based on a partitionning of Z and X in the sense that two examples are close if they belong to the same region. In general, the partition is based on the notion of covering number Kolmogorov & Tikhomirov (1961) allowing one to cover Z (resp. X) by regions where the distance/norm between two elements in the same region are no more than a fixed quantity ρ_1 (resp. ρ_2). The covering over the labeled set Z is built as follows: first we consider a ρ_1-cover over the instance X, then we partition Z by considering one ρ_1-cover over X for the positive instances and another ρ_1-cover over X for the negative instances ensuring that two examples in the same region belong to the same class and the distance between them is no more than ρ_1 (see Xu & Mannor (2010*b*, 2012*b*) for details).

Theorem 1. Given a partition of Z into M_1 subsets $\{C_i\}$ such that the two labeled source examples $z = (x_S, y), z' = (x'_S, y') \in C_i$ with $y = y'$ and given a partition of X into M_2 subsets $\{D_j\}$ such that the two unlabeled target examples $x_T, x'_T \in D_j$, the optimization problem 3 with constraints $\sum_k \kappa_k = 1$ (convex

combination of the stumps) is $(M_1, M_2, \epsilon(S,T))$-robust with $\epsilon(S,T) = \rho_1 + \rho_2$, where $\rho_1 = \sup_{x_S, x'_S \in C_i} ||x_S - x'_S||$ and $\rho_2 = \sup_{x_T, x'_T \in D_j} ||x_T - x'_T||$.

Proof of Theorem 1

$$
\begin{aligned}
&|\ell_1(\kappa, z) + \ell_2(\kappa, z) - \ell_1(\kappa, z') + \ell_2(\kappa, x')| \\
\leq &|\ell_1(\kappa, z) - \ell_1(\kappa, z')| + |\ell_2(\kappa, z) - \ell_2(\kappa, x')| \\
\leq &\left|\sum_k \kappa_k (h_n^k(x_S) - h_n^k(x'_S))\right| + \left|\sum_k \kappa_k (f_{DA}[h_n^k(x_T)] - f_{DA}[h_n^k(x'_T)])\right| \qquad (4) \\
\leq &\sum_k |\kappa_k| \cdot \left|h_n^k(x_S) - h_n^k(x'_S)\right| + \sum_k |\kappa_k| \cdot \left|f_{DA}[h_n^k(x_T)] - f_{DA}[h_n^k(x'_T)]\right| \qquad (5) \\
\leq &\sum_k |\kappa_k| \cdot ||x_S - x'_S|| + \sum_k |\kappa_k| \cdot ||x_T - x'_T|| \qquad (6) \\
\leq &\rho_1 + \rho_2 \qquad (7)
\end{aligned}
$$

We get Inequality (4) from the 1-lipschitzness of the hinge loss; Inequality (5) comes from the classical triangle inequality; The first term of Inequality (6) is due to the 1-lipschitzness of $h_n^k(x_S)$. Indeed, since $h_n^k(x_S)$ comes from a stump it takes the form of $x_S - \sigma$. Therefore, $|h_n^k(x_S) - h_n^k(x'_S)| = 1.|x_S - x'_S|$. The second term of Inequality (6) is due to the 1-lipschitzness of $f_{DA}[h_n^k(x_T)] = |h_n^k(x_T)| - \lambda g_n$. Indeed, $|f_{DA}[h_n^k(x_T)] - f_{DA}[h_n^k(x'_T)]| = ||x_T| - |x'_T|| \leq 1.|x_T - x'_T|$. Finally, we get Inequality (7) due to the constraint of convex combinaison and $\kappa_k \geq 0$. □

We now give a PAC generalization bound on the true loss making use of the previous robustness result. Let $\mathcal{R}^{\ell_1,\ell_2} = \mathbb{E}_{z\sim\mathcal{S}}\ell_1(\kappa, z) + \mathbb{E}_{x\sim\mathcal{T}}\ell_2(\kappa, x)$ be the true loss w.r.t. the unknown distributions $\mathcal{S}$ and $\mathcal{T}$.
Let $\hat{\mathcal{R}}^{\ell_1,\ell_2} = \min_{\kappa} \sum_{z=(x,y)\in S} D_n^S(x)\ell_1(\kappa, z) + \sum_{x\in T} D_n^T(x)\ell_2(\kappa, x)$ be the empirical loss over the training sets S and T. Based on the results of Xu & Mannor (2010*b*, 2012*b*), the proof requires the use of the following concentration inequality over multinomial random variables allowing one to capture statistical information coming from the different regions of the partitions of $\mathcal{Z}$ and $\mathcal{X}$.

Proposition 2. van der Vaart & Wellner (1996)
Let $(|N_1|, \ldots, |N_M|)$ an i.i.d. multinomial random variable with parameters $N = \sum_{i=1}^M |N_i|$ and $(p(C_1), \ldots, p(C_M))$. By the Bretagnolle-Huber-Carol inequality we have: $\Pr\left\{\sum_{i=1}^M \left|\frac{|N_i|}{N} - p(C_i)\right| \geq \lambda\right\} \leq 2^M \exp\left(\frac{-N\lambda^2}{2}\right)$, hence with probability at least $1 - \delta$,

$$\sum_{i=1}^M \left|\frac{N_i}{N} - p(C_i)\right| \leq \sqrt{\frac{2M\ln 2 + 2\ln(1/\delta)}{N}}. \qquad (8)$$

We are now able to present our generalization bound thanks to the following theorem.

Theorem 3. Considering that problem 3 is $(M_1, M_2, \epsilon(S,T))$-robust, for any $\delta > 0$ with probability at least $1 - \delta$, we have:

$$|\mathcal{R}^{\ell_1,\ell_2} - \hat{\mathcal{R}}^{\ell_1,\ell_2}| \leq 2\max(\rho_1, \rho_2) + 2\max(B_1, B_2)\sqrt{\frac{2\max(M_1, M_2)\ln 2 + 2\ln(1/\delta)}{\min(|S|, |T|)}},$$

where B_1 (resp. B_2) is an upper bound of the loss ℓ_1 (resp. ℓ_2).

Note that in robustness bounds, the cover radius ρ_1 (resp. ρ_2) can be made arbitrarily small at the expense of larger values of M_1 (resp. M_2). As M_1 and M_2 appear in the second term, which decreases to 0 when $\min(|S|, |T|)$ tends to infinity, this bound provides a standard $O(1/\sqrt{\min(|S|, |T|)})$ asymptotic convergence.

Proof of Theorem 3
Inspired from Xu & Mannor (2010*b*, 2012*b*)

$$
\begin{aligned}
&\left|\mathcal{R}^{\ell_1,\ell_2} - \hat{\mathcal{R}}^{\ell_1,\ell_2}\right| \\
\leq &\left|\mathbb{E}_{z\sim\mathcal{S}}\ell_1(\kappa,z) - \sum_{z=(x,y)\in S} D_n^S(x)\ell_1(\kappa,z)\right| + \left|\mathbb{E}_{x\sim\mathcal{T}}\ell_2(\kappa,x) - \sum_{x\in T} D_n^T(x)\ell_2(\kappa,x)\right| \\
= &\left|\sum_{i=1}^{M_1}\mathbb{E}_{z\sim\mathcal{S}}\left(\ell_1(\kappa,z)|z\in C_i\right)p(C_i) - \sum_{z\in S} D_n^S(x)\ell_1(\kappa,z)\right| \\
&+ \left|\sum_{j=1}^{M_2}\mathbb{E}_{x\sim\mathcal{T}}\left(\ell_2(\kappa,x)|x\in D_j\right)p(D_j) - \sum_{x\in T} D_n^T(x)\ell_2(\kappa,x)\right| \\
\leq &\left|\sum_{i=1}^{M_1}\mathbb{E}_{z\sim\mathcal{S}}\left(\ell_1(\kappa,z)|z\in C_i\right)p(C_i) - \sum_{i=1}^{M_1}\mathbb{E}_{z\sim\mathcal{S}}(\ell_1(\kappa,z)|z\in C_i)\frac{|N_i|}{|S|}\right| \\
&+ \left|\sum_{i=1}^{M_1}\mathbb{E}_{z\sim\mathcal{S}}(\ell_1(\kappa,z)|z\in C_i)\frac{|N_i|}{|S|} - \sum_{z\in S} D_n^S(x)\ell_1(\kappa,z)\right| \qquad (9) \\
&+ \left|\sum_{j=1}^{M_2}\mathbb{E}_{x\sim\mathcal{T}}\left(\ell_2(\kappa,x)|x\in D_j\right)p(D_j) - \sum_{j=1}^{M_2}\mathbb{E}_{x\sim\mathcal{T}}(\ell_2(\kappa,x)|x\in D_j)\frac{|N_j|}{|T|}\right| \\
&+ \left|\sum_{j=1}^{M_2}\mathbb{E}_{x\sim\mathcal{T}}(\ell_2(\kappa,x)|x\in D_j)\frac{|N_j|}{|T|} - \sum_{x\in T} D_n^T(x)\ell_2(\kappa,x)\right|
\end{aligned}
$$

$$
\begin{aligned}
&= \left| \sum_{i=1}^{M_1} \mathbb{E}_{z\sim\mathcal{S}} \left(\ell_1(\kappa, z) | z \in C_i\right) \left| \frac{|N_i|}{|S|} - p(C_i) \right| \right| + \left| \sum_{j=1}^{M_2} \mathbb{E}_{x\sim\mathcal{T}} \left(\ell_2(\kappa, x) | x \in D_j\right) \left| \frac{|N_j|}{|T|} - p(D_j) \right| \right| \\
&\quad + \left| \sum_{i=1}^{M_1} \sum_{z_j \in C_i} \mathbb{E}_{z\sim\mathcal{S}} \left(\ell_1(\kappa, z) | z \in C_i\right) - \sum_{i=1}^{M_1} \sum_{z_j \in C_i} D_n^S(x_j) \ell_1(\kappa, z_j) \right| \\
&\quad + \left| \sum_{j=1}^{M_2} \sum_{x_i \in D_j} \mathbb{E}_{x\sim\mathcal{T}} \left(\ell_2(\kappa, x) | x \in D_j\right) - \sum_{j=1}^{M_2} \sum_{x_i \in D_j} D_n^T(x_i) \ell_2(\kappa, x_i) \right| \\
&\leq \left| \max_{z\sim\mathcal{S}} \ell_1(\kappa, z) \sum_{i=1}^{M_1} \left| \frac{|N_i|}{|S|} - p(C_i) \right| \right| + \left| \max_{x\sim\mathcal{T}} \ell_2(\kappa, x) \sum_{j=1}^{M_2} \left| \frac{|N_j|}{|T|} - p(D_j) \right| \right| \\
&\quad + \left| \sum_{i=1}^{M_1} \sum_{j \in N_i} \max_{z \in C_i} |\ell_1(\kappa, z_j) - \ell_1(\kappa, z)| \right| + \left| \sum_{i=1}^{M_2} \sum_{j \in N_i} \max_{x \in D_i} |\ell_2(\kappa, x_j) - \ell_2(\kappa, x)| \right| \\
&\leq \rho_1 + B_1 \sqrt{\frac{2M_1 \ln 2 + 2\ln(1/\delta)}{|S|}} + \rho_2 + B_2 \sqrt{\frac{2M_2 \ln 2 + 2\ln(1/\delta)}{|T|}} \qquad (10) \\
&\leq 2\max(\rho_1, \rho_2) + 2\max(B_1, B_2) \sqrt{\frac{2\max(M_1, M_2) \ln 2 + 2\ln(1/\delta)}{\min(|S|, |T|)}}.
\end{aligned}
$$

Inequality 9 is due to the triangle inequality. Inequality 10 comes from the application of Proposition 2 and Theorem 1.

6.2. Upper Bound on the Empirical Loss of SLDAB

Theorem 4. Let $\hat{L}_{H_N^T}$ be the proportion of target examples of T with a margin smaller than γ w.r.t. the divergences g_n $(n = 1 \dots N)$ after N iterations of SLDAB:

$$\hat{L}_{H_N^T} = \mathbb{E}_{x\sim T}[\mathbf{y}\mathbf{F}_\mathbf{N}^\mathbf{T}(x) < 0] \leq \tfrac{1}{|T|} \textstyle\sum_{x\sim T} e^{-\mathbf{y}\mathbf{F}_\mathbf{N}^\mathbf{T}(x)} = \prod_{n=1}^{N} Z_n,$$

where $\mathbf{y} = (y^1, \dots, y^n, \dots, y^N)$ is the vector of pseudo-classes and $\mathbf{F}_\mathbf{N}^\mathbf{T}(x) = (\beta_1 f_{DA}(h_1(x)), \dots, \beta_N f_{DA}(h_N(x)))$.

Proof. The proof is the same as that of Freund & Schapire (1996) except that $\mathbf{y}$ is the vector of pseudo-classes (which depend on λg_n and γ) rather than the vector of true labels. □

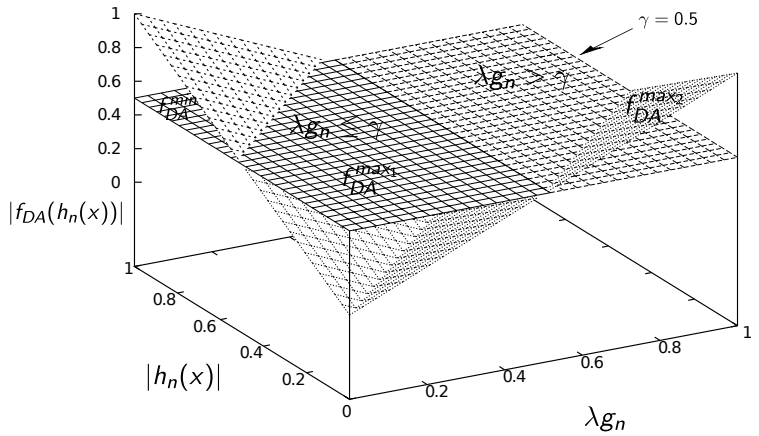


Fig. 4. Upper bounds of the components of Z_n for an arbitrary value $\gamma = 0.5$. When $x \in T_n^+$, the upper bound is obtained with $|f_{DA}| = \gamma$ (see the plateau f_{DA}^{min}). When $x \in T_n^-$, we get the upper bound with $\max(\gamma, \lambda g_n)$, that is either γ when $\lambda g_n \leq \gamma$ (see $f_{DA}^{max_1}$) or λg_n otherwise (see $f_{DA}^{max_2}$).

6.3. Optimal Confidence Values

Theorem 4 suggests the minimization of each Z_n to reduce the empirical loss $\hat{L}_{H_N^T}$ over T. To do this, let us rewrite Z_n as follows:

$$Z_n = \sum_{x \in T_n^-} D_n^T(x) e^{-\beta_n f_{DA}(h_n(x)) y^n} + \sum_{x \in T_n^+} D_n^T(x) e^{-\beta_n f_{DA}(h_n(x)) y^n}. \quad (11)$$

The two terms of the right-hand side of Equation(11) can be upper bounded as follows:

- $\forall x \in T_n^+$, $D_n^T(x) e^{-\beta_n f_{DA}(h_n(x)) y^n} \leq D_n^T(x) e^{-\beta_n \gamma}$.
- $\forall x \in T_n^-$, $D_n^T(x) e^{-\beta_n f_{DA}(h_n(x)) y^n} \leq D_n^T(x) e^{\beta_n max(\gamma, \lambda g_n)}$.

Figure 4 gives a geometrical explanation of these upper bounds. When $x \in T_n^+$, the weights are decreased. We get an upper bound by taking the smallest drop, that is $f_{DA}(h_n(x)) y^n = |f_{DA}| = \gamma$ (see f_{DA}^{min} in Figure 4). On the other hand, if $x \in T_n^-$, we get an upper bound by taking the maximum value of f_{DA} (i.e. the largest increase). We differentiate two cases: (i) when $\lambda g_n \leq \gamma$, the maximum is γ (see $f_{DA}^{max_1}$), (ii) when $\lambda g_n > \gamma$, Figure 4 shows that one can always find a configuration where $\gamma < f_{DA} \leq \lambda g_n$. In this case, $f_{DA}^{max_2} = \lambda g_n$, and we get the upper bound with $|f_{DA}| = \max(\gamma, \lambda g_n)$.

Plugging the previous upper bounds in Equation(11), we get:

$$Z_n \leq W_n^+ e^{-\beta_n \gamma} + W_n^- e^{\beta_n \max(\gamma, \lambda g_n)} = \tilde{Z}_n. \quad (12)$$

Deriving the previous convex combination w.r.t. β_n and equating to zero, we get the optimal values for β_n in Equation(11)[1]:

[1] Note that the approximation $\tilde{Z}_n$ used in Equation(12) is essentially a linear upper bound of Equation(11) on the range $[-1; +1]$. Clearly, other upper bounds which give a tighter approximation could be used instead (see Schapire & Singer (1999) for more details).

$$\frac{\partial \tilde{Z}_n}{\beta_n} = 0 \Rightarrow \max(\gamma, \lambda g_n) W_n^- e^{\beta_n \max(\gamma, \lambda g_n)} = \gamma W_n^+ e^{-\beta_n \gamma}$$

$$\Rightarrow \qquad \beta_n = \frac{1}{\gamma + \max(\gamma, \lambda g_n)} \ln \frac{\gamma W_n^+}{\max(\gamma, \lambda g_n) W_n^-}. \tag{13}$$

It is important to note that β_n is computable if

$$\frac{\gamma W_n^+}{\max(\gamma, \lambda g_n) W_n^-} \geq 1 \Leftrightarrow \gamma(1 - W_n^-) \geq \max(\gamma, \lambda g_n) W_n^- \Leftrightarrow W_n^- < \frac{\gamma}{\gamma + max(\gamma, \lambda g_n)},$$

that is always true because h_n is a weak DA hypothesis and satisfies Condition 2 of Definition 1. Moreover, from Equation(13), it is worth noting that β_n gets smaller as the divergence gets larger. In other words, a hypothesis h_n of weights W_n^+ and W_n^- (which depend on the divergence g_n) will have a greater confidence than a hypothesis $h_{n'}$ of same weights $W_{n'}^+ = W_n^+$ and $W_{n'}^- = W_n^-$ if $g_n < g_{n'}$.

Let $\max(\gamma, \lambda g_n) = c_n \times \gamma$, where $c_n \geq 1$. We can rewrite Equation(13) as follows:

$$\beta_n = \frac{1}{\gamma(1 + c_n)} \ln \frac{W_n^+}{c_n W_n^-}, \tag{14}$$

and Condition 2 of Definition 1 becomes $W_n^- < \frac{1}{1+c_n}$.

6.4. Convergence of the Empirical Loss

The following theorem shows that, provided the weak DA constraint on T is fulfilled (that is, $W_n^- < \frac{1}{1+c_n}$), Z_n is always smaller than 1 that leads (from Theorem 4) to a decrease of the empirical loss $\hat{L}_{H_N^T}$ with the number of iterations.

Theorem 5. If H_N^T is the linear combination produced by SLDAB from N weak DA hypotheses, then $\lim_{N \to \infty} \hat{L}_{H_N^T} = 0$.

Proof. Plugging Equation(14) into Equation(12) we get:

$$Z_n \leq W_n^+ \left(\frac{c_n W_n^-}{W_n^+} \right)^{\frac{1}{(1+c_n)}} + W_n^- \left(\frac{W_n^+}{c_n W_n^-} \right)^{\frac{c_n}{(1+c_n)}} \tag{15}$$

$$= \left(W_n^+\right)^{\frac{c_n}{(1+c_n)}} \left(W_n^-\right)^{\frac{1}{(1+c_n)}} \left(c_n^{\frac{1}{(1+c_n)}} + c_n^{-\frac{c_n}{(1+c_n)}} \right)$$

$$= \left(W_n^+\right)^{\frac{c_n}{(1+c_n)}} \left(W_n^-\right)^{\frac{1}{(1+c_n)}} \left(\frac{c_n + 1}{c_n^{\frac{c_n}{(1+c_n)}}} \right)$$

$$= u_n \times v_n \times w_n, \tag{16}$$

where $u_n = (W_n^+)^{\frac{c_n}{(1+c_n)}}$, $v_n = (W_n^-)^{\frac{1}{(1+c_n)}}$ and $w_n = \left(\frac{c_n+1}{c_n^{\frac{c_n}{(1+c_n)}}} \right)$. Computing the derivative of u_n, v_n and w_n w.r.t. c_n, we get

$$\frac{\partial u_n}{\partial c_n} = \frac{\ln W_n^+}{(c_n+1)^2}\left(W_n^+\right)^{\frac{c_n}{(1+c_n)}}, \frac{\partial v_n}{\partial c_n} = -\frac{\ln W_n^-}{(c_n+1)^2}\left(W_n^-\right)^{\frac{1}{(1+c_n)}}, \frac{\partial w_n}{\partial c_n} -\frac{\ln c_n}{(c_n+1)^2}\frac{c_n+1}{c_n^{\frac{c_n}{(1+c_n)}}}.$$

We deduce that

$$\begin{aligned}\frac{\partial Z_n}{\partial c_n} &= (\frac{\partial u_n}{\partial c_n} \times v_n + \frac{\partial v_n}{\partial c_n} \times u_n) \times w_n + \frac{\partial w_n}{\partial c_n} \times u_n \times v_n \\ &= \left(W_n^+\right)^{\frac{c_n}{(1+c_n)}} \times \left(W_n^-\right)^{\frac{1}{(1+c_n)}} \times \left(\frac{c_n+1}{c_n^{\frac{c_n}{(1+c_n)}}}\right) \times \frac{1}{(c_n+1)^2} \times \left(\ln W_n^+ - \ln W_n^- - \ln c_n\right) \\ &= \left(W_n^+\right)^{\frac{c_n}{(1+c_n)}} \times \left(W_n^-\right)^{\frac{1}{(1+c_n)}} \times \frac{c_n^{\frac{-c_n}{(1+c_n)}}}{c_n+1} \times \left(\ln W_n^+ - \ln W_n^- - \ln c_n\right).\end{aligned}$$

The first three terms of the previous equation are positive. Therefore,

$$\frac{\partial Z_n}{\partial c_n} > 0 \Leftrightarrow \ln W_n^+ - \ln W_n^- - \ln c_n > 0 \Leftrightarrow W_n^- < \frac{1}{c_n+1},$$

that is always true because of the weak DA assumption. Therefore, $Z_n(c_n)$ is a monotonic increasing function over $[1, \frac{W_n^+}{W_n^-}[$, with:

–$Z_n < 2\sqrt{W_n^+ W_n^-}$ (standard result of AdaBoost) when $c_n = 1$,

–and $\lim_{c_n \to \frac{W_n^+}{W_n^-}} Z_n = 1.$

Therefore, $\forall n$, $Z_n < 1$
$\Leftrightarrow \lim_{N\to\infty} \hat{L}_{H_N^T} < \lim_{N\to\infty} \prod_{n=1}^N Z_n = 0.$ □

Let us now give some insight about the nature of the convergence of $\hat{L}_{H_N^T}$. A hypothesis h_n is DA weak if $W_n^- < \frac{1}{1+c_n} \Leftrightarrow c_n < \frac{W_n^+}{W_n^-} \Leftrightarrow c_n = \tau_n \frac{W_n^+}{W_n^-}$ with $\tau_n \in]\frac{W_n^-}{W_n^+}; 1[$. τ_n measures how close is h_n to the weak assumption requirement. Note that β_n gets larger as τ_n gets smaller. From Equation(16) and $c_n = \tau_n \frac{W_n^+}{W_n^-}$ (that is $W_n^- = \frac{\tau_n}{\tau_n + c_n}$), we get:

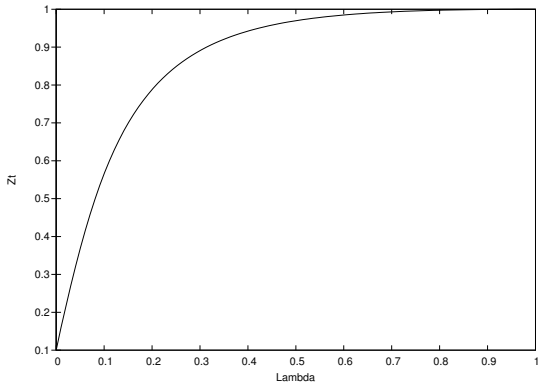


Fig. 5. Evolution of $\ln Z_n$ w.r.t. τ_n.

$$
\begin{aligned}
Z_n &< \left(W_n^+\right)^{\frac{c_n}{(1+c_n)}} \left(W_n^-\right)^{\frac{1}{(1+c_n)}} \left(\frac{c_n+1}{c_n^{\frac{c_n}{(1+c_n)}}}\right) \\
&= \left(1-\frac{\tau_n}{\tau_n+c_n}\right)^{\frac{c_n}{(1+c_n)}} \left(\frac{\tau_n}{\tau_n+c_n}\right)^{\frac{1}{(1+c_n)}} \left(\frac{c_n+1}{c_n^{\frac{c_n}{(1+c_n)}}}\right) \\
&= \frac{c_n^{\frac{c_n}{(1+c_n)}}}{(\tau_n+c_n)^{\frac{c_n}{(1+c_n)}}} \cdot \frac{\tau_n^{\frac{1}{(1+c_n)}}}{(\tau_n+c_n)^{\frac{1}{(1+c_n)}}} \cdot \frac{c_n+1}{c_n^{\frac{c_n}{(1+c_n)}}} \\
&= \left(\frac{\tau_n^{\frac{1}{1+c_n}}}{\tau_n+c_n}\right)(c_n+1).
\end{aligned}
$$

We deduce that

$$
\prod_{n=1}^{N} Z_n = exp \sum_{n=1}^{N} \ln Z_n \leq exp \sum_{n=1}^{N} \left(\ln\left(\left(\frac{\tau_n^{\frac{1}{1+c_n}}}{\tau_n+c_n}\right)(c_n+1)\right)\right) = exp \sum_{n=1}^{N} \left(\frac{1}{1+c_n}\ln\tau_n + \ln(\frac{c_n+1}{\tau_n+c_n})\right).
$$

Theorem 5 tells us that the term between brackets is negative (that is $\ln Z_n < 0, \forall Z_n$). Therefore, the empirical loss decreases exponentially fast towards 0 with the number of iterations N. Moreover, let us study the behaviour of $\ln Z_n$ w.r.t. τ_n. Since Z_n is a monotonic increasing function of c_n over $[1, \frac{W_n^+}{W_n^-}[$, it is also a monotonic increasing function of τ_n over $[\frac{W_n^-}{W_n^+}; 1[$. In other words, the smaller τ_n the faster the convergence of the empirical loss $\hat{L}_{H_N^T}$. Figure 5 illustrates this claim for an arbitrarily selected configuration of W_n^+ and W_n^-. It shows that $\ln Z_n$, and thus $\hat{L}_{H_N^T}$, decreases exponentially fast with τ_n.

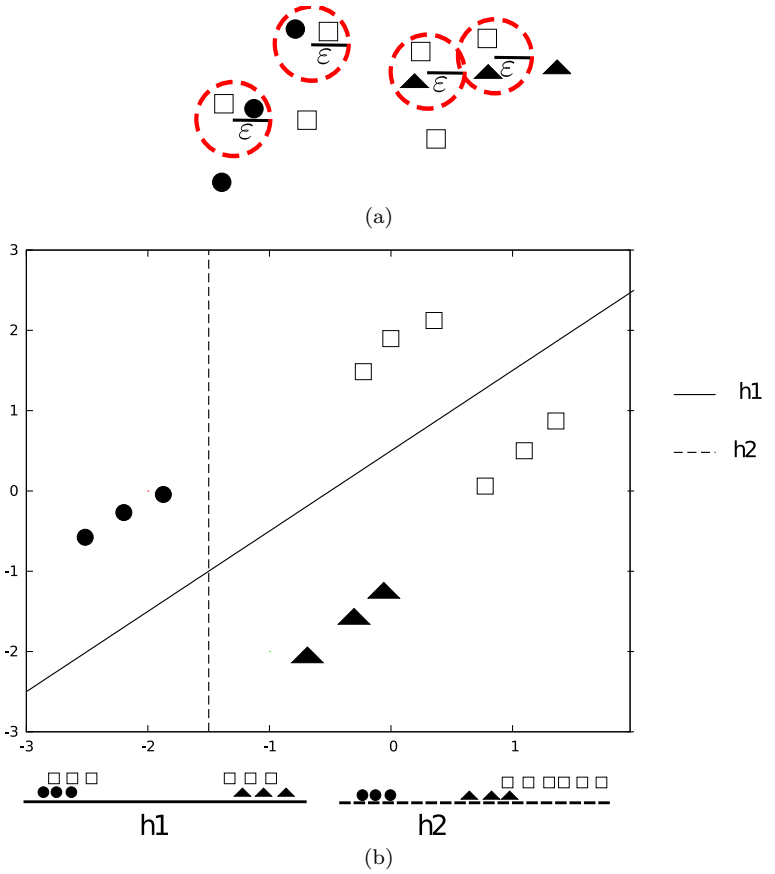
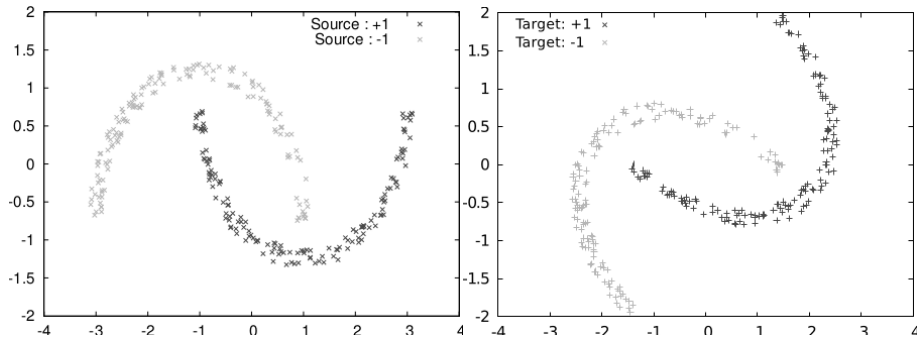


Fig. 6. Illustration of PV benefit in our divergence measure. The source examples are here represented by black points, while target ones are in white. In Figure (a), a source example is matched with a target one, with respect to the Euclidian distance, when they are closer than ε. In Figure (b), the two hypotheses h_1 and h_2 correctly separate source data. However, when comparing the values returned by each of the hypotheses (as indicated at the bottom of the figure), those of h_1 allow us to match all source and target examples, while those of h_2 do not. Indeed, h_1 correctly separates the two target classes, while h_2 gives all target examples the same label.

7. Measure of Divergence

The theoretical results in DA (e.g. Ben-David et al. (2010), Mansour et al. (2009)) state that a good adaptation is possible when the mismatch between the two distributions is small while maintaining a good accuracy on the source. In our algorithm, the latter condition is satisfied via the use of a standard boosting scheme. Concerning the mismatch, we inserted in our framework a measure of divergence g_n, induced by h_n. An important issue of SLDAB is the definition of this measure. A straightforward solution may consist in computing a divergence with respect to the considered *class of hypotheses*, like the well-known



(a) (b)

Fig. 7. Examples from the MOONS database. Figure (a) describes examples from the source domain, while Figure (b) contains data from the target domain, obtained after a 30° rotation.

$\mathcal{H}$-divergence[2] Ben-David et al. (2010). We claim that such a strategy is not suited to our framework because SLDAB rather aims at evaluating the discrepancy induced by a *specific classifier* h_n. We propose to consider a divergence g_n able to both evaluate the mismatch between the source and target data and avoid degenerate hypotheses.

Algorithm 3 Computation of $\hat{PV}(S,T)$.

Input: $S = \{x'_1, \ldots, x'_n\}$, $T = \{x_1, \ldots, x_m\}$, $\epsilon > 0$ and a distance d

1. Define the graph $\hat{G} = (\hat{V} = (\hat{A}, \hat{B}), \hat{E})$ where $\hat{A} = \{x'_i \in S\}$ and $\hat{B} = \{x_j \in T\}$, Connect an edge $e_{ij} \in \hat{E}$ if $d(x'_i, x_j) \leq \epsilon$
2. Compute the maximum matching on $\hat{G}$
3. S_u and T_u are the number of unmatched vertices in S and T respectively
4. Output $\hat{PV}(S,T) = \frac{1}{2}(\frac{S_u}{n} + \frac{T_u}{m}) \ \in [0,1]$

For the first objective, we use the recent *Perturbed Variation* measure Harel & Mannor (2012) that evaluates the discrepancy between two distributions while allowing small permitted variations assessed by a parameter $\epsilon > 0$ and a distance d:

Definition 3 (Harel & Mannor (2012)). Let P and Q two marginal distributions over X, let $M(P,Q)$ be the set of all joint distributions over $X \times X$ with P and Q. The perturbed variation w.r.t. a distance $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and $\varepsilon > 0$ is defined by

$$PV(P,Q) = \inf_{\mu \in M(P,Q)} Proba_{\mu}[d(P',Q') > \varepsilon]$$

over all pairs $(P',Q') \sim \mu$ s.t. the marginal of P' (resp. Q') is P (resp. Q).

[2] The $\mathcal{H}$-divergence is defined with respect to the hypothesis class $\mathcal{H}$ by: $\sup_{h,h' \in \mathcal{H}} |\mathbb{E}_{x \sim \mathcal{T}}[h(x) \neq h'(x)] - \mathbb{E}_{x' \sim \mathcal{S}}[h(x') \neq h'(x')]|$, it can be empirically estimated by learning a classifier able to discriminate source and target instances Ben-David et al. (2010).

Table 1. Error rates (in%) on MOONS, the Average column reports the means and standard deviations.

Angle	20°	30°	40°	50°	60°	70°	80°	90°	Average
SVM	10.3	24	32.2	40	43.3	55.2	67.7	80.7	44.2 ± 0.9
AdaBoost	20.9	32.1	44.3	53.7	61.2	69.7	77.9	83.4	55.4 ± 0.4
DASVM	**0.0**	21.6	28.4	33.4	38.4	74.7	78.9	81.9	44.6 ± 3.2
SVM-W	6.8	12.9	9.5	26.9	48.2	59.7	66.6	67.8	37.3 ± 5.3
SLDAB-$\mathcal{H}$	6.9	11.3	18.1	32.8	37.5	45.1	55.2	59.7	33.3 ± 2.1
SLDAB-g_n	1.2	**3.6**	**7.9**	**10.8**	**17.2**	**39.7**	**47.1**	**45.5**	**21.6 ± 1.2**

A source example is thus matched with a target one if their distance is lower than ε, with respect to distance d, as illustrated in Figure 6(a). Intuitively two samples are similar if every target instance is close to a source one w.r.t. d. This measure is consistent and the empirical estimate $\hat{PV}(S,T)$ from two samples $S \sim P$ and $T \sim Q$ can be efficiently computed by a maximum graph matching procedure summarized in Algorithm 3. In our context, we apply this empirical measure on the classifier outputs: $S_{h_n} = \{h_n(x'_1), \ldots, h_n(x'_{|S|})\}$, $T_{h_n} = \{h_n(x_1), \ldots, h_n(x_{|T|})\}$ with the L_1 distance as d and use $1 - \hat{PV}(S_{h_n}, T_{h_n})$ as similarity measure, as illustrated by Figure 6(b).

For the second point, we take the following entropy-based measure:

$$ENT(h_n) = 4 \times p_n \times (1 - p_n)$$

where p_n[3] is the proportion of target instances classified as positive by h_n: $p_n = \frac{\sum_{i=1}^{|T|} [h_n(x_i) \geq 0]}{|T|}$. For the degenerate cases where all the target instances have the same class, the value of $ENT(h_n)$ is 0, otherwise, if the labels are equally distributed, this measure is close to 1.

Finally, g_n is defined by 1 minus the product of the two previous similarity measures allowing us to have a divergence of 1 if one of the similarities is null.

$$g_n = 1 - (1 - \hat{PV}(S_{h_n}, T_{h_n})) \times ENT(h_n).$$

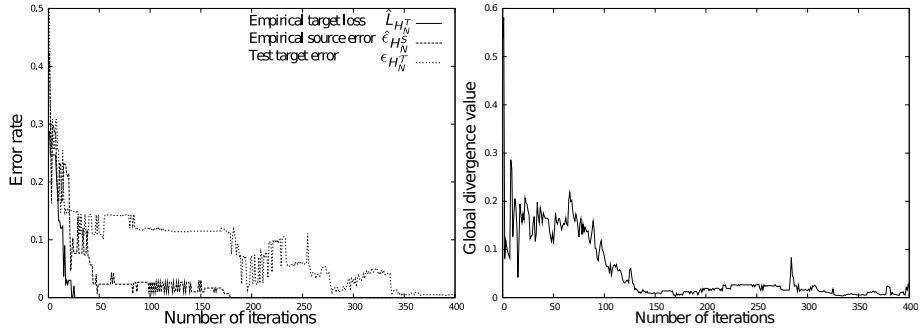
8. Experiments

To assess the practical efficiency of SLDAB and support our claim of Section 4, we perform two kinds of experiments, respectively in the DA and semi-supervised settings. We use two different databases. The first one, MOONS Bruzzone & Marconcini (2010), corresponds to two inter-twinning moons in a 2-dimensional space where the data follow a uniform distribution in each moon representing one class (see Figure 7(a) for the source domain and Figure 7(b) for the target domain, after a 30° rotation). The second one is the UCI SPAM database[4], con-

[3] True labels are assumed well balanced, if not p_n has to be reweighted accordingly.
[4] `http://archive.ics.uci.edu/ml/datasets/Spambase`

Table 2. Error rates on SPAMS.

Algorithm	Error rate (in%)
SVM	38
AdaBoost	59.4
DASVM	37.5
SVM-W	37.9
SLDAB-$\mathcal{H}$	37.1
SLDAB-g_n	**35.8**



(a) (b)

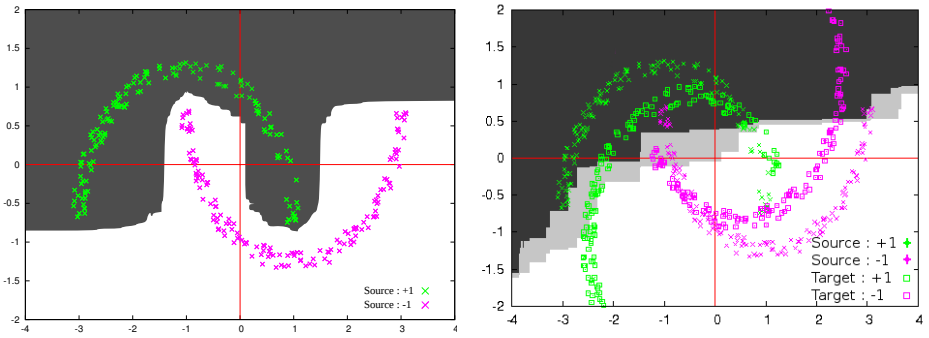
Fig. 8. (a): Loss functions on a 20° task. (b): evolution of the global divergence.

taining 4601 e-mails (2788 considered as "non-spams" and 1813 as "spams") in a 57-dimensional space.

8.1. Domain Adaptation

8.1.1. Moons Database

In this series of experiments, the target domain is obtained by rotating anti-clockwise the source domain, corresponding to the original data. We consider 8 increasingly difficult problems according to 8 rotation angles from 20 degrees to 90 degrees. For each domain, we generate 300 instances (150 of each class). To estimate the generalization error, we make use of an independent test set of 1000 points drawn from the target domain. Each adaptation problem is repeated 10 times and we report the average results obtained on the test sample without the best and the worst draws.
We compare our approach with two non DA baselines: the standard AdaBoost, using decision stumps, and a SVM classifier (with a Gaussian kernel) learned only from the source. We also compare SLDAB with DASVM (based on a LibSVM implementation) and with a reweighting approach for the co-variate shift problem presented in Huang et al. (2006). This unsupervised method (referred to as



(a) (b)

Fig. 9. Illustration of the behaviour of SLDAB in a 30° rotation task. (a): decision boundary for H_S^N on source data. (b): decision boundary for H_T^N on target data.

SVM-W) reweights the source examples by matching source and target distributions by a kernel mean matching process, then a SVM classifier is inferred from the reweighted source sample. Note that all the hyperparameters are tuned by a 10-fold cross-validation. Finally, to confirm the relevance of our divergence measure g_n, we run SLDAB with two different divergences: SLDAB-g_n uses our novel measure g_n introduced in the previous section and SLDAB-$\mathcal{H}$ is based on the $\mathcal{H}$-divergence. We tune the parameters of SLDAB by selecting, through an exhaustive grid search in the range $[0, 1]$ for both parameters λ and γ, those able to fulfill Definition 1 and leading to the smallest divergence over the final combination F_N^T. As expected, the optimal λ grows with the difficulty of the problem.

Results obtained on the different adaptation problems are reported in Table 1. We can see that, except for 20 degrees (for which DASVM is - not significantly - slightly better), SLDAB-g_n achieves a significantly better performance (using a Student paired t-test with $\alpha = 1\%$), especially on important rotation angles. DASVM that is not able to work with large distribution shifts diverges completely. This behaviour shows that our approach is more robust to difficult DA problems. Finally, despite good results compared to other algorithms, SLDAB-$\mathcal{H}$ does not perform as well as the version using our divergence g_n, showing that g_n is indeed much more adapted.

Figure 8(a) illustrates the behaviour of our algorithm on a 20 degrees rotation problem. First, as expected by Theorem 5, the empirical target loss converges very quickly towards 0. Because of the constraints imposed on the target data, the source error $\hat{\epsilon}_{H_N^S}$ requires more iterations to converge than a classical AdaBoost procedure. Moreover, the target error $\epsilon_{H_N^T}$ decreases with N and keeps dropping even when the two empirical losses have converged to zero. This confirms the benefit of having a low source error with large target margins.

Figure 8(b) shows the evolution throughout the iterations of the divergence g_n of the combination $H_n^T = \sum_{k=1}^{n} \beta_k h_k(x)$. We can see that our boosting scheme

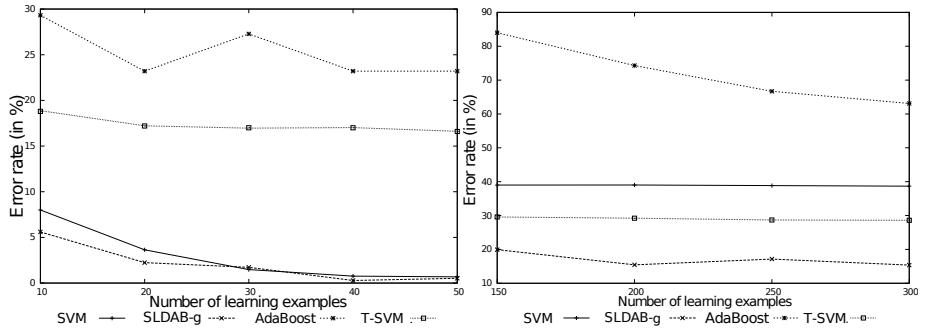


Fig. 10. (a): error rate of different algorithms on the moons semi-supervised problem according to the size of the training set. (b): error rate of different algorithms on the spam recognition semi-supervised problem according to the size of the training set.

allows us to reduce the divergence between the source and the target data, thus explaining the decrease of the target generalization error observed on the figure.

Finally, Figures 9(a) and 9(b) represent decision areas of inferred models on a 30° rotation task. Examples are labeled negative in dark region and positive in bright one. Observing Figure 9 allows us to see that the decision boundary learned on source domain (*i.e.* $H_S^N = sign(\sum_{n=1}^{N} \alpha_n sign(h_n(\cdot)))$) correctly classifies all the examples from the learning set. On Figure 9(b) is reported the decision boundary learned by SLDAB on target domain (*i.e.* $H_T^N = sign(\sum_{n=1}^{N} \beta_n sign(h_n(\cdot)))$). We can see that the rotation has been almost perfectly learned. Let us recall that this boundary decision has been inferred **without any information about target labels**. These two decision boundaries show the benefit of two different weighting schemes.

8.1.2. Spams Database

To design a DA problem from this UCI database, we first split the original data in three different sets of equivalent size. We use the first one as the learning set, representing the source distribution. In the two other samples, we add a gaussian noise to simulate a different distribution. As all the features are normalized in the [0,1] interval, we use, for each feature n, a random real value in [-0.15,0.15] as expected value μ_n and a random real value in [0,0.5] as standard deviation σ_n. We then generate noise according to a normal distribution $\mathcal{N}(\mu_n, \sigma_n)$. After having modified these two samples jointly with the same procedure, we keep one as the target learning set, the other as the test set.

This operation is repeated 5 times. The average results of the different algorithms are reported in Table 2. As for the moons problem, we compare our approach with the standard AdaBoost and a SVM classifier learned only from the source. We also compare it with DASVM and SVM-W. We see that SLDAB is able to obtain better results than all the other algorithms on this real database.

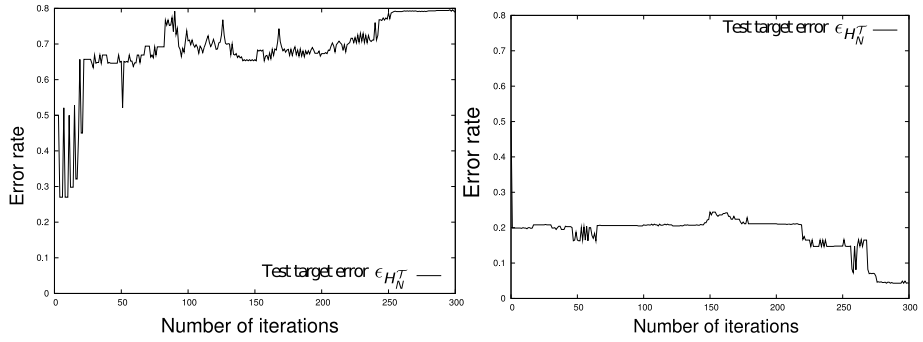


Fig. 11. Error rates obtained by the algorithm on a 50° rotation task: (a) evolution of the generalization target error without taking into account the divergence measure g_n (*i.e.* $\lambda = 0$). (b) evolution of the generalization target error, on the same experiment, using the divergence measure g_n with a non-negative value of λ.

However, it is worth noting that using a Student paired-t test, we get a p-value equal to 16%. Therefore, even though SLDAB is better, the risk of Type I for this second dataset is higher than for the Moons database. On the other hand, note that SLDAB used with our divergence g_n leads again to the best result.

8.2. Semi-Supervised Setting

Our divergence criterion allows us to quantify the distance between the two domains. If its value is low all along the process, this means that we are facing a problem that looks more like a semi-supervised task. In a semi-supervised setting, the learner receives few labeled and many unlabeled data generated from the same distribution. In this series of experiments, we study our algorithm on two semi-supervised variants of the Moons and Spams databases.

8.2.1. Moons Database

We generate randomly a learning set of 300 examples and an independent test set of 1000 examples from the same distribution. We then draw n labeled examples from the learning set, from $n = 10$ to 50 such that exactly half of the examples are positives, and keep the remaining data for the unlabeled sample. The methods are evaluated by computing the error rate on the test set. For this experiment, we compare SLDAB-g_n with AdaBoost, SVM and the transductive SVM T-SVM introduced in Joachims (1999) which is a semi-supervised method using the information given by unlabeled data to train a SVM classifier. We repeat each experiment 5 times and show the average results in Figure 10(a).

Our algorithm performs better than the other methods on small training sets and is competitive to SVM for larger sizes. We can also note that AdaBoost using only the source examples is not able to perform well. This can be explained by an overfitting phenomenon on the small labeled sample leading to poor generalization performances. Surprisingly, T-SVM performs quite poorly too. This is

probably due to the fact that the unlabeled data are incorrectly exploited, with respect to the small labeled sample, producing wrong hypotheses.

8.2.2. Spams Database

We use here the same set up as in the semi-supervised setting for MOONS. We take the 4601 original instances issued from the same distribution and split them into two sets: one third for the training sample and the remaining for the test set used to compute the error rate. From the training set, n labeled instances are drawn as labeled data, n varying from 150 to 300, the remaining part is used as unlabeled data as in the previous experiment. This procedure is repeated 5 times for each n and the average results are provided in Figure 10(b).

All the approaches are able to decrease their error rate according to the size of the labeled data (even if it is not significant for SVM and T-SVM), which is an expected behaviour. SVM and even more ADABOOST (that do not use unlabeled data), achieve a large error rate after 300 learning examples. T-SVM is able to take advantage of the unlabeled examples, with a significant gain compared to SVM. Finally, SLDAB outperforms the other algorithms by at least 10 percentage points. This confirms that SLDAB is also able to perform well in a semi-supervised learning setting. This feature makes our approach very general and relevant for a large class of problems.

8.3. On the usefulness of the divergence measure

Finally, in order to analyse the contribution of our divergence measure g_n, we run SLDAB in two settings: (1) where $\lambda = 0$, that is, the divergence g_n is not taken into account; (2) where $\lambda > 0$, that is, one penalizes hypotheses that generate a large divergence between the source and target domains. As illustrated, by Figure 11(a) on a 50° rotation problem using the MOONS database, as soon as the two domains in the DA problem are not close enough, the absence of the divergence g_n in SLDAB leads to degenerate hypothese. Figure 11(b) illustrates the behaviour of the algorithm on the exact same task, while using a non-negative value as λ. Even if the process is longer, because of the non-selection of some hypotheses which do not fulfill the conditions induced by the divergence, we can see that there is a huge gain in using g_n, confirming the legitimacy of this novel measure and its usefulness in our approach.

9. Discussion about generalization guarantees

The theoretical study introduced in Section 6 allowed us to derive several results about SLDAB. It is worth noting that we did not derive any generalization result, even though the experimental section has shown that the true risk actually decreases with the number of iterations of SLDAB. In this section, we explain why proving such generalization guarantees is complex in this DA setting.

In boosting theory Schapire et al. (1997), let us recall that a generalization error bound has been introduced, whose main advantage is not to depend on the number of iterations of the process in the penalization term.

Theorem 6 (Schapire et al. (1997)). Let $\mathcal{H}$ be a class of classifiers with VC dimension d_h. $\forall \delta > 0$ and $\gamma > 0$, with probability $1 - \delta$, any ensemble of N classifiers built from a learning set S of size $|S|$ drawn from a distribution $\mathcal{S}$ satisfies on the generalization error $\epsilon_{H_N^S}$:

$$\epsilon_{H_N^S} \leq \widehat{Pr}_{x\sim S}[margin(x) \leq \gamma] + \mathcal{O}\left(\sqrt{\frac{d_h}{|S|}\frac{log^2(|S|/d_h)}{\gamma^2} + log(1/\delta)}\right). \quad (17)$$

This well-known theorem states that achieving a large margin on the training set (the first term of the right-hand side) results in an improved bound on the generalization error, considering γ, δ, $|S|$ and d_h fixed. Moreover, Schapire et al. Schapire et al. (1997) proved that with AdaBoost this term decreases exponentially fast with the number N of classifiers. Applying Theorem 6 on the target error in our context, we get:

$$\epsilon_{H_N^T} \leq \widehat{Pr}_{x\sim T}[yF_T^N(x) \leq \gamma] + \mathcal{O}\left(\sqrt{\frac{d_h}{|S|}\frac{log^2(|S|/d_h)}{\gamma^2} + log(1/\delta)}\right). \quad (18)$$

Unlike $\widehat{Pr}_{x\sim S}(margin(x) \leq \gamma)$ in Theorem 6, we are not able to compute the true value of $\widehat{Pr}_{x\sim T}[yF_T^N(x) \leq \gamma]$: indeed, during our adaptation process we make use of the pseudo-labels to compute this loss but the true magin of an example would need the true label y. However, it is possible to go around this problem, by noting that we can introduce our target loss based on pseudo-labels thanks to the triangle inequality:

$$\begin{aligned}\widehat{Pr}_{x\sim T}[yF_T^N(x) \leq \gamma] &\leq \widehat{Pr}_{x\sim T}[yF_T^N(x) \leq \mathbf{yF_N^T}(x)] + \widehat{Pr}_{x\sim T}[\mathbf{yF_N^T}(x) \leq \gamma] \\ &\leq \widehat{Pr}_{x\sim T}[yF_T^N(x) \leq \mathbf{yF_N^T}(x)] + \widehat{Pr}_{x\sim T}[\mathbf{yF_N^T}(x) \leq \gamma] + \\ &\quad \widehat{Pr}_{(x,y)\sim S}[yF_N^S(x) \leq \gamma] - \widehat{Pr}_{(x,y)\sim S}[yF_N^S(x) \leq \gamma]\end{aligned}$$

where $\mathbf{y} = (y^1, \ldots, y^n, \ldots, y^N)$ is the vector of pseudo-classes and $\mathbf{F_N^T}(x) = (\beta_1 f_{DA}(h_1(x)), \ldots, \beta_N f_{DA}(h_N(x)))$. The term $\hat{L}_{\mathbf{H_T^N}} = \widehat{Pr}_{x\sim T}[\mathbf{yF_N^T}(x) \leq \gamma]$ corresponds to the empirical loss that we optimize with respect to the pseudo-labeled target examples and $\widehat{Pr}_{(x,y)\sim S}[yF_N^S(x) \leq \gamma]$ is the empirical loss optimized on the source instances. The quantity $\widehat{Pr}_{x\sim T}[yF_T^N(x) \leq \mathbf{yF_N^T}(x)]$ can be seen as a term assessing the quality of pseudo-labels found with respect to the true target labels and thus as a measure indicating when an adaptation is possible. Then, following some recent results such as in Ben-David et al. (2010), the objective would be to bound the target error by the source error, the margin violations over the pseudo-labeled target examples and our divergence measure computed between the two domains $\mathcal{S}$ and $\mathcal{T}$. This strategy would lead to a bound of the following form:

$$\epsilon_{\mathcal{D}_T}(H_T^N) \leq \hat{L}_{H_S^N} + \hat{L}_{\mathbf{H_T^N}} + div(S,T) + \lambda^\star + \mathcal{O}\left(\sqrt{\frac{d}{|T|}\frac{log^2(|T|/d)}{\gamma^2} + log(1/\delta)}\right).$$

As expressed in many works in DA, reducing the generalization target error is equivalent to reducing the empirical error, while decreasing the divergence between the two domains. We know that the minimization of the empirical source

error and the empirical loss over pseudo-labeled target instances is ensured by our algorithm, but we are only able to observe the empirical decrease of the global divergence between the two distributions, without proving it. In our case, the difficult point is to be able to relate some terms involving the (pseudo) margin violations proportion to our divergence between $\mathcal{S}$ and $\mathcal{T}$.

10. Conclusion

Unsupervised domain adaptation is a very challenging problem where the learner has to fit a model without having access to labeled target data. In this setting, we have introduced a new boosting algorithm, namely SLDAB, which projects both source and target data in a new N-dimensional space, where two source and target hyperplanes are optimized to minimize the source training error and the proportion of target margin violations respectively. We derive several theoretical results showing that both loss functions decrease exponentially fast with the number of iterations of boosting. Even though we could not derive formal results respecting to the generalization target error, we have experimentally shown that our strategy actually reduces the true risk. Moreover, we have shown that projecting both source and target data in this common space leads to a reduction of the divergence between the two domains. This way, SLDAB satisfies the two constraints imposed by the theoretical domain adaptation frameworks: (1) reduce the source error and (2) decrease the discrepancy between the two distributions. Another contribution of this paper takes the form of a new divergence measure, easy to compute and that prevents SLDAB from building degenerate hypotheses. Our experiments have shown that SLDAB performs well in a DA setting both on synthetic and real data. Moreover, it is also general enough to work well in a semi-supervised case, making our approach widely applicable. Despite several original contributions, this work opens the door for further investigation. From a theoretical standpoint, proving that the margin of the training examples increases or that the divergence between the two domains actually decreases would allow us to derive generalization guarantees on the true target risk. As pointed out in this paper, this task is complex because we do not have labeled target examples. From an algorithmic point of view, the generation of weak DA hypotheses also deserves a special attention. Even though solving Problem (2) tends to satisfy the weak DA constraints, the procedure can take time and may be improved by, e.g., inducing oblique induction trees in a DA way.

References

Balcan, M.-F. & Blum, A. (2006), On a Theory of Learning with Similarity Functions, *in* 'Proceedings of ICML'06', pp. 73–80.

Balcan, M.-F., Blum, A. & Srebro, N. (2008), Improved guarantees for learning via similarity functions, *in* 'Proceedings of COLT'08', pp. 287–298.

Bartlett, P. L. & Mendelson, S. (2002), 'Rademacher and gaussian complexities: Risk bounds and structural results', *Journal of Machine Learning Research* **3**, 463–482.

Becker, C., Christoudias, C. & Fua, P. (2013), Non-linear domain adaptation with boosting, *in* 'Proceedings of NIPS'13', pp. 485–493.

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F. & Vaughan, J. (2010), 'A theory of learning from different domains', *Mach. Learn.* **79**(1-2), 151–175.

Bennett, K., Demiriz, A. & Maclin, R. (2002), Exploiting unlabeled data in ensemble methods, *in* 'Proceedings of KDD'02', pp. 289–296.

Bickel, S., Brückner, M. & Scheffer, T. (2007), Discriminative learning for differing training and test distributions, *in* 'Proceedings of ICML'07', ACM, New York, NY, USA, pp. 81–88.

Blitzer, J., Dredze, M. & Pereira, F. (2007), Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification, *in* 'Proceedings of ACL'07'.

Blitzer, J., McDonald, R. & Pereira, F. (2006), Domain adaptation with structural correspondence learning, *in* 'Proceedings of EMNLP'06', pp. 120–128.

Blum, A. & Mitchell, T. (1998), Combining labeled and unlabeled data with co-training, *in* 'Proceedings of the eleventh annual conference on Computational learning theory', Proceedings of COLT'98, ACM, pp. 92–100.

Bruzzone, L. & Marconcini, M. (2010), 'Domain adaptation problems: A dasvm classification technique and a circular validation strategy', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **32**, 770–787.

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

Dai, W., Yang, Q., Xue, G. & Yu, Y. (2007), Boosting for transfer learning, *in* 'Proceedings of ICML'07', pp. 193–200.

Daumé III, H. (2007), Frustratingly easy domain adaptation, *in* 'Proceedings of ACL'07', pp. 256–263.

Dudík, M., Schapire, R. E. & Phillips, S. J. (2005), Correcting sample selection bias in maximum entropy density estimation, *in* 'Proceedings of NIPS'05'.

Florian, R., Hassan, H., Ittycheriah, A., Jing, H., Kambhatla, N., Luo, X., Nicolov, N. & Roukos, S. (2004), A statistical model for multilingual entity detection and tracking, *in* 'Proceedings of HLT-NAACL'04', pp. 1–8.

Freund, Y. & Schapire, R. (1996), Experiments with a new boosting algorithm, *in* 'Proceedings of ICML'96', pp. 148–156.

Habrard, A., Peyrache, J.-P. & Sebban, M. (2013), 'Iterative self-labeling domain adaptation for linear structured image classification', *International Journal on Artificial Intelligence Tools* **22**(5).

Harel, M. & Mannor, S. (2012), The perturbed variation, *in* 'Proceedings of NIPS'12', pp. 1943–1951.

Huang, J., Smola, A., Gretton, A., Borgwardt, K. & Schölkopf, B. (2006), Correcting sample selection bias by unlabeled data, *in* 'Proceedings of NIPS'06', pp. 601–608.

Ji, Y., Chen, J., Niu, G., Shang, L. & Dai, X. (2011), 'Transfer learning via multi-view principal component analysis', *J. Comput. Sci. Technol.* **26**(1), 81–98.

Jiang, J. (2008), 'A Literature Survey on Domain Adaptation of Statistical Classifiers'.

Joachims, T. (1999), Transductive inference for text classification using support vector machines, *in* 'Proceedings of ICML'1999', ICML '99, pp. 200–209.

Kolmogorov, A. & Tikhomirov, V. (1961), 'ϵ-entropy and ϵ-capacity of sets in

functional spaces', *American Mathematical Society Translations* **2**(17), 277–364.

Koltchinskii, V. (2001), 'Rademacher penalties and structural risk minimization', *IEEE Transactions on Information Theory* **47**(5), 1902–1914.

Leggetter, C. & Woodland, P. (1995), 'Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models', *Computer Speech & Language* pp. 171–185.

Mallapragada, P., Jin, R., Jain, A. & Liu, Y. (2009), 'Semiboost: Boosting for semi-supervised learning', *IEEE T. PAMI* **31**(11), 2000–2014.

Mansour, Y., Mohri, M. & Rostamizadeh, A. (2008), Domain adaptation with multiple sources, *in* 'Proceedings of NIPS'08', pp. 1041–1048.

Mansour, Y., Mohri, M. & Rostamizadeh, A. (2009), Domain adaptation: Learning bounds and algorithms, *in* 'Proceedings of COLT'09'.

Mansour, Y. & Schain, M. (2012), Robust domain adaptation, *in* 'Proceedings of ISAIM'12'.

Margolis, A. (2011), 'A literature review of domain adaptation with unlabeled data', *Tec. Report* pp. 1–42.

Martínez, A. (2002), 'Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class', *IEEE T. PAMI* **24**(6), 748–763.

Morvant, E., Habrard, A. & Ayache, S. (2011), Sparse Domain Adaptation in Projection Spaces based on Good Similarity Functions, *in* 'Proceedings of ICDM'11', pp. 457–466.

Morvant, E., Habrard, A. & Ayache, S. (2012), 'Parsimonious Unsupervised and Semi-Supervised Domain Adaptation with Good Similarity Functions', *Knowledge and Information Systems (KAIS)* **33**(2), 309–349.

Pan, S. J. & Yang, Q. (2010), 'A survey on transfer learning', *IEEE Transactions on Knowledge and Data Engineering* **22**, 1345–1359.

Pérez, Ó. & Sánchez-Montañés, M. A. (2007), A new learning strategy for classification problems with different training and test distributions, *in* 'Proceedings of IWANN'07', pp. 178–185.

Quionero-Candela, J., Sugiyama, M., Schwaighofer, A. & Lawrence, N. (2009), *Dataset Shift in Machine Learning*, MIT Press.

Roark, B. & Bacchiani, M. (2003), Supervised and unsupervised pcfg adaptation to novel domains, *in* 'Proceedings of HLT-NAACL'03'.

Satpal, S. & Sarawagi, S. (2007), Domain adaptation of conditional probability models via feature subsetting, *in* 'Proceedings of PKDD'07', pp. 224–235.

Schapire, R. E. & Singer, Y. (1999), 'Improved boosting algorithms using confidence-rated predictions', *Machine Learning* **37**(3), 297–336.

Schapire, R., Freund, Y., Barlett, P. & Lee, W. (1997), Boosting the margin: A new explanation for the effectiveness of voting methods, *in* 'Proceedings of ICML'97', pp. 322–330.

Sugiyama, M., Nakajima, S., Kashima, H., von Bünau, P. & Kawanabe, M. (2008), Direct importance estimation with model selection and its application to covariate shift adaptation, *in* 'Proceedings of NIPS'07'.

Tsuboi, Y., Kashima, H., Hido, S., Bickel, S. & Sugiyama, M. (2009), 'Direct density ratio estimation for large-scale covariate shift adaptation', *Proceedings of JIP'09* **17**, 138–155.

Valiant, L. (1984), 'A theory of the learnable', *Commun. ACM* **27**(11), 1134–1142.

van der Vaart, A. & Wellner, J. (1996), *Weak Convergence and Empirical Processes*, Springer series in statistics, Springer.

Xu, H. & Mannor, S. (2010*a*), Robustness and generalization, *in* 'Proceedings of COLT'10', pp. 503–515.

Xu, H. & Mannor, S. (2010*b*), Robustness and generalization, *in* 'Proceedings of COLT'10', pp. 503–515.

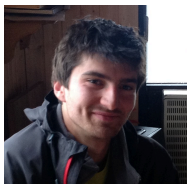
Xu, H. & Mannor, S. (2012*a*), 'Robustness and generalization', *Machine Learning* **86**(3), 391–423.

Xu, H. & Mannor, S. (2012*b*), 'Robustness and generalization', *Machine Learning* **86**(3), 391–423.

Yao, Y. & Doretto, G. (2010), Boosting for transfer learning with multiple sources, *in* 'Proceedings of CVPR'10', pp. 1855–1862.

Author Biographies

Amaury Habrard received a Ph.D. in Machine Learning in 2004 from the University of Saint- Etienne. He was Assistant Professor at the Laboratoire dInformatique Fondamentale of Aix- Marseille University until 2011, where he received a habilitation thesis in 2010. He is currently Professor in the Machine Learning group at the Hubert Curien laboratory of the University of Saint-Etienne. His research interests include metric learning, transfer learning, online learning and learning theory.



Jean-Philippe Peyrache received his Ph.D. in Machine Learning from the University of Saint-Etienne (France) in 2014. His work focused on transfer learning and domain adaptation problems using ensemble methods like boosting. Until 2014, he has been member of the machine learning team of the Hubert Curien laboratory.

Marc Sebban received a Ph.D. in Machine Learning in 1996 from the Universit of Lyon 1. After four years spent at the French West Indies and Guyana University as Assistant Professor, he got a position of Professor in 2002 at the University of Saint-Etienne (France). Since 2010, he is the head of the Computer Science, Cryptography and Imaging department of the Hubert Curien laboratory. His research interests focus on ensemble methods, metric learning, transfer learning and more generally on statistical learning theory.