Boosting for Unsupervised Domain Adaptation

Amaury Habrard, Jean-Philippe Peyrache and Marc Sebban

Université Jean Monnet de Saint-Etienne
Laboratoire Hubert Curien, UMR CNRS 5516
18 rue du Professeur Benoit Lauras - 42000 Saint-Etienne Cedex 2 - France
{amaury.habrard,jean-philippe.peyrache,marc.sebban}@univ-st-etienne.fr

Abstract. To cope with machine learning problems where the learner receives data from different source and target distributions, a new learning framework named *domain adaptation* (DA) has emerged, opening the door for designing theoretically well-founded algorithms. In this paper, we present SLDAB, a self-labeling DA algorithm, which takes its origin from both the theory of boosting and the theory of DA. SLDAB works in the difficult unsupervised DA setting where source and target training data are available, but only the former are labeled. To deal with the absence of labeled target information, SLDAB jointly minimizes the classification error over the source domain and the proportion of margin violations over the target domain. To prevent the algorithm from inducing degenerate models, we introduce a measure of divergence whose goal is to penalize hypotheses that are not able to decrease the discrepancy between the two domains. We present a theoretical analysis of our algorithm and show practical evidences of its efficiency compared to two widely used DA approaches.

1 Introduction

In many learning algorithms, it is usually required to assume that the training and test data are drawn from the same distribution. However, this assumption does not hold in many real applications challenging common learning theories such as the PAC model [20]. To cope with such situations, a new machine learning framework has been recently studied leading to the emergence of the theory of *domain adaptation* (DA) [1, 14]. 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 [13]) and very few or no labeled points from the *target* distribution. DA arises in a large spectrum of applications, such as in computer vision [16], speech processing [11, 18], natural language processing [3, 5], 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 [15, 17]). 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

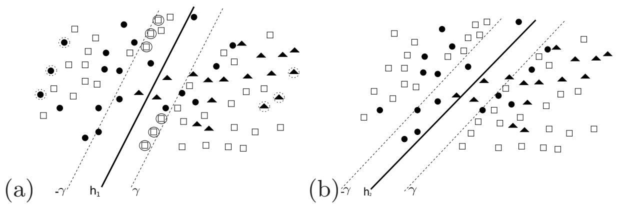


Fig. 1. Underlying principle of DASVM. (a): black examples are labeled source data (circle or triangle). Squares are unlabeled target data. A first SVM classifier h_1 is learned from the labeled source data. Then, DASVM iteratively changes some source data by semi-labeled target examples selected in a margin band (black source instances in a dashed circle and target squares in a circle). (b): new hypothesis h_2 learned using the newly semi-labeled data. h_2 works well on the source and satisfies some margin constraints on the target.

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. In this paper, we focus on a third class of approaches, called *iterative self-labeling methods.* For example, in DASVM [4], a SVM classifier is learned from the labeled source examples. Then, some of them are replaced by target data selected within a margin band (to allow slight modifications of the current classifier) but at a reasonable enough distance from the hyperplane (to have a sufficient confidence in those unlabeled examples). A new classifier is then learned using these newly *semi-labeled* target data (see Figures 1(a) and 1(b)). The process is repeated until having only semi-labeled data in the training set.

In the context of self-labeling DA, DASVM has become during the past few years a reference method. However, beyond algorithmic constraints due to the resolution of many non trivial optimization problems, it faces an important limitation: it is based on the strong assumption that, if a classifier h works well on the source data, the higher the distance from h, the higher the probability for an unlabeled sample to be correctly classified. It is worth noting that such an assumption holds only if the underlying DA problem does not require to substantially move closer the source and target distributions. As suggested by the theoretical frameworks presented in [1, 14], a DA algorithm may have not only to induce a classifier that works well on the source but also to reduce the divergence between the two distributions. This latter condition essentially enables us to have confidence in the ability of the hypothesis learned from the source to correctly classify target data. It is important to note that DASVM has not been designed for such a discrepancy reduction. In this paper, our objective is to fill the gap between the iterative self-labeling strategy and these theoretical recommendations. We present a novel DA algorithm which takes its origin from both the theory of boosting [7] 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 [6], 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 [21]. 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 [2, 12].

In this paper, we present SLDAB, 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: notations and definitions are given in Section 2; SLDAB is presented in Section 3 and theoretically analyzed in Section 4; We discuss the way to compute the divergence between the source and target domains in Section 5; Finally, we conduct two series of experiments and show practical evidences of the efficiency of SLDAB in Section 6.

2 Definitions and Notations

Let S be a set of labeled data (x', y') drawn from a source distribution $\mathcal{S}$ over $X \times \{-1, +1\}$, where X is the instance space and $\{-1, +1\}$ is the set of labels. Let T be a set of unlabeled examples x drawn from a target distribution $\mathcal{T}$ over X. Let $\mathcal{H}$ be a class of hypotheses and $h_n \in \mathcal{H} : X \rightarrow [-1, +1]$ 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 $f_{DA} : [-1, +1] \rightarrow [-1, +1]$ 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 tuning 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 $W_n^+ = \displaystyle\sum_{x \in T_n^+} D_n^T$ and $W_n^- = \displaystyle\sum_{x \in T_n^-} D_n^T$ such that $W_n^+ + W_n^- = 1$.

Let us remind that the weak assumption presented in [7] 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 Def.(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 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.

3 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

Algorithm 1 SLDAB

```
Input: a set $S$ of labeled data and $T$ 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 (1).
    Compute the divergence value $g_n$ (see Section 5 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 $H_N^T(x) = sgn(F_N^T(x))$.
```

verifies the weak DA assumption of Def.(1). 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 complicated. 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 $h_n = \sum_k \kappa_k h_n^k$ of the k stumps that satisfies simultaneously the two conditions of Def.(1). To do so, we propose to solve the following convex optimization problem:

$$\operatorname*{argmin}_{\kappa} \sum_{(x',y') \in S} D_n^S(x') \left[-y' \sum_k \kappa_k sgn(h_n^k(x')) \right]_+ + \sum_{x \in T} D_n^T(x) \left[1 - \left(\sum_k \kappa_k marg(f_{DA}(h_n^k(x))) \right) \right]_+ \quad (1)$$

where $[1-x]_+ = max(0, 1-x)$ is the hinge loss, and $marg(f_{DA}(h_n^k(x)))$ returns -1 if $f_{DA}(h_n^k(x))$ is lower than γ (i.e. h_n does not achieve a sufficient margin w.r.t. g_n) and $+1$ otherwise. Solving this optimization problem tends to fulfill Def.(1). Indeed, minimizing the first term of Eq.(1) 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.

Note that in order to generate a simple weak DA learner, we start the process with $k = 2$. If the weak DA assumption is not satisfied, we increase the dimension of the induced hypothesis h_n. Moreover, if the optimized combination does not satisfy the weak DA assumption, we draw a new set of k stumps.

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.

4 Theoretical Analysis

In this section, we present a theoretical analysis of SLDAB. Recall that the goodness of a hypothesis h_n is measured by its ability to not only 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 Def.(1) are satisfied, the standard results of AdaBoost directly hold on $\mathcal{S}$. 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.

4.1 Upper bound on the empirical loss

Theorem 1. *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*:*

$$\hat{L}_{H_N^T} = \mathbb{E}_{x \sim T}[\mathbf{y}\mathbf{F_N^T}(x) < 0] \leq \frac{1}{|T|} \sum_{x \sim T} e^{-\mathbf{y}\mathbf{F_N^T}(x)} = \prod_{n=1}^{N} Z_n, \tag{2}$$

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

Proof. The proof is the same as that of [7] except that $\mathbf{y}$ is the vector of pseudo-classes (which depend on λg_n and γ) rather than the vector of true labels. □

4.2 Optimal confidence values

Theorem 1 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:

$$Z_n = \sum_{x \in T_n^-} D_n^T(x) e^{-\beta_n f_{DA}(h_n(x)) y^n} + \sum_{x \in T_n^+} D_n^T(x) e^{-\beta_n f_{DA}(h_n(x)) y^n}. \tag{3}$$

The two terms of the right-hand side of Eq.(3) can be upper bounded as follows:

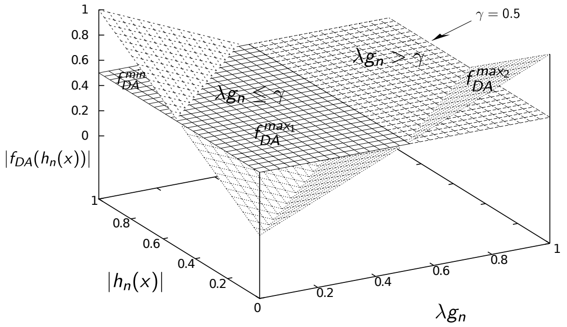


Fig. 2. Upper bounds of the components of Z_n for an arbitrary value $\gamma = 0.5$. When $x \in T_n^+$, the upper bound is obtained with $|f_{DA}| = \gamma$ (see the plateau f_{DA}^{min}). When $x \in T_n^-$, we get the upper bound with $\max(\gamma, \lambda g_n)$, that is either γ when $\lambda g_n \leq \gamma$ (see $f_{DA}^{max_1}$) or λg_n otherwise (see $f_{DA}^{max_2}$).

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

Figure 2 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 2). 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 2 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 Eq.(3), we get:

$$Z_n \leq W_n^+ e^{-\beta_n \gamma} + W_n^- e^{\beta_n \max(\gamma, \lambda g_n)} = \tilde{Z}_n. \tag{4}$$

Deriving the previous convex combination w.r.t. β_n and equating to zero, we get the optimal values for β_n in Eq.(3)[1]:

$$\begin{aligned} \frac{\partial \tilde{Z}_n}{\beta_n} = 0 &\Rightarrow \max(\gamma, \lambda g_n) W_n^- e^{\beta_n \max(\gamma, \lambda g_n)} = \gamma W_n^+ e^{-\beta_n \gamma} \\ &\Rightarrow \beta_n = \frac{1}{\gamma + \max(\gamma, \lambda g_n)} \ln \frac{\gamma W_n^+}{\max(\gamma, \lambda g_n) W_n^-}. \end{aligned} \tag{5}$$

It is important to note that β_n is computable if

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

[1] Note that the approximation $\tilde{Z}_n$ used in Eq.(4) is essentially a linear upper bound of Eq.(3) on the range $[-1; +1]$. Clearly, other upper bounds which give a tighter approximation could be used instead (see [19] for more details).

that is always true because h_n is a weak DA hypothesis and satisfies Condition 2 of Def.(1). Moreover, from Eq.(5), 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 Eq.(5) as follows:

$$\beta_n = \frac{1}{\gamma(1+c_n)} \ln \frac{W_n^+}{c_n W_n^-}, \tag{6}$$

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

4.3 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 1) to a decrease of the empirical loss $\hat{L}_{H_N^T}$ with the number of iterations.

Theorem 2. *If H_N^T is the linear combination produced by* SLDAB *from N weak DA hypotheses, then* $\lim\limits_{N\to\infty} \hat{L}_{H_N^T} = 0$.

Proof. Plugging Eq.(6) into Eq.(4) we get:

$$\begin{aligned} Z_n &\leq W_n^+ \left(\frac{c_n W_n^-}{W_n^+}\right)^{\frac{1}{(1+c_n)}} + W_n^- \left(\frac{W_n^+}{c_n W_n^-}\right)^{\frac{c_n}{(1+c_n)}} \qquad (7)\\ &= \left(W_n^+\right)^{\frac{c_n}{(1+c_n)}} \left(W_n^-\right)^{\frac{1}{(1+c_n)}} \left(c_n^{\frac{1}{(1+c_n)}} + c_n^{-\frac{c_n}{(1+c_n)}}\right) \\ &= \left(W_n^+\right)^{\frac{c_n}{(1+c_n)}} \left(W_n^-\right)^{\frac{1}{(1+c_n)}} \left(\frac{c_n+1}{c_n^{\frac{c_n}{(1+c_n)}}}\right) = u_n \times v_n \times w_n, \qquad (8) \end{aligned}$$

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

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

We deduce that

$$\frac{\partial Z_n}{\partial c_n} = (\frac{\partial u_n}{\partial c_n} \times v_n + \frac{\partial v_n}{\partial c_n} \times u_n) \times w_n + \frac{\partial w_n}{\partial c_n} \times u_n \times v_n$$

$$= \left(W_n^+\right)^{\frac{c_n}{(1+c_n)}} \times \left(W_n^-\right)^{\frac{1}{(1+c_n)}} \times \left(\frac{c_n+1}{c_n^{\frac{c_n}{(1+c_n)}}}\right) \times \frac{1}{(c_n+1)^2} \times \left(\ln W_n^+ - \ln W_n^- - \ln c_n\right)$$

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

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 $\lim\limits_{c_n \to \frac{W_n^+}{W_n^-}} Z_n = 1$.

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

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 Eq.(8) and $c_n = \tau_n \frac{W_n^+}{W_n^-}$ (that is $W_n^- = \frac{\tau_n}{\tau_n + c_n}$), we get (see Appendix 1 for more details):

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

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 2 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.

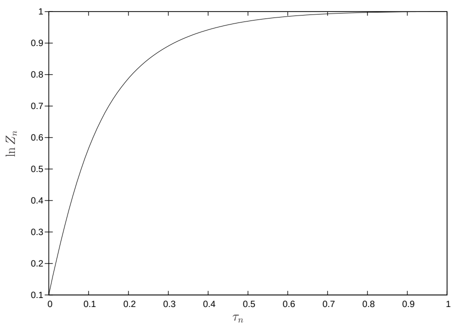


Fig. 3. Evolution of $\ln Z_n$ w.r.t. τ_n.

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 $\hat{L}_{H_N^{\mathcal{T}}}$. Figure 4.3 illustrates this claim for an arbitrarily selected configuration of W_n^+ and W_n^-. It shows that $\ln Z_n$, and thus $\hat{L}_{H_N^{\mathcal{T}}}$, decreases exponentially fast with τ_n.

5 Measure of divergence

From DA frameworks [1, 14], 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 solution is to compute a divergence with respect to the considered *class of hypotheses*, like the well-known $\mathcal{H}$-divergence[2] [1]. We claim that such a divergence 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.

For the first objective, we use the recent *Perturbed Variation* measure [8] that evaluates the discrepancy between two distributions while allowing small permitted variations assessed by a parameter $\epsilon > 0$ and a distance d:

Definition 2 ([8]). *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 $\epsilon > 0$ is defined by*

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

[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 [1].

Algorithm 2 Computation of $\hat{PV}(S,T)$ [8].

```
Input: $S = \{x'_1, \ldots, x'_{|S|}\}$, $T = \{x_1, \ldots, x_{|T|}\}$, $\epsilon > 0$ 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 $\hat{PV}(S,T) = \frac{1}{2}(\frac{S_u}{n} + \frac{T_u}{m}) \in [0,1]$
```

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

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 2. 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 $1 - \hat{PV}(S_{h_n}, T_{h_n})$.

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.

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

6 Experiments

To assess the practical efficiency of SLDAB and support our claim of Section 2, we perform two kinds of experiments, respectively in the DA and semi-supervised settings. We use two different databases. The first one, Moons [4], corresponds to two inter-twinning moons in a 2-dimensional space where the data follow a uniform distribution in each moon representing one class. The second one is the UCI Spam database[4], containing 4601 e-mails (2788 considered as "non-spams" and 1813 as "spams") in a 57-dimensional space.

6.1 Domain Adaptation

Moons database In this series of experiments, the target domain is obtained by rotating anticlockwise the source domain, corresponding to the original data.

[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

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**

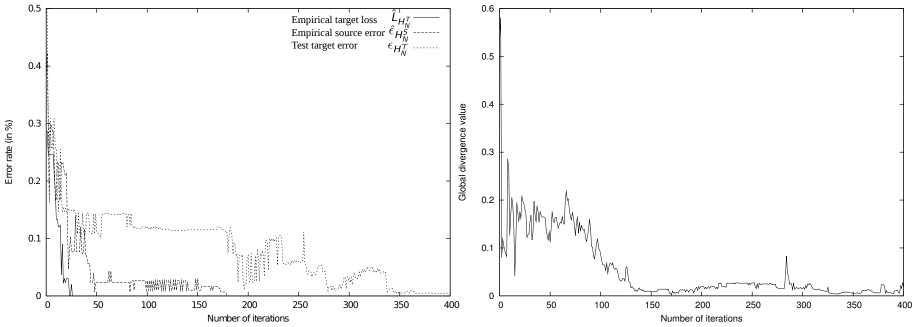
Algorithm	Error rate (in%)
SVM	38
AdaBoost	59.4
DASVM	37.5
SVM-W	37.9
SLDAB-$\mathcal{H}$	37.1
SLDAB-g_n	**35.8**

Table 1. On the left: error rates (in%) on MOONS, the Average column reports the rate averages along with average standard deviations. On the right: error rates on SPAMS.

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 and a SVM classifier (with a Gaussian kernel and hyperparameters tuned by cross-validation) 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 [9]. This unsupervised method (referred to as 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. 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, threw a grid search, those able to fulfill Def.(1) of weak DA learner 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 slightly better), SLDAB-g_n achieves a significantly better performance, 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 more adapted to our approach.

Figure 4(a) illustrates the behaviour of our algorithm on a 20 degrees rotation problem. First, as expected by Theorem 2, 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.



(a) (b)

Fig. 4. (a): loss functions on a 20° task. (b): evolution of the global divergence.

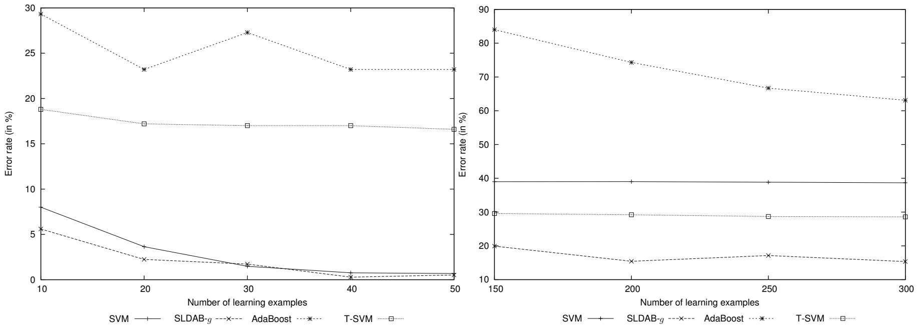
Figure 4(b) shows the evolution throughout the iterations of the divergence g_n of the combination $H_n^T = \sum_{k=1}^{n} \beta_k h_k(x)$. We can see that our boosting scheme allows us to reduce the divergence between the source and the target data.

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 1. 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. Moreover, SLDAB used with our divergence g_n leads again to the best result.

6.2 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.



(a) (b)

Fig. 5. (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.

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 [10] 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 5(a).

Our algorithm performs better than the other methods on small training sets and is competitive to SVM for larger sizes. We can also remark 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.

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 5(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.

7 Conclusion

In this paper, we have presented a new boosting-based DA algorithm called SLDAB. This algorithm, working in the difficult unsupervised DA setting, iteratively builds a combination of weak DA learners able to minimize both the source classification error and margin violations on the unlabeled target instances. The originality of this approach is to introduce the use of a new distribution divergence during the iterative process for avoiding bad adaptation due to the production of degenerate hypotheses. This divergence gives more importance to classifiers able to move closer source and target distributions with respect to the outputs of the classifiers. In this context, we have theoretically proved that our approach converges exponentially fast with the number of iterations. Our experiments have shown that SLDAB performs well in a DA setting both on synthetic and real data. Moreover, SLDAB is also general enough to work well in a semi-supervised case, making our approach widely applicable.

Even if our experiments have shown good results, we did not prove yet that the generalization error decreases. Such a result deserves further investigation but we conjecture that this is true for SLDAB. Indeed, the minimization of the margin violations on the target instances implies a minimization of our divergence in the space induced by the classifiers $\beta_n h_n$. Classical DA frameworks indicate that good generalization capabilities arise when a DA algorithm is able both to ensure a good performance on the source domain and to decrease the distribution mismatch, which is what SLDAB does. A perspective is then to show that the specific divergence we propose is able to ensure good generalization guarantees up to the ϵ used in the perturbed variation measure. Another one is to extend our approach to allow the use of a small labeled target sample.

References

1. Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., Vaughan, J.: A theory of learning from different domains. Mach. Learn. 79(1-2), 151–175 (2010)
2. Bennett, K., Demiriz, A., Maclin, R.: Exploiting unlabeled data in ensemble methods. In: KDD. pp. 289–296 (2002)
3. Blitzer, J., Dredze, M., Pereira, F.: Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In: ACL (2007)
4. Bruzzone, L., Marconcini, M.: Domain adaptation problems: a DASVM classification technique and a circular validation strategy. T. PAMI 32(5), 770–787 (2010)
5. Chelba, C., Acero, A.: Adaptation of maximum entropy capitalizer: Little data can help a lot. Computer Speech & Language 20(4), 382–399 (2006)

6. Dai, W., Yang, Q., Xue, G., Yu, Y.: Boosting for transfer learning. In: ICML. pp. 193–200 (2007)
7. Freund, Y., Schapire, R.: Experiments with a new boosting algorithm. In: ICML. pp. 148–156 (1996)
8. Harel, M., Mannor, S.: The perturbed variation. In: Proceedings of NIPS. pp. 1943–1951 (2012)
9. Huang, J., Smola, A., Gretton, A., Borgwardt, K., Schölkopf, B.: Correcting sample selection bias by unlabeled data. In: NIPS. pp. 601–608 (2006)
10. Joachims, T.: Transductive inference for text classification using support vector machines. In: Proceedings of ICML. pp. 200–209. ICML '99 (1999)
11. Leggetter, C., Woodland, P.: Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models. Computer Speech & Language pp. 171–185 (1995)
12. Mallapragada, P., Jin, R., Jain, A., Liu, Y.: Semiboost: Boosting for semi-supervised learning. IEEE T. PAMI 31(11), 2000–2014 (2009)
13. Mansour, Y., Mohri, M., Rostamizadeh, A.: Domain adaptation with multiple sources. In: NIPS. pp. 1041–1048 (2008)
14. Mansour, Y., Mohri, M., Rostamizadeh, A.: Domain adaptation: Learning bounds and algorithms. In: COLT (2009)
15. Margolis, A.: A literature review of domain adaptation with unlabeled data. Tech. rep., Univ. Washington (2011)
16. Martínez, A.: Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class. IEEE T. PAMI 24(6), 748–763 (2002)
17. Quionero-Candela, J., Sugiyama, M., Schwaighofer, A., Lawrence, N.: Dataset Shift in Machine Learning. MIT Press (2009)
18. Roark, B., Bacchiani, M.: Supervised and unsupervised pcfg adaptation to novel domains. In: HLT-NAACL (2003)
19. Schapire, R.E., Singer, Y.: Improved boosting algorithms using confidence-rated predictions. Machine Learning 37(3), 297–336 (1999)
20. Valiant, L.: A theory of the learnable. Commun. ACM 27(11), 1134–1142 (1984)
21. Yao, Y., Doretto, G.: Boosting for transfer learning with multiple sources. In: CVPR. pp. 1855–1862 (2010)

Appendix 1

If the weak DA assumption is satisfied, $c_n = \tau_n \frac{W_n^+}{W_n^-}$ with $\tau_n \in]\frac{W_n^-}{W_n^+};1[$. We deduce that $W_n^- = \frac{\tau_n}{\tau_n + c_n}$.

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