On learning a large margin classifier for domain adaptation based on similarity functions

Sofien Dhouib, Ievgen Redko, Carole Lartizien

On learning a large margin classifier for domain adaptation based on similarity functions

Sofien Dhouib[1], Ievgen Redko*[2], et Carole Lartizien†[1]

[1]Univ Lyon, INSA-Lyon, Université Claude Bernard Lyon 1, UJM-Saint Etienne, CNRS, Inserm, CREATIS UMR 5220, U1206, F-69100, LYON, France
[2]Univ Lyon, UJM-Saint-Etienne, CNRS, Institut d Optique Graduate School, Laboratoire Hubert Curien UMR 5516, F-42023, Saint-Etienne, France

May 15, 2019

Abstract

Traditional supervised classification algorithms fail when unlabeled test data arise from a probability distribution that differs from that of the labeled training data. This problem is addressed by domain adaptation, an active research area in which one would like to transfer the knowledge acquired from a first labeled domain, the source, to a second one, the target. In this paper, we tackle this problem from the perspective of large margin classifiers based on $(\epsilon, \gamma, \tau)-$good similarity functions. We first prove a bound on the error of such a classifier on the target domain. Then we present our algorithm consisting in minimizing this bound, allowing to learn a good classifier directly on the target domain without an intermediate domain alignment step. Under specific conditions, our algorithm can be formulated as a convex optimization problem that is solved efficiently. Its performance is assessed via experiments on on a toy set and a real world problem.

Keywords: binary classification, domain adaptation, large margin, similarity functions

1 Introduction

Classification algorithms are used in several real world applications such as image recognition and sentiment analysis. Some of these algorithms output decision rules are based on a pairwise similarity between different data instances, with two of the most known algorithms of this kind being the nearest neighbors classifier [CH67] and the support vector machine [BGV92] (SVM). In order to determine the class of a given example, the former relies on the labels of its nearest neighbors in the sense of a given distance via a voting rule, whereas the latter outputs a classifier that is a linear combination of the point's similarity to the rest of the training data, where the used similarity is restricted to verify Mercer's theorem, i.e to be a kernel. Despite such a restriction, SVM's are appealing due to their great generalization capacity that is empirically observed and theoretically proven [CV95]. In fact, their aim at separating the data with the largest possible margin is a major cause of their success. As a result, it seems interesting to keep this large margin separation aspect without constraining the used similarity function to be a kernel. This is the main topic of the two seminal papers of [BBS08b, BBS08a], which define and analyze the goodness of a similarity function for a given binary classification problem. The definition they introduce is rather intuitive, stating that with a high probability $(1 - \epsilon)$, the average similarity of a data point to landmarks of its own label is greater than that to the opposite label, with the difference between the two average similarities being at least equal to a margin γ. The landmarks are a priori fixed instances that represent a τ fraction of the available data. Given a similarity function verifying this intuitive condition, the authors define a new representation space in which an instance's features are its similarities to the different landmarks, and prove that with enough drawn landmarks, the two classes are linearly separable with a large margin in that representation. Given such encouraging theoretical guarantees, several works in the literature considered the problem of learning such func-

*https://ievred.github.io/
†https://www.creatis.insa-lyon.fr/ lartizien/

tions ([BHS12, GY13, NGHS15, ISH+15]).

The way they are defined, $(\epsilon, \gamma, \tau)-$good similarity functions are convenient for a supervised learning setting in which the training and testing sets arise from the same probability distribution and belong to the same input space. This might not always be the case for real world applications where new partially or totally unlabeled data is available for a given application, but follows a probability distribution that differs from the one generating the labeled training data. An efficient approach to tackle such a problem is transferring the knowledge acquired on the training data to the new test data, and this task is at the heart of the domain adaptation, a currently active research area [PY10, WKW16, Mar11], in which the labeled domain is called the source, while the unlabeled one is called the target. More precisely, domain adaptation offers methodological frameworks and algorithms allowing to leverage information available from both domains to output decision rule that is good for the target one.

In this paper, we consider the aforementioned similarity functions in the challenging setting of unsupervised domain adaptation, where no labels are available for the target domain. We start our contribution by providing a theoretical bound on the error of a given similarity function on the target domain by bounding the error of the corresponding classifier. Up to some terms that are neglected afterwards, the bound is a trade off between a source error term and a disagreement between the two domains. The latter is similar to the $H\Delta H$ divergence [BDBC+10], and its generalization, the discrepancy distance [MMR09]. Following this theoretical contribution, we derive an algorithm that minimizes this bound so that a classifier is directly learnt for the target domain without an additional domain alignment step.

As far as we know, the only works that consider the application of $(\epsilon, \gamma, \tau)-$good similarities in domain adaptation are [MHA12] and [DR18]. The first one uses a heuristic to select landmarks that move closer the two distributions in the projection space, with the similarity function they used being a kernel that is iteratively reweighted. Our work differs from theirs as in our case the similarity function is learnt in one step and using all of the source instances as landmarks. The second establishes theoretical bounds for the error of a similarity function on a target domain in terms of ℓ_1 and χ_2 divergences between probability distributions. In this work, we provide a new bound that involves a domain disagreement term taking into account the considered hypothesis spaces.

In terms of its mechanism consisting in directly learning a classifier on the target domain without an alignment step, our algorithm is similar to the one proposed in [GHLM17]. However, while our domain disagreement term is defined by a supremum, theirs is rather an expectation over the set of considered hypotheses classes, as they consider a PAC-Bayesian setting.

The rest of the paper is organized as follows: the first section introduces required preliminary knowledge and notations. Section 2 is dedicated to our contributions, where we first derive a bound on the error term of the target domain. Then, this bound is used to derive an algorithm by considering a particular case of bilinear similarity functions and linear classifiers, resulting in a a convex programming formulation that is solved efficiently. Finally, in the last section we evaluate our algorithm on a toy data set and on a real world problem.

2 Preliminaries and notations

We consider a binary classification setting, in which the feature space is $\mathcal{X} \subset \mathbb{R}^d$ and the labels set is $\mathcal{Y} = \{-1, 1\}$. Since we work in a domain adaptation context, we suppose having access to a labeled source sample S and an unlabeled target one T, drawn respectively from probability distributions $\mathcal{S}$ and $\mathcal{T}$. Furthermore, we denote by f_S and f_T the two functions labeling the instances of both distributions.

We now recall the definition of a good similarity function $K : \mathcal{X} \times \mathcal{X} \to [-1, 1]$ introduced in [BBS08a]

Definition 1 (Balcan et. al. 2008)**.** *A similarity function K is (ϵ, γ, τ)-good in hinge loss for problem (distribution) $\mathcal{P}$ if there exists a (probabilistic) indicator function R of a set of "reasonable points" such that:*

$$\mathop{\mathbb{E}}_{(x,y)\sim\mathcal{P}} \left[\left(1 - \frac{y.k(x)}{\gamma} \right)_+ \right] \leq \epsilon, \tag{1}$$

$$\mathop{\mathbb{P}}_{x'\sim\mathcal{P}} [R(x')] \geq \tau, \tag{2}$$

where $k(x) = \mathop{\mathbb{E}}_{(x',y')\sim\mathcal{P}} [y' K(x, x') | R(x')]$.

This definition formalizes the intuition that most of instances drawn from a probability distribution $\mathcal{P}$ should have a greater average similarity to landmarks of their own class, than those of the opposite class by a margin γ at least. In fact, $k(x)$ represents the difference between the instance x's average similarity to its own class and to the opposite class, and Equation (1) reflects a penalization for the case where $yk(x) < \gamma$, i.e where the signed difference is not large enough.

Equivalently, k can be seen as a hypothesis used to classify instances, and Equation (1) says that this hypothesis has an expected hinge loss bounded by ϵ at margin γ. We will consider this classifier throughout the next section detailing our contribution.

Given such a similarity function, the authors of [BBS08b, BBS08a] prove that with enough landmark instances, one can construct a new representation space where the features of an instance is its similarities to those landmarks, and where the two classes are linearly separable with a large margin. This result is reminiscent of the kernel trick for SVM's, but it is more general as K does not necessarily have to be a kernel. We note that Definition 1 was modified in [DR18] in a way allowing landmarks to come from a probability distribution that is not necessarily that of the tested data instances.

Using that modified definition, we introduce the following quantity for a probability distribution $\mathcal{P}$ that will be either $\mathcal{S}$ or $\mathcal{T}$ in the rest of the paper.

$$\mathcal{E}_{\mathcal{P}}(k,h) := \mathop{\mathbb{E}}_{x\sim\mathcal{P}}[(1-h(x)k(x)/\gamma_p)_+] \tag{3}$$

where

$$k(x) := \mathop{\mathbb{E}}_{(x',y')\sim\mathcal{S}}[K(x,x')f(x')] \tag{4}$$

γ_p is a margin associated to the probability distribution $\mathcal{P}$ and h is a classifier. We note that regardless of distribution $\mathcal{P}$, landmarks in $\mathcal{E}_{\mathcal{P}}(k,h)$ are drawn from distribution $\mathcal{S}$. In the case where $\mathcal{P}=\mathcal{S}$, this corresponds to Definition 1 with $\tau=1$ and in the case of $\mathcal{P}=\mathcal{T}$, it corresponds to its modification in [DR18].

3 Learning a good classifier for the target domain

We hereby present our contribution, starting by establishing a theoretical bound on the average hinge loss of a classifier k on the target domain. This bound is further used to derive an algorithm that directly learns a good classifier for the target.

3.1 Problem setup

Our goal is to learn a classifier k, or equivalently a similarity function K that has a low error on the target domain $\mathcal{E}_{\mathcal{T}}(k,f_T)$. We follow an approach analogous to the ones presented in [BDBC$^+$10] and [MMR09]. We recall the main result stated in [MMR09, Theorem 8]:

Theorem 1 (Mansour et al., 2009)**.** *Let $\mathcal{H}$ be a hypothesis space, and $L:\mathcal{H}\times\mathcal{H}\to\mathbb{R}^+$ a symmetric loss function verifying the triangle inequality. For a probability distribution $\mathcal{P}$ and $h,g\in\mathcal{H}$, let $\mathcal{L}_{\mathcal{P}}(h,g)=\mathop{\mathbb{E}}_{x\sim\mathcal{P}}[L(g(x),h(x))]$. Let h_S^* and h_T^* as the best classifiers from $\mathcal{H}$ achieving the lowest errors on $\mathcal{S}$ and $\mathcal{T}$ respectively. Finally, define the discrepancy distance between $\mathcal{S}$ and $\mathcal{T}$ as $\mathrm{disc}_L(\mathcal{T},\mathcal{S})=\sup_{h,h'\in\mathcal{H}}|\mathcal{L}_T(h,h')-\mathcal{L}_S(h,h')|$. Then,*

$$\mathcal{L}_T(h,f_T)\le\mathcal{L}_S(h,h_S^*)+\mathrm{disc}(\mathcal{T},\mathcal{S})\\ \mathcal{L}_T(h_T^*,f_T)+\mathcal{L}_T(h_T^*,h_S^*)$$

The authors assume that $\mathcal{L}_T(h_T^*,h_S^*)$, the average loss between the best in-class hypotheses for each domain, is small for adaptation to be possible. Furthermore, the term $\mathcal{L}_T(h_T^*,f_T)$ is assumed to be small given that the hypothesis space has enough richness to represent f_T with low error, and $\mathcal{L}_S(h,h_S^*)$ is close to the considered classifier h's error on the source again if f_S is well approximated by h_S^*. This bound will hence be small if h performs well on the source domain and if $\mathcal{S}$ and $\mathcal{T}$ are close in terms of the discrepancy distance.

The above result cannot be used directly as we consider the hinge loss that does not verify the triangular inequality. In order to prove our result, we assume that there exists a function $f:\mathcal{X}\to[-1,1]$ that performs well on the the source and target domains. In the case f achieves a perfect labeling of both the source and target domains, this assumption corresponds in the domain adaptation literature to the covariate shift [SKM07]. Otherwise, it is similar to the ideal joint hypothesis defined in [BDBC$^+$10] and considered in [GHLM17]. While such a function is unknown, we suppose that it belongs to a hypothesis space $\mathcal{H}$ verifying $h\in\mathcal{H}\Rightarrow -h\in\mathcal{H}$.

With these assumptions, we are ready to state our bound on the expected error on the target domain $\mathcal{E}_{\mathcal{T}}(k,f_T)$.

Proposition 1. *Given a similarity function K with a corresponding classifier k, two margins γ_s,γ_t respectively associated to the source and target domains and their ratio $\delta=\frac{\gamma_s}{\gamma_t}$, the following holds:*

$$\mathcal{E}_{\mathcal{T}}(k,f_T)\le\mathcal{E}_{\mathcal{S}}(k,f_S)+\frac{1}{\gamma_s}\sup_{h\in\mathcal{H}}\Delta_\delta(k,h)\\ +\mathop{\mathbb{E}}_{x\sim\mathcal{S}}\left[\frac{|f_S(x)-f(x)|}{\gamma_s}\right]+\mathop{\mathbb{E}}_{x\sim\mathcal{T}}\left[\frac{|f_T(x)-f(x)|}{\gamma_t}\right]$$

where $\Delta_\delta(k,h)=\mathop{\mathbb{E}}_{x\sim\mathcal{T}}\left[\left|\mathop{\mathbb{E}}_{x'\sim\mathcal{S}}[k(x')h(x')]-\delta k(x)h(x)\right|\right]$

Proof. We start by writing:

$$\begin{aligned}\mathcal{E}_{\mathcal{T}}(k, f_T) &= \mathcal{E}_{\mathcal{T}}(k, f_T) - \mathcal{E}_{\mathcal{T}}(k, f) && (5)\\ &+ \mathcal{E}_{\mathcal{T}}(k, f) - \mathcal{E}_{\mathcal{S}}(k, f) && (6)\\ &+ \mathcal{E}_{\mathcal{S}}(k, f) - \mathcal{E}_{\mathcal{S}}(k, f_S) && (7)\\ &+ \mathcal{E}_{\mathcal{S}}(k, f_S).\end{aligned}$$

For (7), we have:

$$\begin{aligned}&\mathcal{E}_{\mathcal{S}}(k, f) - \mathcal{E}_{\mathcal{S}}(k, f_S)\\ =&\mathop{\mathbb{E}}_{x\sim\mathcal{S}}\left[(1 - \frac{k(x)f(x)}{\gamma_s})\right] - \mathop{\mathbb{E}}_{x\sim\mathcal{S}}\left[(1 - \frac{k(x)f_S(x)}{\gamma_s})\right]\\ \leq&\frac{1}{\gamma_s}\mathop{\mathbb{E}}_{x\sim\mathcal{S}}\left[k(x)(f_S(x) - f(x))_+\right].\end{aligned}$$

Term (5) can be bounded in the same manner. Concerning term (6), we have:

$$\begin{aligned}&\mathcal{E}_{\mathcal{T}}(k, f) - \mathcal{E}_{\mathcal{S}}(k, f)\\ =&\mathop{\mathbb{E}}_{x\sim\mathcal{T}}\left[\left(1 - \frac{k(x)f(x)}{\gamma_t}\right)_+\right] - \mathop{\mathbb{E}}_{x\sim\mathcal{S}}\left[\left(1 - \frac{k(x)f(x)}{\gamma_s}\right)_+\right]\\ \leq&\mathop{\mathbb{E}}_{x\sim\mathcal{T}}\left[\left(1 - \frac{k(x)f(x)}{\gamma_t}\right)_+\right] - \left(1 - \mathop{\mathbb{E}}_{x\sim\mathcal{S}}\left[\frac{k(x)f(x)}{\gamma_s}\right]\right)_+ && (8)\\ \leq&\mathop{\mathbb{E}}_{x\sim\mathcal{T}}\left[\left(\frac{\mathop{\mathbb{E}}_{x'\sim\mathcal{S}}\left[k(x')f(x')\right]}{\gamma_s} - \frac{k(x)f(x)}{\gamma_t}\right)_+\right] && (9)\\ =&\frac{1}{\gamma_s}\mathop{\mathbb{E}}_{x\sim\mathcal{T}}\left[\left(\mathop{\mathbb{E}}_{x'\sim\mathcal{S}}\left[k(x')f(x')\right] - \delta k(x)f(x)\right)_+\right]\\ \leq&\frac{1}{\gamma_s}\sup_{h\in\mathcal{H}}\left(\mathop{\mathbb{E}}_{x\sim\mathcal{T}}\left[\left(\mathop{\mathbb{E}}_{x'\sim\mathcal{S}}\left[k(x')h(x')\right] - \delta k(x)h(x)\right)_+\right]\right)\\ =&\frac{1}{\gamma_s}\sup_{h\in\mathcal{H}}\left(\mathop{\mathbb{E}}_{x\sim\mathcal{T}}\left[\left|\mathop{\mathbb{E}}_{x'\sim\mathcal{S}}\left[k(x')h(x')\right] - \delta k(x)h(x)\right|\right]\right)\end{aligned}$$

where (8) is obtained by applying Jensen's inequality to the convex function $(1 - \cdot)_+$. Then, using the inequality $(t)_+ - (s)_+ \leq (t - s)_+$ (sub-additivity of the positive part), one gets line (9). The last line is a consequence of the fact that $h \in \mathcal{H} \Rightarrow -h \in \mathcal{H}$. □

The established bound has 4 terms: the first one is the error on the source w.r.t function f_S, known through the labels y. This term is similar to $\mathcal{L}_S(h, h^*_S)$ in theorem 1.

The second term reflects a disagreement between the source and the target w.r.t hypothesis k. Its counterpart in 1 is the discrepancy disc, and if $\mathcal{K} = \mathcal{H}$, both disagreement measures bound the same quantity:

$$\begin{aligned}\mathcal{E}_{\mathcal{T}}(k, f) - \mathcal{E}_{\mathcal{S}}(k, f) &\leq \sup_{h,h'\in\mathcal{H}} |\mathcal{E}_{\mathcal{T}}(h, h') - \mathcal{E}_{\mathcal{S}}(h, h')|\\ &= \mathrm{disc}(\mathcal{T}, \mathcal{S})\\ \mathcal{E}_{\mathcal{T}}(k, f) - \mathcal{E}_{\mathcal{S}}(k, f) &\leq \frac{1}{\gamma_s}\sup_{x\in}\Delta_\delta(k, h)\end{aligned}$$

The last two terms are distances between f and the best hypotheses in $\mathcal{H}$ that respectively label elements drawn from $\mathcal{S}$ and $\mathcal{T}$. We will neglect both of them in our algorithm's definition, as we would expect them to be small for the adaptation to be possible, given the existence of f. After omitting these two terms, the sum of the two remaining terms defines our algorithm that we detail in the next section.

3.2 Algorithm derivation

We suppose searching for the similarity function K in hypothesis space, and we denote by $\mathcal{K}$ the hypothesis space of classifiers it induces, i.e from which the resulting classifier k is picked. Our algorithm is then formulated as follows:

$$\underset{k\in\mathcal{K}}{\text{minimize}}\ \mathcal{E}_{\mathcal{S}}(k, f_S) + \frac{1}{\gamma_s}\sup_{h\in\mathcal{H}}\Delta_\delta(k, h)$$

The cost function in this case contains a supremum over the potentially infinite hypothesis set $\mathcal{H}$, which makes the optimization difficult. However, we will show in the next section that a particular choice of $\mathcal{H}$ allows to deal efficiently with this term, transforming it into a maximum over a finite set.

Using the Lagrange multipliers, the minimization problem is equivalent to:

$$\begin{aligned}&\underset{k\in\mathcal{K}}{\text{minimize}}\ \sup_{h\in\mathcal{H}}\Delta_\delta(k, h)\\ &\text{subject to } \mathcal{E}_{\mathcal{S}}(k, f_S) \leq \epsilon\end{aligned}$$

where ϵ is a hyperparameter. Without the constraint on the source domain error, the trivial null similarity function $K \equiv 0$ is a solution to the problem. Thus, it plays a major role in avoiding this solution, in addition to imposing a low error on the source domain.

We note that if the constrained version of the minimization algorithm manages to find a small value for $\sup_{h\in\mathcal{H}}\Delta_\delta(k, h)$, then from the proof of Proposition 1, we have

$$\begin{aligned}\mathop{\mathbb{E}}_{x\sim\mathcal{T}}\left[\left(\mathop{\mathbb{E}}_{x'\sim\mathcal{S}}\left[k(x')f(x')\right] - \delta k(x)f(x)\right)_+\right] &= \Delta_\delta(k, f)\\ &\leq \sup_{h\in\mathcal{H}}\Delta_\delta(k, h).\end{aligned}$$

The left hand side of these inequalities penalizes the case $\frac{1}{\delta}\mathop{\mathbb{E}}_{x'\sim\mathcal{S}}[k(x')f(x')] > k(x)f(x)$ for every instance

x drawn from the target distribution, resulting in similarity K that is good on $\mathcal{T}$ with margin $\bar{\gamma} = \frac{\gamma_t}{\gamma_s} \mathop{\mathbb{E}}_{x\sim\mathcal{S}} \left[\mathop{\mathbb{E}}_{x'\sim\mathcal{S}} [K(x,x')f(x)f_S(x')] \right]$. This is the mean margin of K on the source domain when f and f_S label respectively the data points and the landmarks, up to a scaling factor $\frac{\gamma_t}{\gamma_s}$. This also means that the classifier k would have a low error on the target with a margin $\frac{\bar{\gamma}}{\gamma_s}\gamma_t$, which is a scaled version of the originally considered margin γ_t for the target domain.

3.3 Case of bilinear similarities and linear classifiers

Below, we discuss a particular choice for both hypotheses classes $\mathcal{K}$ and $\mathcal{H}$ in order to make the optimization problem tractable. To this end, we consider bilinear similarity functions $K(x,x') = x^T A x'$ with $A \in \mathbb{R}^{d\times d}$, $\|A\| \leq 1$. Such functions were studied in the $(\epsilon,\gamma,\tau)-$good framework in [BHS12]. By scaling the data instances so that their Euclidean norms are bounded by 1, K takes values in $[-1,1]$. We proceed to determine the implied $\mathcal{K}$ space in this case. Let $x \in \mathbb{R}^d$:

$$\begin{aligned} k(x) &= \mathop{\mathbb{E}}_{x'\sim\mathcal{S}} [K(x,x')f_S(x')] \\ &= \mathop{\mathbb{E}}_{x'\sim\mathcal{S}} \left[x^T A x' f_S(x')\right] \\ &= x^T A \mu \end{aligned}$$

with $\mu = \mathop{\mathbb{E}}_{x\sim\mathcal{S}} [x.f_S(x)]$ and

$$\begin{aligned} \mathcal{K} =& \{k : x \mapsto x^T A\mu; A \in \mathbb{R}^{d\times d} \|A\| \leq 1\} \\ \simeq& \{A\mu; A \in \mathbb{R}^{d\times d}; \|A\| \leq 1\} \\ \subset& \{a \in \mathbb{R}^d; \|a\| \leq 1\}, \end{aligned}$$

where the $\simeq$ symbol denotes equality up to an isomorphism of vector spaces.

As for $\mathcal{H}$, we choose the space of linear classifiers with bounded ℓ_1 norm:

$$\mathcal{H} \simeq \{w \in \mathbb{R}^d; \|w\|_1 \leq 1\}.$$

Hence, for $k(x) = a^T x$ and $h(x) = w^T x$, one has

$$\begin{aligned} \Delta_\delta(a,w) &= \mathop{\mathbb{E}}_{x\sim\mathcal{T}} \left[\left(\mathop{\mathbb{E}}_{x'\sim\mathcal{S}} \left[a^T x' x'^T w\right] - \delta a^T x x^T w \right)_+ \right] \\ &= \mathop{\mathbb{E}}_{x\sim\mathcal{T}} \left[\left| a^T \left(\mathop{\mathbb{E}}_{x'\sim\mathcal{S}} \left[x' x'^T\right] - \delta x x^T \right) w \right| \right]. \end{aligned}$$

For a fixed $a \in \mathbb{R}^d$, the function $w \mapsto \Delta_\delta(a,w)$ is convex, hence its supremum over $\mathcal{H}$, an ℓ_1 unit ball, is reached on one of its vertices

$$\sup_{\|w\|_1\leq 1} \Delta_\delta(a,w) = \max_{1\leq i\leq d} \left(\mathop{\mathbb{E}}_{x\sim\mathcal{T}} \left[\left| a^T (\Sigma_S - \delta x x^T) e_i \right| \right] \right),$$

where $\Sigma_S = \mathop{\mathbb{E}}_{x\sim\mathcal{S}} \left[xx^T\right]$ and $\{e_1, ..., e_d\}$ is the canonical basis of $\mathbb{R}^d$. Multiplying the cost function by γ_s, the problem is written:

$$\begin{aligned} &\underset{\|a\|\leq 1}{\text{minimize}} \mathop{\mathbb{E}}_{x\sim\mathcal{S}} [(\gamma_s - f_S(x)k(x))_+] + M \\ &\text{s.t. } M \geq \mathop{\mathbb{E}}_{x\sim\mathcal{T}} \left[\left| a^T (\Sigma_S - \delta x x^T) e_i \right| \right] \forall 1 \leq i \leq d. \end{aligned}$$

In the empirical case, i.e when the expectation terms are replaced by corresponding empirical means over the source and target samples of respective sizes m and n, this is a quadratic optimization problem having $d+m$ variables (d for the size of vector a, m for the positive variables representing the positive parts) and $1+2m+2nd$ constraints (one for $\|a\|$, 2 for each source instance and $2d$ for each target one). It can be solved efficiently using standard convex optimization solvers.

4 Experiments

In this section, we evaluate our method on two domain adaptation problems. The first is defined by a toy set with a controllable difficulty, while the second is a real world problem.

4.1 Cross-validation

We choose the best values of hyperparameters γ_s and δ by a reverse validation procedure ([ZFY+10], [BM10]) following the protocol of [GHLM17] with a 5 folds validation. Given a fold $i \in 1, ..., 5$ defining a training set $S \setminus S_i$ and a validation set and S_i for the source, and similarly $T \setminus T_i$ and T_i for the target, a classifier h is learnt from labeled $S \setminus S_i$ and unlabeled $T \setminus T_i$. Then, keeping the same hyperparameters used to learn h, a reverse classifier h_r is learnt from $T \setminus T_i$ labeled by h and unlabeled $S \setminus S_i$, and finally evaluated on S_i. The chosen hyperparameters are those minimizing the average error of h_r over the folds. This way we do not make use of the target labels, which fits with the unsupervised domain adaptation setting. We slightly modify this procedure to suit our method: γ_s and δ, or equivalently γ_s and γ_t are respectively associated with $\mathcal{S}$ and $\mathcal{T}$, hence when computing the reverse classifier h_r, we replace γ and δ respectively by $\frac{\gamma_s}{\delta} = \gamma_t$ and $\frac{1}{\delta} = \frac{\gamma_t}{\gamma_s}$. Moreover, the best hyperparameters are those minimizing the average margin violation loss of h_r on the source validation sets S_i, i.e

$$\frac{1}{5} \sum_{i=1}^{5} \left(\frac{1}{|S_i|} \sum_{x\in S_i} [y(x)h_r(x) < \gamma_s] \right)$$

Angle (°)	20	30	40	50	70	90
SVM (no adaptation) [CFTR15]	10.4	24	31.2	40	76.4	82.8
DASVM [BM10]	**0**	25.9	28.4	33.4	74.7	82
PBDA [GHLM17]	9.4	10.3	22.5	41.2	62.6	68.7
OT-GL [CFTR15]	**0**	**0**	**1.3**	19.6	**37.8**	**50.8**
DASF [MHA12]	**0.20**	**0.45**	8.97	18.73	38.05	**40.25**
Our algorithm (8 KPCA features)	1.31	2.37	**3.56**	**12.9**	54.45	72.41
Our algorithm (20 KPCA features)	0.84	4.11	8.76	**13.93**	**36.19**	56.32

Table 1: Average 0-1 loss (percentage) over 10 realizations for the moons toy set.

where $[\ldots]$ denotes the Iverson bracket. After choosing the hyperparameters by this procedure, the resulting classifier's performance is evaluated on two independent source and target sets. This whole procedure (reverse validation then testing) is repeated 10 times, and the average performances over those repetitions are reported. For all of the experiments, we use the `CVXPY` modeling language ([DB16], [AVDB18]) with the solver `MOSEK` ([ApS17]).

4.2 Moons data set

We carry on our experiments on the moons data set used in [GHLM13] and [CFTR15]. The source data set is centered at the origin $(0, 0)$, and has 300 instances, which are rotated around that point by a certain angle to get the target distribution. Obviously the greater is the angle, the further from each other are the two domains and the harder is the adaptation. To ensure the data is linearly separable, we apply a Kernel PCA [SSM97] with a Gaussian kernel having a parameter σ equal to the mean Euclidean distance between instances (as done in [KJ11]), keeping respectively 8 and 20 components. The mean over 10 tests on independent data sets of 1000 instances are reported in Table 1, where our algorithm is compared to an SVM trained on the source domain (without adaptation) and domain adaptation algorithms DASVM [BM10], PBDA [GHLM17], OT-GL [CFTR15] and DASF[MHA12]. We observe that our method outperforms both DASVM and PBDA for angles up to 70°. However, its performance remains lower than DASF and OT-GL except for angles 50°and 70°. We think that this difference is due to the fact that our bound overestimates the divergence between the two domains for small angles due to the supremum term in Proposition 1.

4.3 Prostate cancer data set

We test our algorithm on a real world problem: a clinical data set of multi-parametric magnetic resonance images (mp-MRI) collected to train a computer-aided diagnosis system for prostate cancer mapping [NRML$^+$12, ALP$^+$15]. This system learns a binary decision model in a multidimensional feature space based on training samples (voxels) from different classes of interest. This model is then used to generate cancer probability maps.

Data description The considered source and target data are mp-MRI exams of 90 patients acquired on two different MRI scanners (1.5 T and 3T) and thus leading to images with different resolution and texture patterns. Details are given in Table 2.

Scanner (domain)	1.5T	3T
Number of patients	49	41
Number of voxels	419348	987396
% of positive class	13.38	14.26

Table 2: mp-MRI data description.

Each individual voxel is described by a binary label (Cancer, Non Cancer) and a set of 95 handcrafted features consisting of image descriptors, texture coefficients, gradients and other visual characteristics (more details in [NRML$^+$12]).

To tackle the class imbalance illustrated in Table 2, we randomly select a balanced 2000 voxels data set for each of the 10 repeated reverse validation procedures. Moreover, we run a principal component analysis keeping 95% of the total variance, and reducing the number of features to 28. With the best found hyperparameters, our algorithm is evaluated 10 times on independent data sets of 10000 instances for each domain. Results are reported in Table 3, where we compare our algorithm with a linear SVM without adaptation

(`Scikit-Learn` [PVG+11]'s implementation). We notice that for both domain adaptation where we consider the 3T and 1.5T domains as source and target interchangeably, the scores on the target domain are close, reflecting a symmetry of the problem. Our method exhibits an improvement of 11% over the case without adaptation.

Problem	3T→1.5T	1.5T→3T
Linear SVM (no adaptation)	49.99	49.56
Our algorithm	38.99	38.39

Table 3: Average 0-1 loss (percentage) over 10 realizations for the mp-MRI data set.

5 Conclusion and future perspectives

In this paper, we presented a new algorithm for domain adaptation that is suitable for large margin classifiers. Our algorithm is derived from a theoretical bound and allows to learn a classifier on the target domain directly without an additional alignment step. In the case of bilinear similarity functions and linear classifiers, it is formulated as a quadratic optimization problem that is solved efficiently. The results we obtain are encouraging, although not surpassing state of the art methods ([CFTR15], [MHA12]), and suggest trying to obtain a tighter bound from which the algorithm is derived or using a domain disagreement term that is less strict than a supremum. For example, considering an expectation over the considered set of classifiers [GHLM17] is worth a try. Moreover, in the context of large margin classifiers, questions about our method's generalization guarantees are naturally raised. Finally, the multi-class case and the semi-supervised domain adaptation (i.e when some labels are available on the target) are crucial extensions to add as they would allow us to test on more real world data sets.

References

[ALP+15] Rahaf Aljundi, Jérôme Lehaire, Fabrice Prost-Boucle, Olivier Rouvière, and Carole Lartizien. Transfer learning for prostate cancer mapping based on multicentric MR imaging databases. In *Machine Learning Meets Medical Imaging workshop at ICML*, pages 74–82, 2015.

[ApS17] MOSEK ApS. *MOSEK Optimizer API for Python 8.1.0.80*, 2017.

[AVDB18] Akshay Agrawal, Robin Verschueren, Steven Diamond, and Stephen Boyd. A rewriting system for convex optimization problems. *Journal of Control and Decision*, 5(1):42–60, 2018.

[BBS08a] Maria-Florina Balcan, Avrim Blum, and Nathan Srebro. *Improved guarantees for learning via similarity functions.* 2008.

[BBS08b] Maria-Florina Balcan, Avrim Blum, and Nathan Srebro. A theory of learning with similarity functions. *Machine Learning*, 72(1), 2008.

[BDBC+10] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning*, 79(1), 2010.

[BGV92] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A Training Algorithm for Optimal Margin Classifiers. In *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, 1992.

[BHS12] Aurélien Bellet, Amaury Habrard, and Marc Sebban. Similarity Learning for Provably Accurate Sparse Linear Classification. In *ICML*, 2012.

[BM10] L. Bruzzone and M. Marconcini. Domain Adaptation Problems: A DASVM Classification Technique and a Circular Validation Strategy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(5), 2010.

[CFTR15] Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal Transport for Domain Adaptation. *arXiv:1507.00504 [cs]*, 2015. arXiv: 1507.00504.

[CH67] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1), 1967.

[CV95] Corinna Cortes and Vladimir Vapnik. Support-Vector Networks. *Machine Learning*, 20(3), 1995.

[DB16] Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

[DR18] Sofiane Dhouib and Ievgen Redko. Revisiting (ϵ, γ, τ)-similarity learning for domain adaptation. In *Advances in Neural Information Processing Systems 31*. 2018.

[GHLM13] Pascal Germain, Amaury Habrard, François Laviolette, and Emilie Morvant. A PAC-Bayesian Approach for Domain Adaptation with Specialization to Linear Classifiers. In *International Conference on Machine Learning*, 2013.

[GHLM17] Pascal Germain, Amaury Habrard, François Laviolette, and Emilie Morvant. PAC-Bayes and Domain Adaptation. *arXiv:1707.05712 [stat]*, 2017. arXiv: 1707.05712.

[GY13] Zheng-Chu Guo and Yiming Ying. Guaranteed Classification via Regularized Similarity Learning. *arXiv:1306.3108 [cs]*, 2013. arXiv: 1306.3108.

[ISH+15] Nicolae Irina, Marc Sebban, Amaury Habrard, Eric Gaussier, and Massih-Reza Amini. Algorithmic Robustness for Semi-Supervised (ϵ, γ, τ)-Good Metric Learning. In *ICONIP*, 2015.

[KJ11] Purushottam Kar and Prateek Jain. Similarity-based Learning via Data Driven Embeddings. In *Advances in Neural Information Processing Systems 24*. 2011.

[Mar11] Anna Margolis. A Literature Review of Domain Adaptation with Unlabeled Data. 2011.

[MHA12] Emilie Morvant, Amaury Habrard, and Stéphane Ayache. Parsimonious unsupervised and semi-supervised domain adaptation with good similarity functions. *Knowledge and Information Systems*, 33(2), 2012.

[MMR09] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain Adaptation: Learning Bounds and Algorithms. *arXiv:0902.3430 [cs]*, 2009. arXiv: 0902.3430.

[NGHS15] Maria-Irina Nicolae, Éric Gaussier, Amaury Habrard, and Marc Sebban. Joint Semi-supervised Similarity Learning for Linear Classification. In *ECML/PKDD*, volume 9284. 2015.

[NRML+12] Emilie Niaf, Olivier Rouvière, Florence Mège-Lechevallier, Flavie Bratan, and Carole Lartizien. Computer-aided diagnosis of prostate cancer in the peripheral zone using multiparametric mri. *Physics in medicine and biology*, 57(12):3833–51, 2012.

[PVG+11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[PY10] S. J. Pan and Q. Yang. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10), 2010.

[SKM07] Masashi Sugiyama, Matthias Krauledat, and Klaus-Robert Müller. Covariate Shift Adaptation by Importance Weighted Cross Validation. *Journal of Machine Learning Research*, 8(May), 2007.

[SSM97] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Kernel principal component analysis. In *Artificial Neural Networks — ICANN'97*, Lecture Notes in Computer Science, 1997.

[WKW16] Karl Weiss, Taghi M. Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big Data*, 3(1), 2016.

[ZFY+10] Erheng Zhong, Wei Fan, Qiang Yang, Olivier Verscheure, and Jiangtao Ren. Cross Validation Framework to Choose amongst Models and Datasets for Transfer Learning. In *ECML/PKDD*, volume 6323. 2010.