Domain Adaptation in Regression

Corinna Cortes[1] and Mehryar Mohri[2,1]

[1] Google Research,
76 Ninth Avenue, New York, NY 10011.
[2] Courant Institute of Mathematical Sciences,
251 Mercer Street, New York, NY 10012.

Abstract. This paper presents a series of new results for domain adaptation in the regression setting. We prove that the discrepancy is a distance for the squared loss when the hypothesis set is the reproducing kernel Hilbert space induced by a universal kernel such as the Gaussian kernel. We give new pointwise loss guarantees based on the discrepancy of the empirical source and target distributions for the general class of kernel-based regularization algorithms. These bounds have a simpler form than previous results and hold for a broader class of convex loss functions not necessarily differentiable, including L_q losses and the hinge loss. We extend the discrepancy minimization adaptation algorithm to the more significant case where kernels are used and show that the problem can be cast as an SDP similar to the one in the feature space. We also show that techniques from smooth optimization can be used to derive an efficient algorithm for solving such SDPs even for very high-dimensional feature spaces. We have implemented this algorithm and report the results of experiments demonstrating its benefits for adaptation and show that, unlike previous algorithms, it can scale to large data sets of tens of thousands or more points.

1 Introduction

A standard assumption in learning theory and applications is that training and test points are drawn according to the same distribution. But, a more challenging problem of *domain adaptation* arises in a variety of applications, including natural language processing, speech processing, or computer vision [7, 3, 9, 10, 17, 18, 12]. This problem occurs when little or no labeled data is available from the *target domain*, but labeled data from a *source domain* somewhat similar to the target, as well as large amounts of unlabeled data from the target domain, are accessible. The domain adaptation problem then consists of using the source labeled and target unlabeled data to learn a hypothesis performing well on the target domain.

The theoretical analysis of this problem has been the topic of some recent publications. An analysis of adaptation was initiated by Ben-David et al. [1]. Several issues of that paper were later corrected by Blitzer et al. [4]. These authors gave VC-dimension bounds for binary classification based on a d_A distance between distributions that can be estimated from finite samples, and a term λ_H depending on the distributions and the hypothesis set H, which cannot be estimated from data. In [11], we presented alternative learning bounds which hold in particular in the classification setting and depend on the optimal classifiers in the hypothesis set for the source and target distributions.

Our bounds are in general not comparable to those of [1, 4] but we showed that under some plausible assumptions they are superior to those of [1, 4] and that in many cases the bounds of [1, 4] have a factor of 3 of the error that can make them vacuous. The assumptions made in the analysis of adaptation were more recently discussed by Ben-David et al. [2] who also presented several negative results for this problem. These negative results hold only for the 0-1 loss used in classification.

This paper deals with the problem of adaptation in regression, for which many of the observations made in the case of the 0-1 loss do not hold. In [11], we introduced a distance between distributions specifically tailored to domain adaptation, the *discrepancy distance*, which generalizes the d_A distance to arbitrary loss functions, and presented several theoretical guarantees based on that discrepancy, including data-dependent Rademacher complexity generalization bounds. In this paper we present a series of novel results for domain adaptation in regression extending those of [11] and making them more significant and practically applicable.

In Section 2, we describe more formally the learning scenario of domain adaptation in regression and briefly review the definition and key properties of the discrepancy. We then present several theoretical results in Section 3. For the squared loss, we prove that the discrepancy is a distance when the hypothesis set is the reproducing kernel Hilbert space of a universal kernel, such as a Gaussian kernel. This implies that minimizing the discrepancy to zero guarantees matching the target distribution, a result that does not hold in the case of the 0-1 loss. We further give pointwise loss guarantees depending on the discrepancy of the empirical source and target distributions for the class of kernel-based regularization algorithms, including kernel ridge regression, support vector machines (SVMs), or support vector regression (SVR). These bounds have a simpler form than a previous result we presented in the specific case of the squared loss in [11] and hold for a broader class of convex loss functions not necessarily differentiable, which includes all L_q losses ($q \geq 1$), but also the hinge loss used in classification.

When the magnitude of the difference between the source and target labeling functions is small on the training set, these bounds provide a strong guarantee based on the empirical discrepancy and suggest an empirical discrepancy minimization algorithm [11]. In Section 4, we extend the discrepancy minimization adaptation algorithm with the squared loss to the more significant case where kernels are used. We show that the problem can be cast as a semi-definite programming (SDP) problem similar to the one given in [11] in the feature space, but formulated only in terms of the kernel matrix.

Such SDP optimization problems can only be solved practically for modest sample sizes of a few hundred points using existing solvers, even with the most efficient publicly available one. In Section 5, we prove, however, that an algorithm with significantly better time and space complexities can be derived to solve these SDPs using techniques from smooth optimization [14]. We describe the algorithm in detail. We prove a bound on the number of iterations and analyze the computational cost of each iteration.

We have implemented that algorithm and carried out extensive experiments showing that it can indeed scale to large data sets of tens of thousands or more points. Our kernelized version of the SDP further enables us to run the algorithm for very high-dimensional and even infinite-dimensional feature spaces. Section 6 reports our empirical results demonstrating the effectiveness of this algorithm for domain adaptation.

2 Preliminaries

This section describes the learning scenario of domain adaptation and reviews the key definitions and properties of the *discrepancy distance* between distributions.

2.1 Learning Scenario

Let X denote the input space and Y the output space, a measurable subset of $\mathbb{R}$, as in standard regression problems. In the adaptation problem we are considering, there are different *domains*, defined by a distribution over X and a target labeling function mapping from X to Y. We denote by Q the distribution over X for the *source domain* and by $f_Q \colon X \to Y$ the corresponding labeling function. Similarly, we denote by P the distribution over X for the *target domain* and by f_P the target labeling function. When the two domains share the same labeling function, we simply denote it by f.

In the *domain adaptation problem* in regression, the learning algorithm receives a labeled sample of m points $\mathcal{S} = ((x_1, y_1), \ldots, (x_m, y_m)) \in (X \times Y)^m$ from the source domain, that is $x_1, \ldots, x_m$ are drawn i.i.d. according to Q and $y_i = f_Q(x_i)$ for $i \in [1, m]$. We denote by $\widehat{Q}$ the empirical distribution corresponding to $x_1, \ldots, x_m$. Unlike the standard supervised learning setting, the test points are drawn from the target domain, which is based on a different input distribution P and possibly different labeling function f_P. The learner is additionally provided with an unlabeled sample $\mathcal{T}$ of size n drawn i.i.d. according to the target distribution P. We denote by $\widehat{P}$ the empirical distribution corresponding to $\mathcal{T}$.

We consider a loss function $L \colon Y \times Y \to \mathbb{R}_+$ that is symmetric and convex with respect to each of its argument. In particular L may be the squared loss commonly used in regression. For any two functions $h, h' \colon X \to Y$ and any distribution D over X, we denote by $\mathcal{L}_D(h, h')$ the expected loss of $h(x)$ and $h'(x)$:

$$\mathcal{L}_D(h, h') = \mathop{\mathrm{E}}_{x \sim D}[L(h(x), h'(x))]. \tag{1}$$

The domain adaptation problem consists of selecting a hypothesis h out of a hypothesis set H with a small expected loss according to the target distribution P, $\mathcal{L}_P(h, f_P)$.

2.2 Discrepancy Distance

A key question for adaptation is a measure of the difference between the distributions Q and P. As pointed out in [11], a general-purpose measure such as the L_1 distance is not helpful in this context since the L_1 distance can be large even in some rather favorable situations for adaptation. Furthermore, this distance cannot be accurately estimated from finite samples and ignores the loss function. Instead, the discrepancy provides a measure of the dissimilarity of two distributions that is specifically tailored to adaptation and is defined based on the loss function and the hypothesis set used.

Observe that for a fixed hypothesis $h \in H$, the quantity of interest in adaptation is the difference of expected losses $|\mathcal{L}_P(f_P, h) - \mathcal{L}_Q(f_P, h)|$. A natural distance between distributions in this context is thus one based on the supremum of this quantity over all $h \in H$. The target hypothesis f_P is unknown and could match any hypothesis h'. This leads to the following definition [11].

Definition 1. *Given a hypothesis set H and loss function L, the* discrepancy distance disc *between two distributions P and Q over X is defined by:*

$$\operatorname{disc}(P, Q) = \max_{h,h' \in H} \big|\mathcal{L}_P(h', h) - \mathcal{L}_Q(h', h)\big|. \tag{2}$$

The discrepancy is by definition symmetric and verifies the triangle inequality for any loss function L. But, in general, it does not define a *distance* since we may have $\operatorname{disc}(P, Q) = 0$ for $P \neq Q$. We shall prove, however, that for a large family of kernel-based hypothesis set, it does verify all the axioms of a distance.

3 Theoretical Analysis

In what follows, we consider the case where the hypothesis set H is a subset of the reproducing kernel Hilbert space (RKHS) $\mathbb{H}$ associated to a positive definite symmetric (PDS) kernel K: $H = \{h \in \mathbb{H} : \|h\|_K \leq \Lambda\}$, where $\| \cdot \|_K$ denotes the norm defined by the inner product on $\mathbb{H}$ and $\Lambda \geq 0$. We shall assume that there exists $R > 0$ such that $K(x, x) \leq R^2$ for all $x \in X$. By the reproducing property, for any $h \in H$ and $x \in X$, $h(x) = \langle h, K(x, \cdot)\rangle_K$, thus this implies that $|h(x)| \leq \|h\|_K \sqrt{K(x, x)} \leq \Lambda R$.

3.1 Discrepancy with universal kernels

We first prove that for a *universal kernel* K, such as a Gaussian kernel [20], the discrepancy defines a distance. Let $C(X)$ denote the set of all continuous functions mapping X to $\mathbb{R}$. We shall assume that X is a compact set, thus the functions in $C(X)$ are also bounded. A PDS kernel K over $X \times X$ is said to be universal if it is continuous and if the RKHS $\mathbb{H}$ it induces is dense in $C(X)$ for the norm infinity $\| \cdot \|_\infty$.

Theorem 1. *Let L be the squared loss and let K be a universal kernel. Then, for any two distributions P and Q, if $\operatorname{disc}(P, Q) = 0$, then $P = Q$.*

Proof. Consider the function $\Psi : C(X) \to \mathbb{R}$ defined for any $h \in C(X)$ by $\Psi(h) = \mathrm{E}_{x\sim P}[h^2] - \mathrm{E}_{x\sim Q}[h^2]$. Ψ is continuous for the norm infinity over $C(X)$ since $h \mapsto \mathrm{E}_{x\sim P}[h^2]$ is continuous. Indeed, for any $h, h' \in H$,

$$|\operatorname*{E}_P[h'^2] - \operatorname*{E}_P[h^2]| = |\operatorname*{E}_P[(h' + h)(h' - h)]| \leq (\|h\|_\infty + \|h'\|_\infty)\|h' - h\|_\infty,$$

and similarly with $h \mapsto \mathrm{E}_{x\sim Q}[h^2]$. If $\operatorname{disc}(P, Q) = 0$, then, by definition,

$$\forall h, h' \in H, \quad \big|\operatorname*{E}_{x\sim P}[(h'(x) - h(x))^2] - \operatorname*{E}_{x\sim Q}[(h'(x) - h(x))^2]\big| = 0.$$

Thus, $\mathrm{E}_P[h''^2] - \mathrm{E}_Q[h''^2] = 0$ for any $h'' = h' - h \in \mathbb{H}$ with $\|h''\|_K \leq 2\Lambda R$, therefore for any $h'' \in \mathbb{H}$ with $\|h''\|_K \leq 2\Lambda R$, hence for any $h'' \in \mathbb{H}$ regardless of the norm. Thus, $\Psi = 0$ over $\mathbb{H}$. Since K is universal, $\mathbb{H}$ is dense in $C(X)$ for the norm $\| \cdot \|_\infty$ and by continuity of Ψ for $\| \cdot \|_\infty$, for all $h \in C(X)$, $\mathrm{E}_P[h^2] - \mathrm{E}_Q[h^2] = 0$. Let f be any non-negative function in $C(X)$, then $\sqrt{f}$ is well defined and is in $C(X)$, thus,

$$\operatorname*{E}_P[(\sqrt{f})^2] - \operatorname*{E}_Q[(\sqrt{f})^2] = \operatorname*{E}_P[f] - \operatorname*{E}_Q[f] = 0.$$

It is known that if $\mathrm{E}_P[f] - \mathrm{E}_Q[f] = 0$ for all $f \in C(X)$ with $f \geq 0$, then $P = Q$ (see [8][proof of lemma 9.3.2]). This concludes the proof. □

Thus, the theorem shows that if we could find a source distribution Q that would reduce to zero the discrepancy in the case of the familiar Gaussian kernels, then that distribution would in fact match the target distribution P.

3.2 Guarantees for kernel-based regularization algorithms

We now present pointwise loss guarantees in domain adaptation for a broad class of kernel-based regularization algorithms, which also demonstrate the key role played by the discrepancy in adaptation and suggest the benefits of minimizing that quantity. These algorithms are defined by the minimization of the following objective function:

$$F_{\widehat{Q}}(h) = \widehat{R}_{\widehat{Q}}(h) + \lambda \|h\|_K^2, \tag{3}$$

where $\lambda \geq 0$ is a trade-off parameter and $R_{\widehat{Q}}(h) = \frac{1}{m}\sum_{i=1}^m L(h(x_i), y_i)$ the empirical error of hypothesis $h \in \mathbb{H}$. This family of algorithms includes support vector machines (SVM) [6], support vector regression (SVR) [21], kernel ridge regression (KRR) [19], and many other algorithms. We shall assume that the loss function L is μ-admissible for some $\mu > 0$: that is, it is symmetric and convex with respect to both of its arguments and for all $x \in X$ and $y \in Y$ and $h, h' \in H$, it verifies the following Lipschitz condition:

$$|L(h'(x), y) - L(h(x), y)| \leq \mu |h'(x) - h(x)|.$$

μ-admissible losses include the hinge loss and all L_q losses with $q \geq 1$, in particular the squared loss, when the hypothesis set and the set of output labels are bounded.

The labeling functions f_P and f_Q may not coincide on the training set $\operatorname{supp}(\widehat{Q})$. But, for adaptation to be possible, the difference between the labels received for the training points and their target values should be assumed to be small, even if the input space distributions P and Q are very close.

When the magnitude of the difference between the source and target labeling functions is small on the training set, that is $\eta = \max\{L(f_Q(x), f_P(x)) \colon x \in \operatorname{supp}(\widehat{Q})\} \ll 1$, the following theorem gives a strong guarantee on the pointwise difference of the loss between the hypothesis h returned by the algorithm when training on the source domain and the hypothesis h' returned when training on a sample drawn from the target distribution in terms of the empirical discrepancy $\operatorname{disc}(\widehat{P}, \widehat{Q})$. The theorem holds for all μ-admissible losses and has a simpler form than a previous result we presented in [11].

Theorem 2. *Let L be a μ-admissible loss. Assume that $f_p \in H$ and let η denote $\max\{L(f_Q(x), f_P(x)) \colon x \in \operatorname{supp}(\widehat{Q})\}$. Let h' be the hypothesis returned by the kernel-based regularization algorithm* (3) *when minimizing $F_{\widehat{P}}$ and h the one returned when minimizing $F_{\widehat{Q}}$. Then, for all $x \in X$ and $y \in Y$,*

$$\big|L(h'(x), y) - L(h(x), y)\big| \leq \mu R \sqrt{\frac{\operatorname{disc}(\widehat{P}, \widehat{Q}) + \mu\eta}{\lambda}}. \tag{4}$$

Proof. The proof makes use of a generalized Bregman divergence, which we first introduce. For a convex function $F \colon \mathbb{H} \to \mathbb{R}$, we denote by $\partial F(h)$ the subgradient of F at h: $\partial F(h) = \{g \in \mathbb{H} \colon \forall h' \in \mathbb{H}, F(h') - F(h) \geq \langle h' - h, g \rangle\}$. $\partial F(h)$ coincides with $\nabla F(h)$ when F is differentiable at h. Note that at a point h where F is minimal, 0 is an element of $\partial F(h)$. Furthermore, the subgradient is additive, that is, for two convex function F_1 and F_2, $\partial(F_1 + F_2)(h) = \{g_1 + g_2 \colon g_1 \in \partial F_1(h), g_2 \in \partial F_2(h)\}$. For any $h \in \mathbb{H}$, fix $\delta F(h)$ to be an (arbitrary) element of $\partial F(h)$. For any such choice of δF, we can define the *generalized Bregman divergence* associated to F by:

$$\forall h', h \in \mathbb{H}, B_F(h' \| h) = F(h') - F(h) - \langle h' - h, \delta F(h) \rangle . \tag{5}$$

Note that by definition of the subgradient, $B_F(h' \| h) \geq 0$ for all $h', h \in \mathbb{H}$. Let N denote the convex function $h \rightarrow \|h\|_K^2$. Since N is differentiable, $\delta N(h) = \nabla N(h)$ for all $h \in \mathbb{H}$, and δN and thus B_N are uniquely defined. To make the definition of the Bregman divergences for $F_{\widehat{Q}}$ and $\widehat{R}_{\widehat{Q}}$ compatible so that $B_{F_{\widehat{Q}}} = B_{\widehat{R}_{\widehat{Q}}} + \lambda B_N$, we define $\delta \widehat{R}_{\widehat{Q}}$ from $\delta F_{\widehat{Q}}$ by: $\delta \widehat{R}_{\widehat{Q}}(h) = \delta F_{\widehat{Q}}(h) - \lambda \nabla N(h)$ for all $h \in \mathbb{H}$. Furthermore, we choose $\delta F_{\widehat{Q}}(h)$ to be 0 for any point h where $F_{\widehat{Q}}$ is minimal and let $\delta F_{\widehat{Q}}(h)$ be an arbitrary element of $\partial F_{\widehat{Q}}(h)$ for all other hs. We proceed in a similar way to define the Bregman divergences for $F_{\widehat{P}}$ and $\widehat{R}_{\widehat{P}}$ so that $B_{F_{\widehat{P}}} = B_{\widehat{R}_{\widehat{P}}} + \lambda B_N$.

Since the generalized Bregman divergence is non-negative and since $B_{F_{\widehat{Q}}} = B_{\widehat{R}_{\widehat{Q}}} + \lambda B_N$ and $B_{F_{\widehat{P}}} = B_{\widehat{R}_{\widehat{P}}} + \lambda B_N$, we can write

$$B_{F_{\widehat{Q}}}(h' \| h) + B_{F_{\widehat{P}}}(h \| h') \geq \lambda \big(B_N(h' \| h) + B_N(h \| h') \big).$$

Observe that $B_N(h' \| h) + B_N(h \| h') = - \langle h' - h, 2h \rangle - \langle h - h', 2h' \rangle = 2\|h' - h\|_K^2$. Thus, $B_{F_{\widehat{Q}}}(h' \| h) + B_{F_{\widehat{P}}}(h \| h') \geq 2\lambda \|h' - h\|_K^2$. By definition of h' and h as minimizers and our choice of the subgradients, $\delta F_{\widehat{P}}(h') = 0$ and $\delta F_{\widehat{Q}}(h) = 0$, thus, this inequality can be rewritten as follows:

$$2\lambda \|h' - h\|_K^2 \leq \widehat{R}_{\widehat{Q}}(h') - \widehat{R}_{\widehat{Q}}(h) + \widehat{R}_{\widehat{P}}(h) - \widehat{R}_{\widehat{P}}(h').$$

Now, rewriting this inequality in terms of the expected losses gives:

$$\begin{aligned} 2\lambda \|h' - h\|_K^2 &\leq \big(\mathcal{L}_{\widehat{P}}(h, f_P) - \mathcal{L}_{\widehat{Q}}(h, f_Q) \big) - \big(\mathcal{L}_{\widehat{P}}(h', f_P) - \mathcal{L}_{\widehat{Q}}(h', f_Q) \big) \\ &= \big(\mathcal{L}_{\widehat{P}}(h, f_P) - \mathcal{L}_{\widehat{Q}}(h, f_P) \big) - \big(\mathcal{L}_{\widehat{P}}(h', f_P) - \mathcal{L}_{\widehat{Q}}(h', f_P) \big) \\ &+ \big(\mathcal{L}_{\widehat{Q}}(h, f_P) - \mathcal{L}_{\widehat{Q}}(h, f_Q) \big) - \big(\mathcal{L}_{\widehat{Q}}(h', f_P) - \mathcal{L}_{\widehat{Q}}(h', f_Q) \big). \end{aligned}$$

Since f_P is in H, by definition of the discrepancy, the first two terms can both be bounded by the empirical discrepancy:

$$\Big| \big(\mathcal{L}_{\widehat{P}}(h, f_P) - \mathcal{L}_{\widehat{Q}}(h, f_P) \big) \Big| \leq \text{disc}(\widehat{P}, \widehat{Q}) \text{ and } \Big| \big(\mathcal{L}_{\widehat{P}}(h', f_P) - \mathcal{L}_{\widehat{Q}}(h', f_P) \big) \Big| \leq \text{disc}(\widehat{P}, \widehat{Q}).$$

The last two terms can be bounded using the μ-admissibility of L since for any $h'' \in H$,

$$\Big| \big(\mathcal{L}_{\widehat{Q}}(h'', f_P) - \mathcal{L}_{\widehat{Q}}(h'', f_Q) \big) \Big| \leq \mu \operatorname*{E}_{x \sim \widehat{Q}} [|f_P(x) - f_Q(x)|] \leq \mu\eta.$$

Thus,
$$2\lambda \|h' - h\|_K^2 \leq 2\text{disc}(\widehat{P}, \widehat{Q}) + 2\mu\eta. \tag{6}$$

By the reproducing property, for any $x \in X$, $h' - h(x) = \langle h' - h, K(x, \cdot) \rangle$, thus, for any $x \in X$ and $y \in Y$, $\big| L(h'(x), y) - L(h(x), y) \big| \leq \mu |h' - h(x)| \leq \mu R \|h' - h\|_K$. Upper bounding the right-hand side using (6) directly yields the statement (4). □

A similar theorem can be proven when only $f_Q \in H$ is assumed. These theorems can be extended to the case where neither the target function f_P nor f_Q is in H by replacing in Theorem 2 η with $\eta'' = \max\{L(h_P^*(x), f_Q(x)) \colon x \in \operatorname{supp}(\widehat{Q})\} + \max\{L(h_P^*(x), f_P(x)) \colon x \in \operatorname{supp}(\widehat{P})\}$, where $h_P^* \in \operatorname{argmin}_{h \in H} \mathcal{L}_P(h, f_P)$. They show the key role played by the empirical discrepancy $\operatorname{disc}(\widehat{P}, \widehat{Q})$ in this context when $\eta'' \ll 1$. Note that $\eta = 0$ when $f_P = f_Q = f$ as in the sample bias correction setting or other scenarios where the so-called covariate-shift assumption hold. Under the assumptions $\eta \ll 1$ or $\eta'' \ll 1$, these theorems suggest seeking an empirical distribution q^*, among the family $\mathcal{Q}$ of all distributions with a support included in that of $\widehat{Q}$, that minimizes that discrepancy [11]:

$$q^* = \operatorname*{argmin}_{q \in \mathcal{Q}} \operatorname{disc}(\widehat{P}, q). \tag{7}$$

Using q^* instead of $\widehat{Q}$ amounts to reweighting the loss on each training point. This forms the basis of our adaptation algorithm which consists of: (a) first computing q^*; (b) then modifying (3) using q^*:

$$F_{q^*}(h) = \frac{1}{m} \sum_{i=1}^{m} q^*(x_i) L(h(x_i), y_i) + \lambda \|h\|_K^2, \tag{8}$$

and finding a minimizing h. The minimization of F_{q^*} is no more difficult than that of $F_{\widehat{Q}}$ and is standard. Thus, in the following section, we focus on the first stage of our algorithm and study in detail the optimization problem (7).

4 Optimization Problems

Let X be a subset of $\mathbb{R}^N$, $N > 1$. We denote by S_Q the support of $\widehat{Q}$, by S_P the support of $\widehat{P}$, and by S their union $\operatorname{supp}(\widehat{Q}) \cup \operatorname{supp}(\widehat{P})$, with $|S_Q| = \mathfrak{m} \leq m$ and $|S_P| = \mathfrak{n} \leq n$. The unique elements of S_P are denoted by $\mathbf{x}_1, \ldots, \mathbf{x}_{\mathfrak{m}}$ and those of S_P by $\mathbf{x}_{\mathfrak{m}+1}, \ldots, \mathbf{x}_{\mathfrak{q}}$, with $\mathfrak{q} = \mathfrak{m} + \mathfrak{n}$. For a vector $\mathbf{z} \in \mathbb{R}^{\mathfrak{m}}$, we denote by z_i its ith coordinate. We also denote by $\Delta_{\mathfrak{m}}$ the simplex in $\mathbb{R}^{\mathfrak{m}}$: $\Delta_{\mathfrak{m}} = \{\mathbf{z} \in \mathbb{R}^{\mathfrak{m}} \colon z_i \geq 0 \wedge \sum_{i=1}^{\mathfrak{m}} z_i = 1\}$.

4.1 Discrepancy minimization in feature space

We showed in [11] that the problem of minimizing the empirical discrepancy for the squared loss and the hypothesis space $H = \{\mathbf{x} \mapsto \mathbf{w}^\top \mathbf{x} \colon \|\mathbf{w}\| \leq \Lambda\}$ of bounded linear functions can be cast as the following convex optimization problem:

$$\min_{\mathbf{z} \in \Delta_{\mathfrak{m}}} \quad \|\mathbf{M}(\mathbf{z})\|_2, \tag{9}$$

where $\mathbf{M}(\mathbf{z}) \in \mathbb{S}^N$ is a symmetric matrix that is an affine function of $\mathbf{z}$:

$$\mathbf{M}(\mathbf{z}) = \mathbf{M}_0 - \sum_{i=1}^{\mathfrak{m}} z_i \mathbf{M}_i, \tag{10}$$

with $\mathbf{M}_0 = \sum_{j=\mathfrak{m}+1}^{\mathfrak{q}} \widehat{P}(\mathbf{x}_j) \mathbf{x}_j \mathbf{x}_j^\top$ and for $i \in [1, \mathfrak{m}]$ $\mathbf{M}_i = \mathbf{x}_i \mathbf{x}_i^\top$, $\mathbf{x}_i \in S_Q$. The minimal discrepancy distribution q^* is given by $q^*(\mathbf{x}_i) = z_i$, for all $i \in [1, \mathfrak{m}]$. Since $\|\mathbf{M}(\mathbf{z})\|_2 =$

$\max\{\lambda_{\max}(\mathbf{M}(\mathbf{z})), \lambda_{\max}(-\mathbf{M}(\mathbf{z}))\}$, the problem can be rewritten equivalently as the following semi-definite programming (SDP) problem:

$$\min_{\mathbf{z},t} \quad t \qquad (11)$$
$$\text{subject to} \quad \begin{bmatrix} t\mathbf{I} & \mathbf{M}(\mathbf{z}) \\ \mathbf{M}(\mathbf{z}) & t\mathbf{I} \end{bmatrix} \succeq 0 \ \wedge \ \mathbf{1}^\top \mathbf{z} = 1 \ \wedge \ \mathbf{z} \geq 0.$$

This problem can be solved in polynomial time using interior point methods. The time complexity for each iteration of the algorithm is in our notation [16][pp.234-235] : $O(\mathfrak{m}^3 + \mathfrak{m}N^3 + \mathfrak{m}^2N^2 + \mathfrak{n}N^2)$. This time complexity as well as its space complexity, which is in $O((\mathfrak{m} + N)^2)$, make such algorithms impractical for relatively large or realistic machine learning problems.

4.2 Discrepancy minimization with kernels

Here, we prove that the results of the previous section can be generalized to the case of high-dimensional feature spaces defined implicitly by a PDS kernel K. We denote by $\mathbf{K} = [K(\mathbf{x}_i, \mathbf{x}_j)]_{ij} \in \mathbb{R}^{\mathfrak{q}\times\mathfrak{q}}$ the kernel matrix associated to K for the full sample $S = S_Q \cup S_P$ and for any $\mathbf{z} \in \mathbb{R}^{\mathfrak{m}}$ by $\mathbf{D}(\mathbf{z})$ the diagonal matrix

$$\mathbf{D}(\mathbf{z}) = \mathrm{diag}(-z_1, \ldots, -z_{\mathfrak{m}}, \widehat{P}(\mathbf{x}_{\mathfrak{m}+1}), \ldots, \widehat{P}(\mathbf{x}_{\mathfrak{m}+\mathfrak{n}})).$$

Theorem 3. *For any $\widehat{Q}$ and $\widehat{P}$, the problem of determining the discrepancy minimizing distribution q^* for the squared loss L_2 and the hypothesis set H can be cast as an SDP of the same form as* (9) *but that depends only on the Gram matrix of the kernel K:*

$$\min_{\mathbf{z}\in\Delta_{\mathfrak{m}}} \quad \|\mathbf{M}'(\mathbf{z})\|_2 \qquad (12)$$

where $\mathbf{M}'(\mathbf{z}) = \mathbf{K}^{1/2}\mathbf{D}(\mathbf{z})\mathbf{K}^{1/2} = \mathbf{M}'_0 - \sum_{i=1}^{\mathfrak{m}} z_i\mathbf{M}'_i$, with $\mathbf{M}'_0 = \mathbf{K}^{1/2}\mathbf{D}_0\mathbf{K}^{1/2}$ and $\mathbf{M}'_i = \mathbf{K}^{1/2}\mathbf{D}_i\mathbf{K}^{1/2}$ for $i \in [1, \mathfrak{m}]$, and $\mathbf{D}_0, \mathbf{D}_1, \ldots, \mathbf{D}_{\mathfrak{m}} \in \mathbb{R}^{\mathfrak{q}\times\mathfrak{q}}$ defined by $\mathbf{D}_0 = \mathrm{diag}(0, \ldots, 0, \widehat{P}(\mathbf{x}_{\mathfrak{m}+1}), \ldots, \widehat{P}(\mathbf{x}_{\mathfrak{m}+\mathfrak{n}}))$, and for $i \geq 1$, $\mathbf{D}_i$ is the diagonal matrix of the ith unit vector.

Proof. Let $\Phi\colon X \to \mathcal{F}$ be a feature mapping associated to K, with $\dim(\mathcal{F}) = N'$. Let $\mathfrak{q} = \mathfrak{m} + \mathfrak{n}$. The problem of finding the optimal distribution q^* is equivalent to solving

$$\min_{\substack{\|\mathbf{z}\|_1=1\\ \mathbf{z}\geq 0}} \{\lambda_{\max}(\mathbf{M}(\mathbf{z})), \lambda_{\max}(-\mathbf{M}(\mathbf{z}))\}, \qquad (13)$$

where the matrix $\mathbf{M}(\mathbf{z})$ is defined by

$$\mathbf{M}(\mathbf{z}) = \sum_{i=\mathfrak{m}+1}^{\mathfrak{q}} \widehat{P}(\mathbf{x}_i)\Phi(\mathbf{x}_i)\Phi(\mathbf{x}_i)^\top - \sum_{i=1}^{\mathfrak{m}} z_i\Phi(\mathbf{x}_i)\Phi(\mathbf{x}_i)^\top,$$

with q^* given by: $q^*(\mathbf{x}_i) = z_i$ for all $i \in [1, \mathfrak{m}]$. Let $\mathbf{\Phi}$ denote the matrix in $\mathbb{R}^{N'\times\mathfrak{q}}$ whose columns are the vectors $\Phi(\mathbf{x}_1), \ldots, \Phi(\mathbf{x}_{\mathfrak{m}+\mathfrak{n}})$. Then, observe that $\mathbf{M}(\mathbf{z})$ can be rewritten as

$$\mathbf{M}(\mathbf{z}) = \mathbf{\Phi}\mathbf{D}(\mathbf{z})\mathbf{\Phi}^\top.$$

```
Algorithm 1
for $k \geq 0$ do
    $\mathbf{v}_k \leftarrow T_C(\mathbf{u}_k)$
    $\mathbf{w}_k \leftarrow \operatorname{argmin}_{\mathbf{u}\in C} \left\{\frac{L}{\sigma} d(\mathbf{u}) + \sum_{i=0}^{k} \frac{i+1}{2} [F(\mathbf{u}_i) + \langle \nabla F(\mathbf{u}_i), \mathbf{u} - \mathbf{u}_i \rangle]\right\}$
    $\mathbf{u}_{k+1} \leftarrow \frac{2}{k+3} \mathbf{w}_k + \frac{k+1}{k+3} \mathbf{v}_k$
end for
```

Fig. 1. Convex optimization algorithm.

It is known that for any two matrices $\mathbf{A} \in \mathbb{R}^{N' \times \mathsf{q}}$ and $\mathbf{B} \in \mathbb{R}^{\mathsf{q} \times N'}$, $\mathbf{AB}$ and $\mathbf{BA}$ have the same eigenvalues. Thus, matrices $\mathbf{M}(\mathbf{z}) = (\mathbf{\Phi D}(\mathbf{z}))\mathbf{\Phi}^\top$ and $\mathbf{\Phi}^\top(\mathbf{\Phi D}(\mathbf{z})) = \mathbf{KD}(\mathbf{z})$ have the same eigenvalues. $\mathbf{KD}(\mathbf{z})$ is not a symmetric matrix. To ensure that we obtain an SDP of the same form as (9) minimizing the spectral norm of a symmetric matrix, we can instead consider the matrix $\mathbf{M}'(\mathbf{z}) = \mathbf{K}^{1/2}\mathbf{D}(\mathbf{z})\mathbf{K}^{1/2}$, which, by the same argument as above has the same eigenvalues as $\mathbf{KD}(\mathbf{z})$ and therefore $\mathbf{M}(\mathbf{z})$. In particular, $\mathbf{M}'(\mathbf{z})$ and $\mathbf{M}(\mathbf{z})$ have the same maximum and minimum eigenvalues, thus, $\|\mathbf{M}(\mathbf{z})\|_2 = \|\mathbf{M}'(\mathbf{z})\|_2$. Since $\mathbf{D} = \mathbf{D}_0 - \sum_{i=1}^{\mathsf{m}} z_i \mathbf{D}_i$, this concludes the proof. □

Thus, the discrepancy minimization problem can be formulated in both the original input space and in the RKHS defined by a PDS kernel K as an SDP of the same form. In the next section, we present a specific study of this SDP and use results from smooth convex optimization as well as specific characteristics of the SDP considered in our case to derive an efficient and practical adaptation algorithm.

5 Algorithm

This section presents an algorithm for solving the discrepancy minimization problem using the smooth approximation technique of Nesterov [14]. A general algorithm was given by Nesterov [13] to solve convex optimization problems of the form

$$\text{minimize}_{\mathbf{z}\in C} F(\mathbf{z}), \tag{14}$$

where C is a closed convex set and F admits a Lipschitz continuous gradient over C in time $O(1/\sqrt{\epsilon})$, which was later proven to be optimal for this class of problems. The pseudocode of the algorithm is given in Figure 1. Here, $T_C(\mathbf{u}) \in C$ denotes for any $\mathbf{u} \in C$, an element of $\operatorname{argmin}_{\mathbf{v}\in C} \langle \nabla F(\mathbf{u}), \mathbf{v} - \mathbf{u}\rangle + \frac{1}{2} L \|\mathbf{v} - \mathbf{u}\|^2$ (the specific choice of the minimizing $\mathbf{v}$ is arbitrary for a given $\mathbf{u}$). d denotes a *prox-function for* C, that is d is a continuous and strongly convex function over C with respect to the norm $\|\cdot\|$ with convexity parameter $\sigma > 0$ and $d(\mathbf{u}_0) = 0$ where $\mathbf{u}_0 = \operatorname{argmin}_{\mathbf{u}\in C} d(\mathbf{u})$. The following convergence guarantee was given for this algorithm [14].

Theorem 4. *Let $\mathbf{z}^*$ be an optimal solution for problem* (14) *and let $\mathbf{v}_k$ be defined as in Algorithm 1, then for any $k \geq 0$, $F(\mathbf{v}_k) - F(\mathbf{z}^*) \leq \frac{4Ld(\mathbf{z}^*)}{\sigma(k+1)(k+2)}$.*

Algorithm 1 can be further used to solve in $O(1/\epsilon)$ optimization problems of the same form where F is a Lipschitz-continuous non-smooth convex function [15]. This can be done by finding a uniform ϵ-approximation of F by a smooth convex function G

Algorithm 2

```
$\mathbf{u}_0 \leftarrow \mathrm{argmin}_{\mathbf{u}\in C}\, \mathbf{u}^{\top}\mathbf{J}\mathbf{u}$
for $k \geq 0$ do
    $\mathbf{v}_k \leftarrow \mathrm{argmin}_{\mathbf{u}\in C}\, \frac{2p-1}{2}(\mathbf{u}-\mathbf{u}_k)^{\top}\mathbf{J}(\mathbf{u}-\mathbf{u}_k) + \nabla G_p(\mathbf{M}(\mathbf{u}_k))^{\top}\mathbf{u}$
    $\mathbf{w}_k \leftarrow \mathrm{argmin}_{\mathbf{u}\in C}\, \frac{2p-1}{2}(\mathbf{u}-\mathbf{u}_0)^{\top}\mathbf{J}(\mathbf{u}-\mathbf{u}_0) + \sum_{i=0}^{k} \frac{i+1}{2}\nabla G_p(\mathbf{M}(\mathbf{u}_i))^{\top}\mathbf{u}$
    $\mathbf{u}_{k+1} \leftarrow \frac{2}{k+3}\mathbf{w}_k + \frac{k+1}{k+3}\mathbf{v}_k$
end for
```

Fig. 2. Smooth approximation algorithm.

with Lipschitz-continuous gradient. This is the technique we consider in the following. Recall the general form of the discrepancy minimization SDP in the feature space:

$$\begin{aligned} &\text{minimize} \quad \|\mathbf{M}(\mathbf{z})\|_2 \\ &\text{subject to} \quad \mathbf{M}(\mathbf{z}) = \sum_{i=0}^{\mathfrak{m}} z_i \mathbf{M}_i \wedge z_0 = -1 \wedge \sum_{i=1}^{\mathfrak{m}} z_i = 1 \wedge \forall i \in [1, \mathfrak{m}], z_i \geq 0, \end{aligned} \tag{15}$$

where $\mathbf{z} \in \mathbb{R}^{\mathfrak{m}+1}$ and where the matrices $\mathbf{M}_i \in \mathbb{S}_+^N$, $i \in [0, \mathfrak{m}]$, are fixed SPSD matrices. Thus, here $C = \{\mathbf{z} \in \mathbb{R}^{\mathfrak{m}+1} \colon z_0 = -1 \wedge \sum_{i=1}^{\mathfrak{m}} z_i = 1 \wedge \forall i \in [1, \mathfrak{m}], z_i \geq 0\}$. We further assume in the following that the matrices $\mathbf{M}_i$ are linearly independent since the problem can be reduced to that case straightforwardly. The symmetric matrix $\mathbf{J} = [\langle \mathbf{M}_i, \mathbf{M}_j \rangle_F]_{i,j} \in \mathbb{R}^{(\mathfrak{m}+1)\times(\mathfrak{m}+1)}$ is then PDS and we will be using the norm $\mathbf{x} \mapsto \sqrt{\langle \mathbf{J}\mathbf{x}, \mathbf{x}\rangle} = \|\mathbf{x}\|_{\mathbf{J}}$ on $\mathbb{R}^{\mathfrak{m}+1}$.

A difficulty in solving this SDP is that the function $F \colon \mathbf{z} \mapsto \|\mathbf{M}(\mathbf{z})\|_2$ is not differentiable since eigenvalues are not differentiable functions at points where they coalesce, which, by the nature of the minimization, is likely to be the case precisely at the optimum. Instead, we can seek a smooth approximation of that function. One natural candidate is the function $\mathbf{z} \mapsto \|\mathbf{M}(\mathbf{z})\|_F^2$. However, the Frobenius norm can lead to a very coarse approximation of the spectral norm. As suggested by Nesterov [15], the function $G_p \colon \mathbf{M} \mapsto \frac{1}{2}\operatorname{Tr}[\mathbf{M}^{2p}]^{\frac{1}{p}}$, where $p \geq 1$ is an integer, can be used to give a smooth approximation. Indeed, let $\lambda_1(\mathbf{M}) \geq \lambda_2(\mathbf{M}) \geq \cdots \geq \lambda_N(\mathbf{M})$ denote the list of the eigenvalues of a matrix $\mathbf{M} \in \mathbb{S}^N$ in decreasing order. By the definition of the trace, for all $\mathbf{M} \in \mathbb{S}^N$, $G_p(\mathbf{M}) = \frac{1}{2}\big[\sum_{i=1}^{N} \lambda_i^{2p}(\mathbf{M})\big]^{\frac{1}{p}}$, thus

$$\frac{1}{2}\lambda^2 \leq G_p(\mathbf{M}) \leq \frac{1}{2}(\operatorname{rank}(\mathbf{M})\lambda^{2p})^{\frac{1}{p}},$$

where $\lambda = \max\{\lambda_1(\mathbf{M}), -\lambda_N(\mathbf{M})\} = \|\mathbf{M}\|_2$. Thus, if we choose r as the maximum rank, $r = \max_{\mathbf{z}\in C} \operatorname{rank}(\mathbf{M}(\mathbf{z})) \leq \max\{N, \sum_{i=0}^{n} \operatorname{rank}(\mathbf{M}_i)\}$, then for all $\mathbf{z} \in C$,

$$\frac{1}{2}\|\mathbf{M}(\mathbf{z})\|_2^2 \leq G_p(\mathbf{M}(\mathbf{z})) \leq \frac{1}{2} r^{\frac{1}{p}} \|\mathbf{M}(\mathbf{z})\|_2^2. \tag{16}$$

This leads to a smooth approximation algorithm for solving the SDP (15) derived from Algorithm 1 by replacing the objective function F with G_p. Choosing the prox-function $d \colon \mathbf{u} \mapsto \frac{1}{2}\|\mathbf{u} - \mathbf{u}_0\|_{\mathbf{J}}^2$ leads to the algorithm whose pseudocode is given in Figure 2, after some minor simplifications. The following theorem guarantees that its maximum number of iterations to achieve a relative accuracy of ϵ is in $O(\sqrt{r \log r}/\epsilon)$.

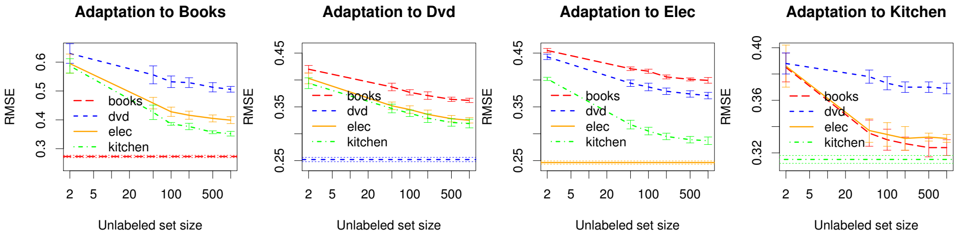

Fig. 3. Performance improvement of the RMSE for the 12 adaptation tasks as a function of the size of the unlabeled data used. Note that the figures do not make use of the same y-scale.

Theorem 5. *For any $\epsilon > 0$, Algorithm 2 solves the SDP* (15) *with relative accuracy ϵ in at most $4\sqrt{(1+\epsilon)r\log r/\epsilon}$ iterations using the objective function G_p with $p \in [q_0, 2q_0)$ and $q_0 = (1+\epsilon)(\log r)/\epsilon$.*

Proof. The proof follows directly [14], it is given in Appendix A for completeness. □

The first step of the algorithm consists of computing the vector $\mathbf{u}_0$ by solving the simple QP of line 1. We now discuss in detail how to efficiently compute the steps of each iteration of the algorithm in the case of our discrepancy minimization problems.

Each iteration of the algorithm requires solving two simple QPs (lines 3 and 4). To do so, the computation of the gradient $\nabla G_p(\mathbf{M}(\mathbf{u}_k))$ is needed. This will therefore represent the main computational cost at each iteration other than solving the QPs already mentioned since, clearly, the sum $\sum_{i=0}^{k} \frac{i+1}{2} \nabla G_p(\mathbf{M}(\mathbf{u}_i))^\top \mathbf{u}$ required at line 4 can be computed in constant time from its value at the previous iteration. Since for any $\mathbf{z} \in \mathbb{R}^{\mathfrak{m}}$

$$G_p(\mathbf{M}(\mathbf{z})) = \mathrm{Tr}[\mathbf{M}^{2p}(\mathbf{z})]^{1/p} = \mathrm{Tr}\left[(\sum_{i=0}^{\mathfrak{m}} z_i \mathbf{M}_i)^{2p}\right]^{1/p},$$

using the linearity of the trace operator, the ith coordinate of the gradient is given by

$$[\nabla G_p(\mathbf{M}(\mathbf{z}))]_i = \langle \mathbf{M}^{2p-1}(\mathbf{z}), \mathbf{M}_i \rangle_F \, \mathrm{Tr}[\mathbf{M}^{2p}(\mathbf{z})]^{\frac{1}{p}-1}, \tag{17}$$

for all $i \in [0, \mathfrak{m}]$. Thus, the computation of the gradient can be reduced to that of the matrices $\mathbf{M}^{2p-1}(\mathbf{z})$ and $\mathbf{M}^{2p}(\mathbf{z})$. When the dimension of the feature space N is not too large, both $\mathbf{M}^{2p-1}(\mathbf{z})$ and $\mathbf{M}^{2p}(\mathbf{z})$ can be computed via $O(\log p)$ matrix multiplications using the binary decomposition method to compute the powers of a matrix [5]. Since each matrix multiplication takes $O(N^3)$, the total computational cost for determining the gradient is then in $O((\log p)N^3)$. The cubic-time matrix multiplication can be replaced by more favorable complexity terms of the form $O(N^{2+\alpha})$, with $\alpha = .376$. Alternatively, for large values of N, that is $N \gg (\mathfrak{m} + \mathfrak{n})$, in view of Theorem 3, we can instead solve the kernelized version of the problem. Since it is formulated as the same SDP, the same smooth optimization technique can be applied. Instead of $\mathbf{M}(\mathbf{z})$, we need to consider the matrix $\mathbf{M}'(\mathbf{z}) = \mathbf{K}^{1/2}\mathbf{D}(\mathbf{z})\mathbf{K}^{1/2}$. Now, observe that

$$\mathbf{M}'^{2p}(\mathbf{z}) = \left[\mathbf{K}^{1/2}\mathbf{D}(\mathbf{z})\mathbf{K}^{1/2}\right]^{2p} = \mathbf{K}^{1/2}\left[\mathbf{D}(\mathbf{z})\mathbf{K}\right]^{2p-1}\mathbf{D}(\mathbf{z})\mathbf{K}^{1/2}.$$

Table 1. RMSE results obtained for the 12 adaptation tasks. Each field of the table has three results: from training only on the source data (top), from the adaptation task (middle), and from training only on the target data (bottom).

	books	dvd	elec	kitchen
books	 $.273 \pm .004$ 	$.450 \pm .005$ $.362 \pm .004$ $.252 \pm .004$	$.544 \pm .002$ $.407 \pm .009$ $.246 \pm .003$	$.331 \pm .001$ $.324 \pm .006$ $.315 \pm .003$
dvd	$.546 \pm .007$ $.506 \pm .010$ $.273 \pm .004$	 $.252 \pm .004$ 	$.505 \pm .004$ $.371 \pm .006$ $.246 \pm .003$	$.383 \pm .003$ $.369 \pm .004$ $.315 \pm .003$
elec	$.412 \pm .005$ $.399 \pm .012$ $.273 \pm .004$	$.429 \pm .006$ $.325 \pm .005$ $.252 \pm .004$	 $.246 \pm .003$ 	$.345 \pm .004$ $.331 \pm .003$ $.315 \pm .003$
kitchen	$.360 \pm .003$ $.352 \pm .008$ $.273 \pm .004$	$.412 \pm .002$ $.319 \pm .008$ $.252 \pm .004$	$.330 \pm .003$ $.287 \pm .007$ $.246 \pm .003$	 $.315 \pm .003$

Thus, by the property of the trace operator,

$$\mathrm{Tr}[\mathbf{M}'^{2p}(\mathbf{z})] = \mathrm{Tr}[\mathbf{D}(\mathbf{z})\mathbf{K}^{1/2}\mathbf{K}^{1/2}[\mathbf{D}(\mathbf{z})\mathbf{K}]^{2p-1}] = \mathrm{Tr}[[\mathbf{D}(\mathbf{z})\mathbf{K}]^{2p}]. \tag{18}$$

The other term appearing in the expression of the gradient can be computed as follows:

$$\begin{aligned}\langle \mathbf{M}'^{2p-1}(\mathbf{z}), \mathbf{M}'_i \rangle_F &= \mathrm{Tr}[[\mathbf{K}^{1/2}\mathbf{D}(\mathbf{z})\mathbf{K}^{1/2}]^{2p-1}\mathbf{K}^{1/2}\mathbf{D}_i\mathbf{K}^{1/2}] \\ &= \mathrm{Tr}[\mathbf{K}^{1/2}\left[\mathbf{D}(\mathbf{z})\mathbf{K}\right]^{2p-2}\mathbf{D}(\mathbf{z})\mathbf{K}^{1/2}\mathbf{K}^{1/2}\mathbf{D}_i\mathbf{K}^{1/2}] \\ &= \mathrm{Tr}[\mathbf{K}[\mathbf{D}(\mathbf{z})\mathbf{K}]^{2p-1}\mathbf{D}_i],\end{aligned}$$

for any $i \in [1, \mathfrak{m}]$. Observe that multiplying a matrix $\mathbf{A}$ by $\mathbf{D}_i$ is equivalent to zeroing all of its columns but the ith one, therefore $\mathrm{Tr}[\mathbf{A}\mathbf{D}_i] = \mathbf{A}_{ii}$. In view of that,

$$\langle \mathbf{M}'^{2p-1}(\mathbf{z}), \mathbf{M}'_i \rangle_F = [\mathbf{K}[\mathbf{D}(\mathbf{z})\mathbf{K}]^{2p-1}]_{ii}. \tag{19}$$

Therefore, the diagonal of the matrix $\mathbf{K}[\mathbf{D}(\mathbf{z})\mathbf{K}]^{2p-1}$ provides all these terms. Thus, in view of (18) and (19), the gradient given by (17) can be computed directly from the $(2p)$th and $(2p-1)$th powers of the matrix $\mathbf{D}(\mathbf{z})\mathbf{K}$. The iterated powers of this matrix, $[\mathbf{D}(\mathbf{z})\mathbf{K}]^{2p}(\mathbf{z})$ and $[\mathbf{D}(\mathbf{z})\mathbf{K}]^{2p-1}(\mathbf{z})$, can be both computed using a binary decomposition in time $O((\log p)(\mathfrak{m}+\mathfrak{n})^3)$. This is a significantly more efficient computational cost per iteration for $N \gg (\mathfrak{m} + \mathfrak{n})$. It is also substantially more favorable than the iteration cost for solving the SDP using interior-point methods $O(\mathfrak{m}^3 + \mathfrak{m}N^3 + \mathfrak{m}^2N^2 + \mathfrak{n}N^2)$. Furthermore, the space complexity of the algorithm is only in $O((\mathfrak{m} + \mathfrak{n})^2)$.

6 Experiments

This section reports the results of extensive experiments demonstrating both the effectiveness of discrepancy minimization in adaptation when using kernel ridge regression and the efficiency of our optimization algorithm. Our results show that the adaptation algorithm presented is practical even for relatively large data sets and for high-dimensional feature spaces.

For our experiments, we used the multi-domain sentiment dataset (version 1.0) of Blitzer et al. [3]. This data set has been used in several publications [4, 11], but despite the ordinal nature of the star labeling of the data, it has always been treated as a classification task, and not as a regression task which is the focus of this work. We are not aware of any other adaptation datasets that can be applied to the regression task.

To make the data conform with the regression setting discussed in the previous sections, we first convert the discrete labels to regression values by fitting all the data for each of the four tasks $\texttt{books}$, $\texttt{dvd}$, $\texttt{elec}$, and $\texttt{kitchen}$ a Gaussian kernel ridge regression with a relatively small width $\sigma = 1$ as feature vectors the normalized counts of the top 5,000 unigrams and bigrams, as measured across all four tasks. These regression values are used as target values for all subsequent modeling.

We then define 12 adaptation problems for each pair of distinct tasks $(\texttt{task}, \texttt{task}')$, where $\texttt{task}$ and $\texttt{task}'$ are in $\{\texttt{books}, \texttt{dvd}, \texttt{elec}, \texttt{kitchen}\}$. For each of these problems, the source empirical distribution is a mixture defined by 500 labeled points from $\texttt{task}$ and 200 from $\texttt{task}'$. This is intended to make the source and target distributions reasonably close, a condition for the theory developed in this paper, but the algorithm receives of course no information about this definition of the source distribution. The target distribution is defined by another set of points all from $\texttt{task}'$.

Figure 3 shows the performance of the algorithm on the 12 adaptation tasks between distinct domains plotted as a function of the amount of unlabeled data received from the target domain. The optimal performance obtained by training purely on the same amount of labeled data from the target domain is also indicated in each case. The input features are again the normalized counts of the top 5,000 unigrams and bigrams, as measured across all four tasks, and for modeling we use kernel ridge regression with the Gaussian kernel of the same width $\sigma = 1$. This setup guarantees that the target labeling function is in the hypothesis space, a condition matching one of the settings analyzed in our theoretical study. The results are mean values obtained from 9-fold cross validation and we plot mean values ± one standard deviation. As can be seen from the figure, adaptation improves, as expected, with increasing amounts of data. One can also observe that not all data sets are equally beneficial for adaptation. The $\texttt{kitchen}$ task primarily discusses electronic gadgets for kitchen use, and hence the $\texttt{kitchen}$ and $\texttt{elec}$ data sets adapt well to each other, an observation also made by Blitzer et al. [3].

Our results on the adaptation tasks are also summarized in Table 1. The row name indicates the source domain and the column name the target domain. Due to lack of space, we only list the results for adaptation with 1,000 unlabeled points from the target domain. In this table, we also provide for reference the results from training purely with labeled data from the source or target domain. We are not aware of any other adaptation algorithms for the regression tasks with which we can compare our performance results.

Algorithm 2 requires solving several QPs to compute $\mathbf{u}_0$ and $\mathbf{u}_{k+1}$, $k \geq 0$. Since $\mathbf{u}_{k+1} \in \mathbb{R}^{\mathtt{m}+1}$, the cost of these computations only depends on the size of the labeled sample $\mathtt{m}$, which is relatively small. Figure 4 displays average run times obtained for $\mathtt{m}$ in the range 500 to $10,000$. All experiments were carried out on a single processor of an Intel Zeon 2.67GHz CPU with 12GB of memory. The algorithm was implemented in R and made use of the $\texttt{quadprog}$ optimization package. As can be seen from the figure, the run times scale cubically in the sample size, reaching roughly $10s$ for $\mathtt{m} = 1,000$.

The dominant cost of each iteration of Algorithm 2 is the computation of the gradient $\nabla G_p(\mathbf{M}(\mathbf{u}_k))$, as already pointed out in Section 5. The iterated power method

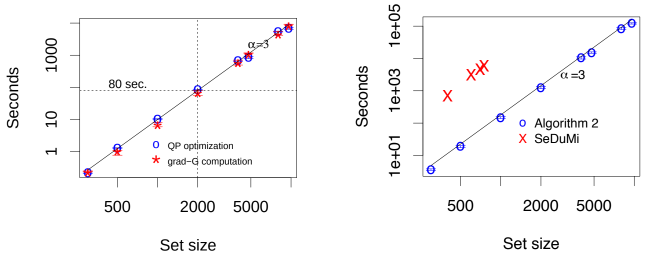

Fig. 4. The left panel shows a plot reporting run times measured empirically (mean ± one standard deviation) for the QP optimization and the computation of ∇G_p as a function of the sample size (log-log scale). The right panel compares the total time taken by Algorithm 2 to compute the optimization solution, to the one taken by SeDuMi (log-log scale).

provides a cost per iteration of $O((\log p)(\mathfrak{m}+\mathfrak{n})^3)$, and thus depends on the combined size of the labeled and unlabeled data. Figure 4 shows typical timing results obtained for different samples sizes in the range $\mathfrak{m}+\mathfrak{n} = 500$ to $\mathfrak{m}+\mathfrak{n} = 10,000$ for $p = 16$, which empirically was observed to guarantee convergence. For a sample size of $\mathfrak{m}+\mathfrak{n} = 2,000$ the time is about 80 seconds. With 5 iterations of Algorithm 2 the total time is $5 \times (80 + 2 * 10) + 10 = 510$ seconds.

In contrast, even the most efficient SDP solvers publicly available, SeDuMi, cannot solve our discrepancy minimization SDPs for more than a few hundred points in the kernelized version. In our experiments, SeDuMi (`http://sedumi.ie.lehigh.edu/`) simply failed for set sizes larger than $\mathfrak{m}+\mathfrak{n} = 750$! In Figure 4, typical run times for Algorithm 2 with 5 iterations are compared to run times using SeDuMi.

7 Conclusion

We presented several theoretical guarantees for domain adaptation in regression and proved that the empirical discrepancy minimization can also be cast as an SDP when using kernels. We gave an efficient algorithm for solving that SDP using results from smooth optimization and specific characteristics of these SDPs in our adaptation case. Our adaptation algorithm is shown to scale to larger data sets than what could be afforded using the best existing software for solving such SDPs. Altogether, our results form a complete solution for domain adaptation in regression, including theoretical guarantees, an efficient algorithmic solution, and extensive empirical results.

Acknowledgments

We thank Steve Boyd, Michael Overton, and Katya Scheinberg for discussions about the optimization problem addressed in this work.

Bibliography

[1] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira. Analysis of representations for domain adaptation. *NIPS 2006*, 2007.

[2] S. Ben-David, T. Lu, T. Luu, and D. Pál. Impossibility theorems for domain adaptation. *Journal of Machine Learning Research - Proceedings Track*, 9:129–136, 2010.

[3] J. Blitzer, M. Dredze, and F. Pereira. Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification. In *ACL 2007*, 2007.

[4] J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Wortman. Learning bounds for domain adaptation. *NIPS 2007*, 2008.

[5] T. Cormen, C. Leiserson, and R. Rivest. *Introduction to Algorithms*. The MIT Press, 1992.

[6] C. Cortes and V. Vapnik. Support-Vector Networks. *Machine Learning*, 20(3), 1995.

[7] M. Dredze, J. Blitzer, P. P. Talukdar, K. Ganchev, J. Graca, and F. Pereira. Frustratingly Hard Domain Adaptation for Parsing. In *CoNLL 2007*, 2007.

[8] R. M. Dudley. *Real Analysis and Probability*. Wadsworth, Belmont, CA, 1989.

[9] J. Jiang and C. Zhai. Instance Weighting for Domain Adaptation in NLP. In *Proceedings of ACL 2007*, pages 264–271, 2007.

[10] C. J. Legetter and P. C. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models. *Comp. Speech and Lang.*, 1995.

[11] Y. Mansour, M. Mohri, and A. Rostamizadeh. Domain adaptation: Learning bounds and algorithms. In *Proceedings of COLT 2009*, Montréal, Canada, 2009. Omnipress.

[12] A. M. Martínez. Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class. *IEEE Trans. Pattern Anal.*, 24(6), 2002.

[13] Y. Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

[14] Y. Nesterov. Smooth minimization of non-smooth functions. *Math. Program.*, 103:127–152, May 2005.

[15] Y. Nesterov. Smoothing technique and its applications in semidefinite optimization. *Math. Program.*, 110:245–259, 2007.

[16] Y. Nesterov and A. Nemirovsky. *Interior Point Polynomial Methods in Convex Programming: Theory and Appl.* SIAM, 1994.

[17] S. D. Pietra, V. D. Pietra, R. L. Mercer, and S. Roukos. Adaptive language modeling using minimum discriminant estimation. In *HLT '91: workshop on Speech and Nat. Lang.*, 1992.

[18] R. Rosenfeld. A Maximum Entropy Approach to Adaptive Statistical Language Modeling. *Computer Speech and Language*, 10:187–228, 1996.

[19] C. Saunders, A. Gammerman, and V. Vovk. Ridge Regression Learning Algorithm in Dual Variables. In *ICML*, 1998.

[20] I. Steinwart. On the influence of the kernel on the consistency of support vector machines. *JMLR*, 2:67–93, 2002.

[21] V. N. Vapnik. *Statistical Learning Theory*. J. Wiley & Sons, 1998.

A Proof of Theorem 5

Proof. Let $\|\mathbf{M}^*\|_2$ be the optimum of the SDP (15), $G_p(\mathbf{M}'^*)$ that of the SDP with F replaced with its smooth approximation G_p, and $\mathbf{z}^* \in C$ a solution of that SDP with relative accuracy ϵ. Then, for $p \geq \frac{(1+\epsilon)\log r}{\epsilon}$, in view of (16), $\mathbf{z}^*$ is a solution of the original SDP (15) with relative accuracy ϵ:

$$\frac{\|\mathbf{M}(z^*)\|_2}{\|\mathbf{M}^*\|_2} \leq r^{\frac{1}{2p}} \frac{\sqrt{G_P(\mathbf{M}(\mathbf{z}^*))}}{\sqrt{G_p(\mathbf{M}'^*)}} \leq r^{\frac{1}{2p}}(1+\epsilon)^{1/2} \leq (1+\epsilon).$$

G_p can be shown to admit a Lipschitz gradient with Lipschitz constant $L = (2p - 1)$ with respect to the norm $\|\cdot\|_{\mathbf{J}}$ and the prox-function d can be chosen as $d(\mathbf{u}) = \frac{1}{2}\|\mathbf{u} - \mathbf{u}_0\|_{\mathbf{J}}^2$, with $\mathbf{u}_0 = \operatorname{argmin}_{\mathbf{u}\in C} \|\mathbf{u}\|_{\mathbf{J}}$ and convexity parameter $\sigma = 1$. It can be shown that $d(\mathbf{z}^*) \leq rG_p(\mathbf{M}'^*)$. Thus, in view of Theorem 4, $\frac{G_p(\mathbf{M}(z_k)) - G_p(\mathbf{M}'^*)}{G_p(\mathbf{M}'^*)} \leq \frac{4(2p-1)r}{(k+1)(k+2)}$. Choosing p such that $2p < 4\frac{(1+\epsilon)\log r}{\epsilon}$, and setting the right-hand side to $\epsilon > 0$, gives the following maximum number of iterations to achieve a relative accuracy of ϵ using Algorithm 2: $k^* = \sqrt{(16r(1+\epsilon)\log r)/\epsilon^2} = 4\sqrt{(1+\epsilon)r\log r}/\epsilon$. □