Domain Adaptation and Sample Bias Correction Theory and Algorithm for Regression

Corinna Cortes $^{\rm a}$ and Mehryar Mohri $^{\rm b,a}$

^aGoogle Research, 76 Ninth Avenue, New York, NY 10011 ^bCourant Institute of Mathematical Sciences, 251 Mercer Street, New York, NY 10012.

Abstract

We present a series of new theoretical, algorithmic, and empirical results for domain adaptation and sample bias correction in regression. 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 also give finer bounds based on the discrepancy and a weighted feature discrepancy parameter. 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 and large samples. We have implemented this algorithm and report the results of experiments both with artificial and real-world data sets demonstrating its benefits both for general scenario of adaptation and the more specific scenario of sample bias correction. Our results show that it can scale to large data sets of tens of thousands or more points and demonstrate its performance improvement benefits.

Key words: machine learning, learning theory, domain adaptation, optimization.

Email addresses: corinna@google.com (Corinna Cortes), mohri@cims.nyu.edu (Mehryar Mohri).

1 Introduction

A standard assumption in learning theory and the design of learning algorithms is that training and test points are drawn according to the same distribution. In practice, however, this assumption often does not hold. A more challenging problem of *domain adaptation* arises in a variety of applications, including natural language processing, speech processing, or computer vision [12, 4, 17, 18, 29, 30, 21]. 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. The first theoretical analysis of adaptation is due to Ben-David et al. [1] (some technical issues of that paper were later corrected by Blitzer et al. [5]). 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. The assumptions made in the analysis of adaptation were more recently discussed by Ben-David et al. [2] who presented some negative results for adaptation in the case of the zero-one loss based on a handful of examples.

In previous work [20], we introduced the notion of discrepancy which generalizes the d_A distance to arbitrary loss functions. We gave data-dependent Rademacher complexity bounds showing how the discrepancy can be estimated from finite samples. We then presented alternative learning bounds for adaptation based on the discrepancy. These bounds hold for a general class of loss functions, including the zero-one loss function used in classification, and depend on the optimal classifiers in the hypothesis set for the source and target distributions. They are in general not comparable to those of Ben-David et al. [1] or Blitzer et al. [5], but we showed that, under some plausible assumptions, they are superior to those of [1, 5] and that in many cases the bounds of Ben-David et al. [1] or Blitzer et al. [5] have a factor of 3 of the error that can make them vacuous. Perhaps more importantly, we also gave a series of pointwise loss guarantees for the broad class of kernel-based regularization algorithms in terms of the empirical discrepancy. These bounds motivated a discrepancy minimization algorithm and we initiated the study of its properties.

Many of the previous techniques or paradigms used for adaptation and similar problems consist of reweighting the training point losses to more closely reflect those in the test distribution. The definition of the reweighting is of course crucial and varies for different techniques. A common choice consists of selecting the weight of a point x of the training sample as an estimate of the ratio $\omega(x) = P(x)/Q(x)$ where P is the target (unbiased) distribution and Q the observed source distribution, since this choice preserves the expected loss [32, 10]. However, we gave an empirical and theoretical analysis of importance weighting [11] which shows that, even when using the exact ratio P/Q, such importance weighting techniques do not succeed in general, even in the simple case of two Gaussians. A critical issue we pointed out is that the weight $\omega(x)$ is unbounded in many practical cases and can become very large for a few points x of the sample that end up fully dominating the learning process, thereby resulting in a very poor performance. We also presented an analysis of the effect of an error in the estimation of the reweighting parameters on the accuracy of the hypothesis returned by the learning algorithm in [10].

Bickel et al. [3] developed a discriminative model that instead characterizes how much more likely an instance is to occur in the test sample than in the training sample. The optimization solution they describe is in general not convex but they prove it to be in the case of the exponential loss. Their method is proposed for a classification setting but can also be applied to the regression setting in a twostage approximation. Weights are first learned by maximizing the posterior probability given all the available data. These weights can then be used in combination with any algorithm that allows for weighted examples. A somewhat different kernel mean matching (KMM) method was described by Huang et al. [16] in the context of kernel methods. This consists of defining the weights assigned to the training sample in a way such that the mean feature vector on the training points be as close as possible to the mean feature vector over the unlabeled points. Yu and Szepesvári recently presented an analysis of the KMM estimator [37]. This should be distinguished from an analysis of the generalization properties of KMM as an algorithm for adaptation or sample bias correction. Sugiyama et al. [34] argued that KMM does not admit a principled cross-validation method helping to select the kernel parameters and proposed instead an algorithm (KLIEP) addressing that issue. Their algorithm determines the weights based on the minimization of the relative entropy of $\omega(x)Q(x)$ and the distribution of unlabeled data over the input domain, or equivalently, the maximization of the log-likelihood of $\omega(x)Q(x)$ for the observed unlabeled data. The weights $\omega(x)$ are modeled, more specifically, as a linear combination of kernel basis functions such as Gaussians. The algorithms just mentioned do not take into account the hypothesis set used by the learning algorithm, or the loss function relevant to the problem. In contrast, we will introduce and analyze an algorithm that precisely takes both of these into consideration.

In this paper, we present a series of novel results for domain adaptation extending those of [20] and making them more significant and practically applicable. ¹ Our analysis concentrates on the problem of adaptation in regression. We also consider the problem of sample bias correction, which can be viewed as as special instance of adaptation.

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 new theoretical results in Section 3. For the squared loss, we prove that the discrepancy is a distance when the hypothesis set is the repro-

¹ This is an extended version of the conference paper [8] including more details and additional theoretical and empirical results.

ducing 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 zero-one 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 [20] 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. We also present finer bounds in the specific case of the squared loss L_2 in terms of the discrepancy and a weighted feature discrepancy parameter that we define and analyze in detail.

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 adaptation algorithm based on empirical discrepancy minimization [20] detailed in Section 4. In Section 5, we extend the discrepancy minimization 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 [20] 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 ones. In Section 6, 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 [25]. 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 7 reports our empirical results with both artificial and real-world data sets demonstrating the effectiveness of this algorithm. We are presenting two sets of experiments: one for the general scenario of domain adaptation, also meant to evaluate the computational efficiency of our optimization solution, another one for the specific scenario of sample bias correction. For sample bias correction in a regression setting, we compare our algorithm with KMM, KLIEP, and the two-stage algorithm of Bickel et al. [3] and report empirical results demonstrating the benefits of our algorithm.

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 $\mathcal X$ denote the input space and $\mathcal Y$ the output space, a measurable subset of $\mathbb R$, as in standard regression problems. In the general adaptation problem we are considering, there are different *domains*, each defined as a pair formed by a distribution over $\mathcal X$ and a target labeling function mapping from $\mathcal X$ to $\mathcal Y$. We denote by (Q,f_Q) the *source domain*, with Q the corresponding distribution over $\mathcal X$ and $f_Q\colon \mathcal X\to \mathcal Y$ the corresponding labeling function. Similarly, we denote by (P,f_P) the *target domain* with P the corresponding distribution over $\mathcal X$ and f_P the corresponding target labeling function.

In the domain adaptation problem, the learning algorithm receives a labeled sample of m points $\mathcal{S} = ((x_1, y_1), \dots, (x_m, y_m)) \in (\mathcal{X} \times \mathcal{Y})^m$ from the source domain, that is x_1, \dots, 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, \dots, 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, with n typically substantially larger than m. We denote by \widehat{P} the empirical distribution corresponding to \mathcal{T} .

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

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

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, $\mathcal{L}_P(h, f_P)$. The sample bias correction problem can be viewed as a special instance of the adaptation problem where the labeling functions f_P and f_Q coincide and where the support of Q is included in that of P: $\operatorname{supp}(Q) \subseteq \operatorname{supp}(P) \subseteq \mathcal{X}$.

2.2 Discrepancy

A key question for adaptation is a measure of the difference between the distributions Q and P. As pointed out in [20], 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 measure of the difference 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 [20].

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

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

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 will prove, however, that for a large family of kernel-based hypothesis set, it does verify all the axioms of a distance.

Note that for a loss function bounded by M, the discrepancy $\operatorname{disc}(P,Q)$ can be upper bounded in terms of the L_1 distance $L_1(P,Q)$. Indeed, if P and Q are absolutely continuous with density functions p and q, then,

$$\operatorname{disc}(P,Q) = \max_{h,h'\in H} \left| \int_{\mathcal{X}} (p(x) - q(x)) L(h'(x), h(x)) dx \right|$$

$$\leq \max_{h,h'\in H} \int_{\mathcal{X}} \left| (p(x) - q(x)) L(h'(x), h(x)) \right| dx$$

$$\leq M \int_{\mathcal{X}} |p(x) - q(x)| dx = M L_1(P, Q).$$
(3)

This shows that the discrepancy is a finer measure than the L_1 distance. Another important advantage of the discrepancy is that it can be accurately estimated from a finite sample of size m drawn from Q and a finite sample of size n drawn from P when the loss is bounded and the empirical Rademacher complexity of the family of functions $L_H = \{x \mapsto L(h'(x), h(x)) : h, h' \in H\}$ is in $O(1/\sqrt{m})$ for a sample of size m, which holds in particular when L_H has a finite pseudo-dimension [20].

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 $x\in\mathcal X$. By the reproducing property, for any $h\in H$ and $x\in\mathcal X$, $h(x)=\langle h,K(x,\cdot)\rangle_K$, 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 and the squared loss the discrepancy defines a distance. Let $C(\mathcal{X})$ denote the set of all continuous functions mapping \mathcal{X} to \mathbb{R} . We shall assume that \mathcal{X} is a compact set, thus the functions in $C(\mathcal{X})$ are also bounded. A PDS kernel K over $\mathcal{X} \times \mathcal{X}$ is said to be universal if it is continuous and if the RKHS \mathbb{H} it induces is dense in $C(\mathcal{X})$ for the norm infinity $\|\cdot\|_{\infty}$. Universal kernels include familiar kernels such as Gaussian kernels [33], perhaps one of the most widely used kernels in applications: for any fixed $\sigma > 0$, the kernel function K defined by

$$\forall \mathbf{x}, \mathbf{x}' \in \mathcal{X}, \quad K(\mathbf{x}, \mathbf{x}') = \exp\left(\frac{\|\mathbf{x}' - \mathbf{x}\|_2^2}{\sigma^2}\right).$$

For any fixed $\alpha > 0$, the so-called *infinite polynomial kernel* defined for any compact set $\mathcal{X} \subset \{\mathbf{x} \in \mathbb{R}^N \colon ||\mathbf{x}||_2 < 1\}$ by

$$\forall \mathbf{x}, \mathbf{x}' \in \mathcal{X}, \quad K(\mathbf{x}, \mathbf{x}') = \frac{1}{(1 - \mathbf{x} \cdot \mathbf{x}')^{\alpha}}$$

is also universal. See [33] for many other examples of universal kernels.

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

$$\begin{aligned} | \mathop{\mathbf{E}}_{x \sim P}[h'^{2}(x)] - \mathop{\mathbf{E}}_{x \sim P}[h^{2}(x)] | &= | \mathop{\mathbf{E}}_{x \sim P}[(h'+h)(x)(h'-h)(x)] | \\ &\leq (\|h\|_{\infty} + \|h'\|_{\infty}) \|h'-h\|_{\infty}, \end{aligned}$$

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

$$\forall h, h' \in H, \quad \left| \underset{x \sim P}{\mathbb{E}} [(h'(x) - h(x))^2] - \underset{x \sim Q}{\mathbb{E}} [(h'(x) - h(x))^2] \right| = 0.$$

Thus, $\Psi(h'')=\mathrm{E}_{x\sim P}[h''^2(x)]-\mathrm{E}_{x\sim Q}[h''^2(x)]=0$ for any $h''=h'-h\in\mathbb{H}$ with $\|h''\|_K\leq 2\Lambda r$. Therefore, we have $\Psi(h'')=0$ for any $h''\in\mathbb{H}$ with $\|h''\|_K\leq 2\Lambda r$. Hence, the equality $\Psi(h'')=0$ holds for any $h''\in\mathbb{H}$ regardless of the value of $\|h''\|_K$. Thus, $\Psi=0$ over \mathbb{H} . Since K is universal, \mathbb{H} is dense in $C(\mathcal{X})$ for the norm $\|\cdot\|_\infty$ and by continuity of Ψ for $\|\cdot\|_\infty$, for all $h\in C(\mathcal{X})$, $\mathrm{E}_P[h^2]-\mathrm{E}_Q[h^2]=0$. Let

f be any non-negative function in $C(\mathcal{X})$, then \sqrt{f} is well defined and is in $C(\mathcal{X})$, thus,

$$\underset{P}{\mathbf{E}}\left[\left(\sqrt{f}\right)^{2}\right] - \underset{Q}{\mathbf{E}}\left[\left(\sqrt{f}\right)^{2}\right] = \underset{P}{\mathbf{E}}[f] - \underset{Q}{\mathbf{E}}[f] = 0.$$

It is known that if $E_P[f] - E_Q[f] = 0$ for all $f \in C(\mathcal{X})$ with $f \geq 0$, then P = Q (see [14][proof of lemma 9.3.2]). This concludes the proof.

The proof of the theorem can be straightforwardly extended to the case of other L_p losses with p>1. In later sections, we will present several guarantees based on the notion of discrepancy and define an algorithm seeking a minimal discrepancy. The theorem shows that if we could find a source distribution Q that would reduce to zero the discrepancy in the case of a universal kernel such as the familiar Gaussian kernels, then that distribution would in fact match the target distribution P. In the absence of that guarantee, even with a zero discrepancy in general adaptation may not be successful (see [2] for simple examples of this situation).

3.2 General 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 over \mathbb{H} of an objective function of the following form:

$$F_{(\widehat{Q},f_O)}(h) = \widehat{R}_{(\widehat{Q},f_O)}(h) + \lambda ||h||_K^2, \tag{4}$$

where $\lambda > 0$ is a regularization parameter and $\widehat{R}_{(\widehat{Q},f_Q)}(h) = \frac{1}{m} \sum_{i=1}^m L(h(x_i),y_i)$ the empirical error of hypothesis $h \in \mathbb{H}$, with $y_i = f_Q(x_i)$ for all $i \in [1,m]$. This family of algorithms includes support vector machines (SVMs) [9], support vector regression (SVR) [36], kernel ridge regression (KRR) [31], and many other algorithms.

We will assume that the loss function L is μ -admissible for some $\mu > 0$: that is, it is convex with respect to its first argument and for all $x \in \mathcal{X}$ and $y, y' \in \mathcal{Y}$ and $h, h' \in H$, it verifies the following Lipschitz-type conditions:

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

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

[Note that this is a weaker requirement than the μ -Lipschitzness of the loss L with respect to its two arguments. 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. Indeed, assume that $L_q(h(x), y) \leq M$ for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$ and $h \in H$. Since the function $x \mapsto x^q$ is q-Lipschitz for $x \in [0, 1]$,

we can write

$$|L_{q}(h'(x), y) - L_{q}(h(x), y)| = M^{q} |(|h'(x) - y|/M)^{q} - (|h(x) - y|/M)^{q}|$$

$$\leq q M^{q} (|h'(x) - y|/M) - (|h(x) - y|/M)|$$

$$= q M^{q-1} ||h'(x) - y| - |h(x) - y||$$

$$\leq q M^{q-1} ||h'(x) - h(x)|.$$

The Lipschitz property with respect to the second argument can be shown in a similar way.

Here, in the case of the kernel-based regularized algorithms just described, we set the parameter Λ defining the hypothesis set H to $\Lambda = \sqrt{\frac{\mu r}{\lambda}}$, that is $H = \{h \in \mathbb{H} \colon \|h\|_K \leq \sqrt{\frac{\mu r}{\lambda}}\}$. This does not impose any additional constraint to the minimization (4) as shown by the following lemma.

Lemma 1 Let $h \in \mathbb{H}$ a be a solution of the minimization (4) for some training sample S, then h satisfies the inequality $||h||_K \leq \sqrt{\frac{\mu r}{\lambda}}$.

Proof. Since 0 is an element of \mathbb{H} , the value of the objective function for the minimizer h is upper bounded by the one for 0:

$$\frac{1}{m} \sum_{i=1}^{m} L(h(x_i), y_i) + \lambda ||h||_K^2 \le \frac{1}{m} \sum_{i=1}^{m} L(0, y_i).$$
 (5)

By the μ -admissibility of the loss, we can then write

$$\lambda \|h\|_{K}^{2} \leq \frac{1}{m} \sum_{i=1}^{m} L(0, y_{i}) - \frac{1}{m} \sum_{i=1}^{m} L(h(x_{i}), y_{i})$$

$$\leq \frac{\mu}{m} \sum_{i=1}^{m} |0 - h(x_{i})| \leq \frac{\mu}{m} \sum_{i=1}^{m} \|h\|_{K} K(x_{i}, x_{i}) \leq \mu r \|h\|_{K}.$$

Comparing the left- and right-hand side expressions shows that $||h||_K \leq \sqrt{\frac{\mu r}{\lambda}}$.

The labeling functions f_P and f_Q may not coincide on $\operatorname{supp}(\widehat{Q})$ or $\operatorname{supp}(\widehat{P})$. 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. In the following theorem, we measure the magnitude of the difference between the source and target labeling functions by the following coefficient $\eta_H(f_P, f_Q)$:

$$\eta_H(f_P, f_Q) = \inf_{h \in H} \Big\{ \max_{x \in \operatorname{supp}(\widehat{P})} |f_P(x) - h(x)| + \max_{x \in \operatorname{supp}(\widehat{Q})} |f_Q(x) - h(x)| \Big\}.$$
 (6)

Note that if the target function f_P is in H, then, by definition of $\eta_H(f_P, f_Q)$ we can

write

$$\eta_H(f_P, f_Q) \le \eta'_H(f_P, f_Q) = \max_{x \in \text{supp}(\widehat{Q})} |f_P(x) - f_Q(x)|.$$
(7)

Similarly, if f_Q is in H, then we have

$$\eta_H(f_P, f_Q) \le \eta_H''(f_P, f_Q) = \max_{x \in \text{supp}(\widehat{P})} |f_P(x) - f_Q(x)|.$$
(8)

The proof of the following theorem as well as that of other theorems presented in this section are given in the appendix.

Theorem 2 (General pointwise bound) Let L be a μ -admissible loss. Define h' as the hypothesis returned by the kernel-based regularization algorithm (4) when minimizing $F_{(\widehat{Q},f_Q)}$ and h the one returned when minimizing $F_{(\widehat{Q},f_Q)}$. Then, for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, the following inequality holds:

$$\left| L(h'(x), y) - L(h(x), y) \right| \le \mu r \sqrt{\frac{\operatorname{disc}(\widehat{P}, \widehat{Q}) + \mu \eta_H(f_P, f_Q)}{\lambda}}, \tag{9}$$

where $\eta_H(f_P, f_Q)$ is defined as in (6).

The 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 results holds for all μ -admissible losses and has a simpler form than a previous result we presented in [20].

The theorem shows the key role played by the empirical discrepancy $\operatorname{disc}(\widehat{P},\widehat{Q})$ in this context when $\eta_H(f_P,f_Q)\ll 1$. When the so-called *covariate-shift assumption* holds, the labeling functions f_P and f_Q coincide. This assumption holds in a variety of scenarios such as that of sample bias correction. Note that when that assumption holds, if f_P is in H (or f_Q is in H) then $\eta_H(f_P,f_Q)=\eta'_H(f_P,f_Q)=0$ (resp. $\eta_H(f_P,f_Q)=\eta'_H(f_P,f_Q)=0$), that is the bound of the theorem only depends on the empirical discrepancy. In the following section, we present a finer guarantee and analysis in the special case of the squared loss.

3.3 Guarantees for kernel ridge regression

In this section, we consider the special case of the squared loss defined by $L(y, y') = (y' - y)^2$ for all $(y, y') \in \mathcal{Y}^2$, for which the kernel-based regularization algorithm

To be more explicit, h' is derived by training on the sample $\mathcal T$ with empirical distribution $\widehat P$ and the corresponding labels based on f_P . In our adaptation scenario, the unlabeled sample $\mathcal T$ is available to the learner but of course its labels are not. h' is thus the hypothesis obtained using these algorithms in the absence of any adaptation problem.

described in the previous section coincides with *kernel ridge regression* (KRR) [31].

As in the previous section, we will present a pointwise loss guarantee for KRR. But, here, the result will be expressed in terms of $\operatorname{disc}(\widehat{P},\widehat{Q})$ and a term $\delta_H(f_P,f_Q)$ finer that $\eta_H(f_P,f_Q)$ and defined by

$$\delta_H(f_P, f_Q) = \inf_{h \in H} \left\| \underset{x \sim \widehat{P}}{\mathbb{E}} \left[\Delta(h, f_P)(x) \right] - \underset{x \sim \widehat{Q}}{\mathbb{E}} \left[\Delta(h, f_Q)(x) \right] \right\|_K, \tag{10}$$

where $\Delta(h,f)(x)=(h(x)-f(x))\Phi_K(x)$ with Φ_K a feature vector associated to K. Note that when $f_P=f_Q$, $\delta_H(f_P,f_Q)$ is similar to the empirical discrepancy in the sense that it is also based on the difference of the expectations of the same quantity with respect to \widehat{P} and \widehat{Q} . We will use the term weighted feature discrepancy to refer to $\delta_H(f_P,f_Q)$ since it measures the discrepancy between the expectation of Φ_K weighted by the difference of the labeling functions.

Theorem 3 (General pointwise bound for the squared loss) Let L be the squared loss and assume that for all $(x,y) \in \mathcal{X} \times \mathcal{Y}$, $L(h(x),y) \leq M$ and $K(x,x) \leq r^2$ for some M>0 and r>0. Let h' be the hypothesis returned by KRR when minimizing $F_{(\widehat{P},f_P)}$ and h the one returned when minimizing $F_{(\widehat{Q},f_Q)}$. Then, for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$,

$$\left| L(h'(x), y) - L(h(x), y) \right| \le \frac{r\sqrt{M}}{\lambda} \left(\delta_H(f_P, f_Q) + \sqrt{\delta_H^2(f_P, f_Q) + 4\lambda \operatorname{disc}(\widehat{P}, \widehat{Q})} \right).$$

where $\delta_H(f_P, f_Q)$ is defined as in (10).

The main advantage of this result is its expression in terms of $\delta_H(f_P, f_Q)$ instead of $\eta_H(f_P, f_Q)$. Since $\delta_H(f_P, f_Q)$ is defined as a difference, it admits the satisfying property of vanishing for $\widehat{P} = \widehat{Q}$, which does not hold for $\eta_H(f_P, f_Q)$.

Other benefits of this bound become more clear when the covariate-shift assumption holds. In that case, we can denote by f the shared labeling function: $f_P = f_Q = f$. In the next section, we give an upper bound on $\delta_H(f,f)$ and show in particular that $\delta_H(f,f) = 0$ when KRR is used with Gaussian kernels as in many applications in practice and when the labeling function f verifies some conditions. The bound of the theorem then reduces to the simpler expression:

$$\left| L(h'(x), y) - L(h(x), y) \right| \le 2r \sqrt{\frac{M \operatorname{disc}(\widehat{P}, \widehat{Q})}{\lambda}},$$
 (11)

for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$.

We also give a bound in terms of δ_H on the difference of common aggregate measures of the performance of h and h', which gives a simpler guarantee on the difference of root mean square errors (RMSE). In particular, this bound, does not depend on M.

Theorem 4 (Bound on RMSEs) Assume that L is bounded and that for all $x \in \mathcal{X}$, $K(x,x) \leq r^2$ for some r>0. Let h' be the hypothesis returned by KRR when minimizing $F_{(\widehat{Q},f_Q)}$ and h the one returned when minimizing $F_{(\widehat{Q},f_Q)}$. Then, the following holds:

$$\left(\sqrt{\mathcal{L}_P(h',f)} - \sqrt{\mathcal{L}_P(h,f)}\right)^2 \le \frac{r}{2\lambda} \left(\delta_H(f) + \sqrt{\delta_H(f)^2 + 4\lambda \operatorname{disc}(\widehat{P},\widehat{Q})}\right). \tag{12}$$

The proofs of both of these theorems are given in the Appendix.

3.4 Analysis of weighted feature discrepancy

Here, we analyze in more detail the term $\delta_H(f)$ in terms of $d_p(f, H)$, the distance of f to the hypothesis set H for the norm $\|\cdot\|_p$, $p \ge 1$:

$$d_p(f, H) = \inf_{h \in H} ||f - h||_p, \tag{13}$$

and the L_q distance between the distributions \widehat{P} and \widehat{Q} , $L_q(\widehat{P},\widehat{Q})$. For any set $\mathcal{A}\subseteq\mathcal{X}$, let $f_{|\mathcal{A}}$ denote the restriction of f to \mathcal{A} and similarly $h_{|\mathcal{A}}$ the restriction of $h\in H$ to \mathcal{A} . Then, we define more generally $d_p(f_{|\mathcal{A}},H_{|\mathcal{A}})$ as the distance $d_p(f_{|\mathcal{A}},\{h_{|\mathcal{A}}\colon h\in H\})$.

Proposition 1 Assume that for all $x \in \mathcal{X}$, $K(x,x) \leq r^2$ for some r > 0. Let \mathcal{A} denote the union of the supports of \widehat{P} and \widehat{Q} . Then, for any p,q > 1, with 1/p + 1/q = 1,

$$\delta_H(f) \le d_p(f_{|\mathcal{A}}, H_{|\mathcal{A}}) L_q(\widehat{P}, \widehat{Q}) r. \tag{14}$$

Proof. We can rewrite $\delta_H(f)$ and upper bound it as follows:

$$\delta_{H}(f) = \inf_{h \in H} \left\| \sum_{x \in \mathcal{A}} \left(\widehat{P}(x) - \widehat{Q}(x) \right) \left(h(x) - f(x) \right) \Phi_{K}(x) \right\|_{K}$$

$$\leq \inf_{h \in H} \sum_{x \in \mathcal{A}} \left| \widehat{P}(x) - \widehat{Q}(x) \right| \left| h(x) - f(x) \right| \left\| \Phi_{K}(x) \right\|_{K}$$

$$\leq r \inf_{h \in H} \sum_{x \in \mathcal{A}} \left| \widehat{P}(x) - \widehat{Q}(x) \right| \left| h(x) - f(x) \right|.$$

Thus, by Hölder's inequality, we can write

$$\delta_H(f) \le r \inf_{h \in H} L_q(\widehat{P}, \widehat{Q}) \|h_{|\mathcal{A}} - f_{|\mathcal{A}}\|_p \le r L_q(\widehat{P}, \widehat{Q}) d_p(f_{|\mathcal{A}}, H_{|\mathcal{A}}),$$

which concludes the proof.

Thus, when $f_{|\mathcal{A}}$ is in the closure of $H_{|\mathcal{A}}$ for $\|\cdot\|_p$, then $d_p(f_{|\mathcal{A}}, H_{|\mathcal{A}}) = 0$ and thus $\delta_H(f) = 0$, regardless of the distributions \widehat{P} and \widehat{Q} . In particular, if the kernel K is universal, for example if K is a Gaussian kernel, then $\mathbb{H}_{|\mathcal{A}}$ is dense in the family

of functions with finite support \mathcal{A} (see [33][proposition 5]), thus $d_{\infty}(f_{|\mathcal{A}}, \mathbb{H}_{|\mathcal{A}}) = 0$ and $\delta_{\mathbb{H}}(f) = 0$. Note that this does not require the labeling function f to be continuous. Of course, since H is a subset of the full Hilbert space \mathbb{H} with function norms bounded by Λ , in general we may not have $\delta_H(f) = 0$, but this may hold if Λ is sufficiently large. More generally, if $f_{|\mathcal{A}}$ is ϵ -close to the family of hypotheses in H restricted to \mathcal{A} for $\|\cdot\|_{\infty}$, then, in view of the result of the proposition, we have $\delta_H(f) \leq 2r\epsilon$, that is $\delta_H(f) \leq 2\epsilon$ for a normalized kernel such as Gaussian kernel since r=1.

4 Discrepancy minimization (DM) adaptation algorithm

For small values of $\eta_H(f_P, f_Q)$ or $\delta_H(f_P, f_Q)$, in particular when these terms vanish as in some scenarios already discussed, the guarantees presented in the previous section depend only on the empirical discrepancy $\operatorname{disc}(\widehat{P}, \widehat{Q})$. This suggests seeking to minimize the empirical discrepancy by selecting an empirical distribution q^* , among the family $\mathcal Q$ of all distributions with a support included in that of \widehat{Q} [20]:

$$q^* = \operatorname*{argmin}_{q \in \mathcal{Q}} \operatorname{disc}(\widehat{P}, q). \tag{15}$$

Using q^* instead of \widehat{Q} amounts to reweighting the loss on each training point. This forms the basis of our discrepancy minimization (DM) adaptation algorithm which consists of two stages:

- (i) first, computing q^* ;
- (ii) then, modifying (4) using q^* as follows:

$$F_{(q^*, f_Q)}(h) = \sum_{i=1}^{m} q^*(x_i) L(h(x_i), y_i) + \lambda ||h||_K^2,$$
(16)

and finding a minimizing hypothesis h.

The minimization of $F_{(q^*,f_Q)}$ is no more difficult than that of $F_{(\widehat{Q},f_Q)}$. It leads to a convex optimization problem similar to the one corresponding to $F_{(\widehat{Q},f_Q)}$ and can be solved in the same way. In fact, most software tools available for solving the optimization problem for $F_{(\widehat{Q},f_Q)}$ also provide the interface for solving the optimization problem $F_{(\widehat{q},f_Q)}$ with an arbitrary empirical distribution $\widehat{q} \in \mathcal{Q}$.

In particular, for the squared loss, the objective can be reduced to a form similar to that of KRR and shown to admit a closed-form solution both in the primal and the dual as in the case of KRR. Indeed, let Φ_K be a feature mapping associated to the kernel K, a hypothesis h is then of the form $x \mapsto \mathbf{w} \cdot \Phi(x)$ for some $\mathbf{w} \in \mathbb{R}^N$. Observe then that the empirical error term can be rewritten as follows

$$\frac{1}{m} \sum_{i=1}^{m} q^*(x_i) (\mathbf{w} \cdot \Phi(x_i) - y_i)^2 = \frac{1}{m} \sum_{i=1}^{m} (\mathbf{w} \cdot \Psi(x_i) - y_i')^2,$$

with
$$\Psi(x_i) = \sqrt{q^*(x_i)}\Phi(x_i)$$
 and $y_i' = \sqrt{q^*(x_i)}y_i$.

Thus, in the following section, we focus on the first stage of our algorithm and study in detail the optimization problem (15). We will show that in the case of the squared loss it can be formulated as a semi-definite programming problem, including when using kernels.

5 Optimization Problems

Let \mathcal{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_Q 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 $\mathbb{R}^{\mathfrak{m}}$: $\Delta_{\mathfrak{m}} = \{\mathbf{z} \in \mathbb{R}^{\mathfrak{m}} : z_i \geq 0 \land \sum_{i=1}^{\mathfrak{m}} z_i = 1\}$.

5.1 Discrepancy minimization in feature space

We showed in [20] that the problem of minimizing the empirical discrepancy for the squared loss and the hypothesis space $H = \{\mathbf{x} \mapsto \mathbf{w}^T\mathbf{x} \colon ||\mathbf{w}|| \le \Lambda\}$ of bounded linear functions can be cast as the following convex optimization problem:

$$\min_{\mathbf{z} \in \Delta_{m}} \|\mathbf{M}(\mathbf{z})\|_{2},\tag{17}$$

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

$$\mathbf{M}(\mathbf{z}) = \mathbf{M}_0 - \sum_{i=1}^{m} z_i \mathbf{M}_i, \tag{18}$$

with $\mathbf{M}_0 = \sum_{j=\mathfrak{m}+1}^{\mathfrak{q}} \widehat{P}(\mathbf{x}_j) \mathbf{x}_j \mathbf{x}_j^{\mathsf{T}}$ and for $i \in [1,\mathfrak{m}]$ $\mathbf{M}_i = \mathbf{x}_i \mathbf{x}_i^{\mathsf{T}}$, $\mathbf{x}_i \in S_Q$. The optimal value of the objective function is the minimal empirical discrepancy. 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:

$$\min_{\mathbf{z},t} \quad t \\
\text{subject to} \quad \begin{bmatrix} t\mathbf{I} & \mathbf{M}(\mathbf{z}) \\ \mathbf{M}(\mathbf{z}) & t\mathbf{I} \end{bmatrix} \succeq 0 \ \land \ \mathbf{1}^{\mathsf{T}}\mathbf{z} = 1 \ \land \ \mathbf{z} \ge 0.$$

This problem can be solved in polynomial time using interior point methods. As shown by [27][pp.234-235], the time complexity for each iteration of the algorithm is in our notation : $O(\mathfrak{m}^3 + \mathfrak{m} N^3 + \mathfrak{m}^2 N^2 + \mathfrak{n} N^2)$. 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. The unconstrained version of this problem, that is one where z is not constrained to be in the simplex, has also been extensively studied in a number of optimization publications, in particular [28].

5.2 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 $\mathbf{K} = [K(\mathbf{x}_i, \mathbf{x}_j)]_{ij} \in \mathbb{R}^{q \times q}$ the kernel matrix associated to K for the full sample $S = S_O \cup S_P$ and for any $\mathbf{z} \in \mathbb{R}^m$ by $\mathbf{D}(\mathbf{z})$ the diagonal matrix

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

Theorem 5 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 (17) but that depends only on the Gram matrix of the kernel K:

$$\min_{\mathbf{z} \in \Delta_{\mathfrak{m}}} \quad \|\mathbf{M}'(\mathbf{z})\|_2 \tag{20}$$

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 $\mathbf{D}_0 = \operatorname{diag}(0, \ldots, 0, \widehat{P}(\mathbf{x}_{\mathfrak{m}+1}), \ldots, \widehat{P}(\mathbf{x}_{\mathfrak{m}+\mathfrak{n}}))$, and for $i \geq 1$, \mathbf{D}_i is the diagonal matrix of the ith unit vector.

Proof. Let $\Phi \colon \mathcal{X} \to \mathcal{F}$ 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

$$\min_{\substack{\|\mathbf{z}\|_1=1\\\mathbf{z}>0}} \{\lambda_{\max}(\mathbf{M}(\mathbf{z})), \lambda_{\max}(-\mathbf{M}(\mathbf{z}))\}, \tag{21}$$

where the matrix M(z) is defined by

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

with q^* given by: $q^*(\mathbf{x}_i) = z_i$ for all $i \in [1, \mathfrak{m}]$. Let Φ 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

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

It is known that for any two matrices $\mathbf{A} \in \mathbb{R}^{N' \times \mathfrak{q}}$ and $\mathbf{B} \in \mathbb{R}^{\mathfrak{q} \times N'}$, $\mathbf{A}\mathbf{B}$ and $\mathbf{B}\mathbf{A}$ have the same eigenvalues. Thus, matrices $\mathbf{M}(\mathbf{z}) = (\mathbf{\Phi}\mathbf{D}(\mathbf{z}))\mathbf{\Phi}^{\mathsf{T}}$ and $\mathbf{\Phi}^{\mathsf{T}}(\mathbf{\Phi}\mathbf{D}(\mathbf{z})) = \mathbf{K}\mathbf{D}(\mathbf{z})$ have the same eigenvalues. $\mathbf{K}\mathbf{D}(\mathbf{z})$ is not a symmetric matrix. To ensure that we obtain an SDP of the same form as (17) minimizing the spectral norm of

Algorithm 1

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

Fig. 1. Convex optimization algorithm.

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 $\mathbf{K}\mathbf{D}(\mathbf{z})$ 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 $\mathbf{D} = \mathbf{D}_0 - \sum_{i=1}^m z_i \mathbf{D}_i$, 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.

6 Optimization solution

6.1 Solution based on smooth approximation

This section presents an algorithm for solving the discrepancy minimization problem using the smooth approximation technique of Nesterov [25]. A general algorithm was given by Nesterov [23] to solve convex optimization problems of the form

$$minimize_{\mathbf{z}\in C}F(\mathbf{z}), \tag{22}$$

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 $\mathop{\mathrm{argmin}}_{\mathbf{v} \in C} \langle \nabla F(\mathbf{u}), \mathbf{v} - \mathbf{u} \rangle + \frac{1}{2}L\|\mathbf{v} - \mathbf{u}\|^2$ (the specific choice of the minimizing \mathbf{v} is arbitrary for a given \mathbf{u}). d denotes a proxfunction 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 $\mathbf{u}_0 = \mathop{\mathrm{argmin}}_{\mathbf{u} \in C} d(\mathbf{u})$. The following convergence guarantee was given for this algorithm [25].

Theorem 6 Let \mathbf{z}^* be an optimal solution for problem (22) 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 2

$$\begin{aligned} \mathbf{u}_0 &\leftarrow \operatorname{argmin}_{\mathbf{u} \in C} \mathbf{u}^\mathsf{T} \mathbf{J} \mathbf{u} \\ \text{for } k \geq 0 \text{ do} \\ \mathbf{v}_k &\leftarrow \operatorname{argmin}_{\mathbf{u} \in C} \frac{2p-1}{2} (\mathbf{u} - \mathbf{u}_k)^\mathsf{T} \mathbf{J} (\mathbf{u} - \mathbf{u}_k) + \nabla G_p (\mathbf{M}(\mathbf{u}_k))^\mathsf{T} \mathbf{u} \\ \mathbf{w}_k &\leftarrow \operatorname{argmin}_{\mathbf{u} \in C} \frac{2p-1}{2} (\mathbf{u} - \mathbf{u}_0)^\mathsf{T} \mathbf{J} (\mathbf{u} - \mathbf{u}_0) + \sum_{i=0}^k \frac{i+1}{2} \nabla G_p (\mathbf{M}(\mathbf{u}_i))^\mathsf{T} \mathbf{u} \\ \mathbf{u}_{k+1} &\leftarrow \frac{2}{k+3} \mathbf{w}_k + \frac{k+1}{k+3} \mathbf{v}_k \\ \text{end for} \\ \text{return } \mathbf{v}_k \end{aligned}$$

Fig. 2. Smooth approximation algorithm.

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 [26]. This can be done by finding a uniform ϵ -approximation of F by a smooth convex function G 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:

minimize
$$\|\mathbf{M}(\mathbf{z})\|_2$$
 (23)
subject to $\mathbf{M}(\mathbf{z}) = \sum_{i=0}^{m} z_i \mathbf{M}_i \wedge z_0 = -1 \wedge \sum_{i=1}^{m} z_i = 1 \wedge \forall i \in [1, m], z_i \ge 0,$

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 symmetric positive semi-definite (SPSD) matrices. Thus, here $C = \{\mathbf{z} \in \mathbb{R}^{\mathfrak{m}+1} \colon z_0 = -1 \land \sum_{i=1}^{\mathfrak{m}} z_i = 1 \land \forall i \in [1,\mathfrak{m}], z_i \geq 0\}$. 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 $F: \mathbf{z} \mapsto \|\mathbf{M}(\mathbf{z})\|_2$ 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 to approximate function F^2 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 [26], the function $G_p: \mathbf{M} \mapsto \frac{1}{2} \operatorname{Tr}[\mathbf{M}^{2p}]^{\frac{1}{p}}$, where $p \geq 1$ is an integer, can be used to give a smooth approximation of $\frac{1}{2}F^2$. 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 $\mathbf{M} \in \mathbb{S}^N$, $G_p(\mathbf{M}) = \frac{1}{2} \left[\sum_{i=1}^N \lambda_i^{2p}(\mathbf{M}) \right]^{\frac{1}{p}}$, thus

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

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

$$\mathbf{z} \in C$$
,

$$\frac{1}{2} \|\mathbf{M}(\mathbf{z})\|_{2}^{2} \le G_{p}(\mathbf{M}(\mathbf{z})) \le \frac{1}{2} r^{\frac{1}{p}} \|\mathbf{M}(\mathbf{z})\|_{2}^{2}.$$
 (24)

This leads to a smooth approximation algorithm for solving the SDP (23) derived from Algorithm 1 by replacing the objective function F with G_p . G_p can be shown to admit a Lipschitz gradient with Lipschitz constant L = (2p-1) with respect to the norm $\|\cdot\|_{\mathbf{J}}$. Choosing the prox-function $d\colon \mathbf{u}\mapsto \frac{1}{2}\|\mathbf{u}-\mathbf{u}_0\|_{\mathbf{J}}^2$ with convexity parameter $\sigma=1$ 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 $O(\sqrt{r\log r}/\epsilon)$.

Theorem 7 Let $r = \max_{\mathbf{z} \in C} \operatorname{rank}(\mathbf{M}(\mathbf{z}))$. For any $\epsilon > 0$, Algorithm 2 solves the SDP (23) with relative accuracy ϵ in at most $4\sqrt{(1+\epsilon)r\log r}/\epsilon$ iterations using the objective function G_p with $p \in [q_0, 2q_0)$, where $q_0 = (1+\epsilon)(\log r)/\epsilon$.

Proof. The proof follows directly [25] and is given in Appendix D for completeness. \Box

The first step of the algorithm consists of computing the vector \mathbf{u}_0 by solving the simple QP of line 1. Note that \mathbf{u}_0 is the minimizer of $\mathbf{z} \mapsto \|\mathbf{M}(\mathbf{z})\|_F^2$ since, for any $\mathbf{z} \in C$, the following identity holds:

$$\mathbf{z}^{ op}\mathbf{J}\mathbf{z} = \sum_{i,j=0}^m z_i z_j \langle \mathbf{M}_i, \mathbf{M}_j \rangle_F = \Big\| \sum_{i=0}^m z_i \mathbf{M}_i \Big\|_F^2 = \| \mathbf{M}(\mathbf{z}) \|_F^2.$$

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 $\sum_{i=0}^k \frac{i+1}{2} \nabla G_p(\mathbf{M}(\mathbf{u}_i))^T \mathbf{u}$ required at line 4 can be computed in constant time from its value at the previous iteration. Since for any $\mathbf{z} \in \mathbb{R}^m$

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

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

$$[\nabla G_p(\mathbf{M}(\mathbf{z}))]_i = \langle \mathbf{M}^{2p-1}(\mathbf{z}), \mathbf{M}_i \rangle_F \operatorname{Tr}[\mathbf{M}^{2p}(\mathbf{z})]^{\frac{1}{p}-1}, \tag{25}$$

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 [7]. 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 5, 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

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

Thus, by the property of the trace operator,

$$\operatorname{Tr}[\mathbf{M}^{\prime 2p}(\mathbf{z})] = \operatorname{Tr}[\mathbf{D}(\mathbf{z})\mathbf{K}^{1/2}\mathbf{K}^{1/2}[\mathbf{D}(\mathbf{z})\mathbf{K}]^{2p-1}] = \operatorname{Tr}[[\mathbf{D}(\mathbf{z})\mathbf{K}]^{2p}]. \tag{26}$$

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

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

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

$$\langle \mathbf{M}^{\prime 2p-1}(\mathbf{z}), \mathbf{M}_i^{\prime} \rangle_F = [\mathbf{K}[\mathbf{D}(\mathbf{z})\mathbf{K}]^{2p-1}]_{ii}. \tag{27}$$

Therefore, the diagonal of the matrix $\mathbf{K}[\mathbf{D}(\mathbf{z})\mathbf{K}]^{2p-1}$ provides all these terms. Thus, in view of (26) and (27), the gradient given by (25) 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 $O((\mathfrak{m}+\mathfrak{n})^2)$.

The analysis of the time and space complexity just presented assumed a standard method for the computation of the powers of a matrix, such as the binary decomposition method [7]. However, the matrices we consider here admit a special structure since they are given as a linear combination of rank-one matrices. It is likely that a faster computational method taking advantage of the structure would be possible with even less space requirements. That could significantly reduce the computational time and space requirements of each step of our algorithm, thereby resulting in a substantially more efficient overall optimization solution.

Projected subgradient update

```
\begin{split} &(\mathbf{u},\lambda) \leftarrow (\mathsf{max\text{-}eigenvector}(\mathbf{M}(\mathbf{z})), \lambda_{\mathsf{max}}(\mathbf{M}(\mathbf{z}))) \\ &(\mathbf{u}',\lambda') \leftarrow (\mathsf{min\text{-}eigenvector}(\mathbf{M}(\mathbf{z})), \lambda_{\mathsf{min}}(\mathbf{M}(\mathbf{z}))) \\ &\mathbf{if} \; (|\lambda| > |\lambda'|) \; \mathbf{then} \\ &\mathbf{g} \leftarrow [0, \mathbf{u}^{\top} \mathbf{M}_1 \mathbf{u}, \dots, \mathbf{u}^{\top} \mathbf{M}_{\mathfrak{m}} \mathbf{u}]^{\top} \\ &\mathbf{else} \\ &\mathbf{g} \leftarrow [0, -\mathbf{u}'^{\top} \mathbf{M}_1 \mathbf{u}', \dots, -\mathbf{u}'^{\top} \mathbf{M}_{\mathfrak{m}} \mathbf{u}']^{\top} \\ &\mathbf{end} \; \mathbf{if} \\ &\mathbf{z} \leftarrow \mathbf{z} - \eta \, \mathbf{g} \\ &\mathbf{for} \; i \leftarrow 1 \; \mathbf{to} \; \mathbf{m} \; \mathbf{do} \\ &z_i \leftarrow \mathsf{max}(z_i, 0) \\ &\mathbf{end} \; \mathbf{for} \\ &a \leftarrow \sum_{i=1}^{\mathfrak{m}} z_i \\ &\mathbf{for} \; i \leftarrow 1 \; \mathbf{to} \; \mathbf{m} \; \mathbf{do} \\ &z_i \leftarrow z_i/a \\ &\mathbf{end} \; \mathbf{for} \end{split}
```

Fig. 3. Main update rule of the projected subgradient algorithm.

6.2 Comparison with the projected-subgradient method

Here, we briefly discuss an alternative optimization technique and compare it with the smooth optimization method we used.

An alternative technique for solving the optimization problem (23) consists of using the standard projected-subgradient method. The objective function $F: \mathbf{z} \mapsto \|\mathbf{M}(\mathbf{z})\|_2$ of problem (23) is not differentiable but it admits a subdifferential at any point (see for example [19] for the gradient computation).

The projected-subgradient method consists of iteratively taking a step in the direction of a negative subgradient at the current tentative solution, followed by a projection on the convex set defined by the constraints. The objective value is not guaranteed to decrease at each step, thus, the method also requires keeping track of the smallest objective value among the iterative values found.

The subgradient of F at any point z can be computed in terms of the eigenvectors of M(z) for its maximum or minimum eigenvalue. Since any subgradient of function F can be used, one can select for its computation an arbitrary maximum (or minimum) eigenvector of matrix M(z), which can be computed in $O(N^2)$ [15]. Let u be the current eigenvector corresponding to the maximum eigenvalue of M(z), then Figure 3 describes the main update rule of the algorithm for a step size $\eta > 0$.

Besides from the $O(N^2)$ algorithm for the computation of the maximum eigenvector the projected subgradient update is relatively simple to implement. The known complexity result for the number of iterations for the subgradient methods is $O(1/\epsilon^2)$, (see for example [24]). This $O(1/\epsilon^2)$ dependence can be very slow, as is well known in optimization, and it is also typically what is experienced in

practice. Since matrices \mathbf{M}_i are simple outer product matrices, each term $\mathbf{u}^{\top}\mathbf{M}_i\mathbf{u}$ can be computed in O(N), but the recomputation of $\mathbf{M}(z)$ after each update of \mathbf{z} takes $O(\mathfrak{m}N^2)$. Therefore, the computational cost of each step of the algorithm, including the computation of the maximum eigenvector and the subgradient and that of $\mathbf{M}(\mathbf{z})$, is in $O(\mathfrak{m}N^2)$. Thus, the overall complexity of the algorithm is in $O(\mathfrak{m}N^2/\epsilon^2)$. In comparison, the overall time complexity of our algorithm is $O([(\log p)N^{2+\alpha}+\mathfrak{m}N^2]/\epsilon)$, with $\alpha=.376$. For $(\log p)N^{\alpha}>\mathfrak{m}$, for the regime of $\epsilon\leq\mathfrak{m}/[(\log p)N^{\alpha}]$, the complexity of our algorithm is more favorable, while that of the projected subgradient algorithm is more favorable in the opposite case. For $(\log p)N^{\alpha}\leq\mathfrak{m}$, the complexity of our algorithm is more favorable by a factor of $O(1/\epsilon)$.

7 Experiments

We report the results of extensive experiments both with artificial data sets and real-world adaptation data sets.

Our first set of experiments in sentiment analysis demonstrate the effectiveness of the DM algorithm in domain 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 highdimensional feature spaces. We describe in detail these experiments and report our results.

We also report the results of extensive experiments with the DM algorithm in the scenario of sample bias correction and a comparison with three sample bias correction algorithms applicable in regression: KMM and KLIEP, which are shown to be state-of-the-art through the extensive and thorough analysis and experimentation presented in [35], and the two-stage version of the procedure of Bickel et al.[3] (Two-Stage). This study provides the first comparative analysis of the DM algorithm in terms of regression error for the problem of sample bias correction in regression.

7.1 Artificial data sets - domain adaptation

Our first set of experiments served to test the efficiency and effectiveness of our algorithm and is carried out on artificial data. For these experiments, the source input distribution was a mixture of N Gaussians with width $\sigma=1$ randomly centered in $[-1;1]^N$, while the target input distribution is a single randomly placed Gaussian in $[-1;1]^N$, also with width $\sigma=1$. The source distribution was constructed to have 20% of its distribution mass from this single target Gaussian distribution to ensure closeness of the source and target distributions, a condition needed for adaptation to be possible.

The labeling function used was defined as the sum of the absolute values of the input vector's components and was the same for the source and target distribution: $f_Q = f_P \colon \mathbb{R}^N \to \mathbb{R}$, with $f_P(\mathbf{x}) = \sum_{i=1}^N |x_i|$ for all $\mathbf{x} = (x_1, \dots, x_N)^\top \in \mathbb{R}^N$. The learning algorithm used was weighted kernel ridge regression (wKRR) with a Gaussian kernel with $\sigma = N$ and a small ridge $\lambda \approx 0.015$, chosen to provide the best performance of the oracle setting when training on the target distribution. In the cost function for wKRR the loss on each training point was reweighted using the solution of the discrepancy minimization given by the kernelized version of Algorithm 2.

In our experiments, we varied the dimension N of the input space $\mathbf{x} \in \mathbb{R}^N$, $N \in \{32, 64, 128, 256\}$ as well as the number of labeled points m from the source distribution and the number of unlabeled points n from the target distribution. Both m and n were kept in the range from 100 to $2{,}000$ points.

The plots of Figure 4 correspond to m=200 and n=200. They show the performance of our adaptation algorithm as a function of the number of iterations of Algorithm 2. The performance values were obtained by determining the value of the smooth approximation function G_p : $\mathbf{M} \mapsto \mathrm{Tr}[\mathbf{M}^{2p}]^{\frac{1}{p}}$ after each iteration of Algorithm 2 and by re-running wKRR to monitor the progress of the root mean squared error (RMSE). The value p of the smooth approximation was kept fixed at p=16 as numerous experiments demonstrated that the iterative powers of the matrices involved converged quickly in just a few steps. The plots show mean values over 50 runs for the same parameter settings. For reference, we also display the performance obtained by training on the unweighted source distribution (naive, in blue) and by training on labeled sample of the same size as the training set but sampled from the target distribution (oracle, in green).

All of these plots demonstrate the benefits of Algorithm 2 with the RMSE improving significantly over the naive solution. The optimization quickly reaches a good solution and continues to improve, albeit slowly, from there. As expected, as the value of G_p decreases, so does the value of the RMSE. Note that, since the mixtures of Gaussians are randomly generated, the tasks used for illustration are not equally hard and the values obtained cannot be easily compared.

7.2 Real-world data sets - domain adaptation

For our experiments with real-world data sets, we used the multi-domain sentiment dataset (version 1.0) of Blitzer et al. [4]. This data set has been used in several publications [5, 13, 20], 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

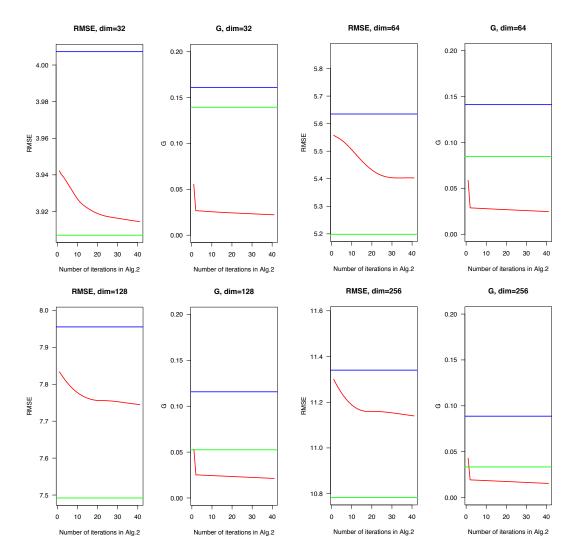


Fig. 4. The performance of our adaptation algorithm as a function of the number of iteration of Algorithm 2. For reference is shown in blue the performance of the naive solution consisting of training on the un-weighted training set, and in green the performance of the oracle solution consisting of training on a set of the same size but drawn from the target distribution. The plots display the RMSE and the value of G for four different values of the dimension N=32,64,128 and 256.

for each of the four tasks books, dvd, elec, kitchen with a Gaussian kernel ridge regression with a relatively small width $\sigma=1$ using as features the normalized counts (mean zero, variance one) 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 (task, task'), where task and task' are in {books, dvd, elec, kitchen}. For each of these problems, the source empirical distribution is a mixture defined by 500 labeled points from task and 200 from task'. This is intended to make the source and target distributions reasonably close, a condition for the theory developed in this paper,

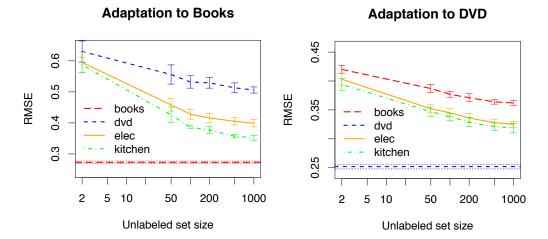


Fig. 5. Performance improvement of the RMSE for the adaptation tasks with target domains books and DVD as a function of the size of the unlabeled data used. Note that the figures do not make use of the same y-scale.

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 task'.

Figures 5 and 6 show 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 \pm one standard deviation. The value of the ridge parameter λ is chosen as the one that gives the best performance when training and testing on the same domain. 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 kitchen task primarily discusses electronic gadgets for kitchen use, and hence the kitchen and elec data sets adapt well to each other, an observation also made by Blitzer et al. [4].

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. For brevity, 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.

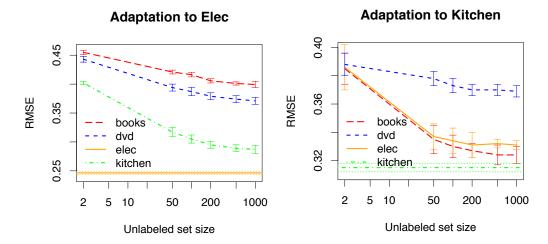


Fig. 6. Performance improvement of the RMSE for the adaptation tasks with target domains Elec and Kitchen as a function of the size of the unlabeled data used. Note that the figures do not make use of the same y-scale.

Algorithm 2 requires solving several QPs to compute \mathbf{u}_0 and \mathbf{u}_{k+1} , $k \geq 0$. Since $\mathbf{u}_{k+1} \in \mathbb{R}^{m+1}$, the cost of these computations only depends on the size of the labeled sample \mathfrak{m} , which is relatively small. Figure 7 displays average run times obtained for $\mathfrak{m} + \mathfrak{n}$ 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 quadprog optimization package. As can be seen from the figure, the run times scale cubically in the sample size, reaching roughly 10s for $\mathfrak{m} + \mathfrak{n} = 1{,}000$.

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 6. The iterated power method 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 7 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 a good estimate of the solution can be found. For 5 iterations the total time is only $5\times(80+2*10)+10=510$ seconds; with 20 iterations of Algorithm 2 the total time is $20\times(80+2*10)+10=2010$ seconds, still only about 30 minutes.

In contrast, even one of 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, SeDuMi³ simply failed for set sizes larger than $\mathfrak{m}+\mathfrak{n}=750!$ In Figure 7, typical run times for Algorithm 2 with 5 iterations (blue) and 20 iterations (green) are compared to run times for solving the SDP problem using SeDuMi (red).

³ See http://sedumi.ie.lehigh.edu/.

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

7.3 Artificial data sets - sample bias correction

Our sample bias correction experiments with synthetic data served first to illustrate the properties of the algorithms compared. We borrowed our example from [35]. It is a one-dimensional toy problem of learning the function $f: x \mapsto \sin(x)/x$ with $Q(x) \sim \mathcal{N}(x,1,(1/2)^2)$ and $P(x) \sim \mathcal{N}(x,2,(1/4)^2)$, where $\mathcal{N}(\cdot,\mu,\sigma)$ is a normal distribution with mean value μ and variance σ^2 , see Figure 8(left). A second purpose of our experiment was to verify that our implementations and results match those of the KLIEP paper [35]. The labeled points are corrupted with random Gaussian noise with zero mean value and variance $1/4^2$. For learning we drew m=200 points from Q and m=100 points from Q. In addition, we drew a test set of size 1,000 points from Q used only for evaluation.

The feature space used for matching the distributions was modeled by Gaussians as in [35]. Note that since we used a Gaussian kernel, as discussed in Section 3 we may have $\delta_H(f)=0$ depending on the value of Λ and the magnitude of the labels defined by f. The KLIEP algorithm admits a selection criterion J for choosing the optimal width of the Gaussian. The plot of that criterion as measured over ten independent runs is given in Figure 8(right).

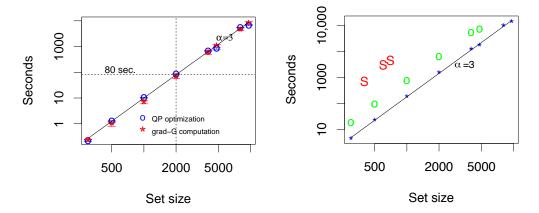


Fig. 7. The left panel shows a plot reporting run times measured empirically (mean \pm one standard deviation) for the QP optimization and the computation of ∇G_p as a function of the sample size (log-log scale). The right panel compares the total time taken by Algorithm 2 to compute an approximate solution using 5 iterations (blue) and the optimization solution using 20 iterations (green), to the one taken by SeDuMi in red (log-log scale).

In accordance with [35] we found an optimal value of $\sigma \approx 0.2$ using J. Figure 9 provides examples of how the four algorithms, KMM, KLIEP, DM, and Two-Stage fit the empirical distribution \hat{P} for different values of σ . Note that the distributions \hat{Q} and \hat{P} are the same in all plots, the only change in the plots is the width of the Gaussians used for modeling the data and the algorithm used.

Indeed, $\sigma=0.2$ is the value for which KLIEP best matches \widehat{P} (Figure 9, middle-left panel). For larger values of σ KLIEP tends towards the uniform distribution. This is also concluded in [35] and our plots are very similar to those provided in this reference. Both KMM and DM appear to produce better matches to \widehat{P} for all values of σ , though for $\sigma=0.2$ the resulting distributions are somewhat smoother. Two-Stage appears to be producing matches slightly better than KLIEP.

Following the DM learning algorithm, the target function was learned using the same Gaussian width $\sigma = 0.2$ as the one used for matching the distributions. The value of the ridge was estimated by 10-fold cross-validation on \hat{Q} to $\lambda = 0.01$. Figure 8 (c) illustrates the performance on the test set of the algorithms for these parameter settings. The solid lines correspond to mean values over 10 runs. Standard deviations are of the order 0.1. As baselines, we include two comparisons. 'Uniform' corresponds to no re-weighting of Q. That is, we simply train on Q and use the resulting model to predict on the test set. 'Optimal' corresponds to the hypothetical scenario where a labeled set from P of the same size as Q would have been available for training. In this example, DM outperforms both KLIEP, Two-Stage, and KMM. It is interesting to observe how the performance of the DM algorithm improves with the number of iterations used in the minimization of the discrepancy as a function of \sqrt{G} , where G is the smooth objective function minimized by Cortes et al. [8] to determine q^* . The plot on the right-hand side shows how \sqrt{G} decreases with the number of iterations to determine q^* . We found that typically after about 30 rounds, the changes in \sqrt{G} were minimal.

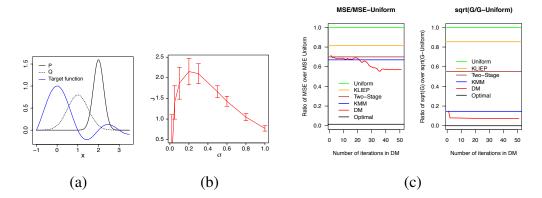


Fig. 8. Left panel: (a) distributions and target function used for the synthetic data example. (b): the value of the selection criterion, J, used by KLIEP. Right panel (c): Normalized performance of the different algorithms on the toy example shown as a function of the number of iterations of the algorithm computing the minimal empirical discrepancy q^* for DM. The right panel illustrates how the discrepancy estimate \sqrt{G} depends on the number of iterations. The MSE and G values are scaled so the average of Uniform is 1.

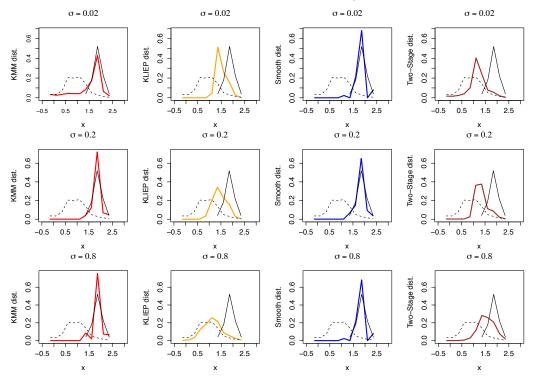


Fig. 9. Histograms of the distributions obtained with KMM (left), KLIEP (middle-left), DM (middle-right), and Two-Stage (right). The solid-line black histogram represents the empirical distribution \widehat{P} to be matched, the dashed-line histograms represent \widehat{Q} .

7.4 Real-world data sets - sample bias correction

We also compared the performance of DM with KMM, KLIEP and Two-Stage on a large set of regression datasets from the DELVE repository. ⁴ In [8] it was

⁴ See http://www.cs.toronto.edu/~delve/.

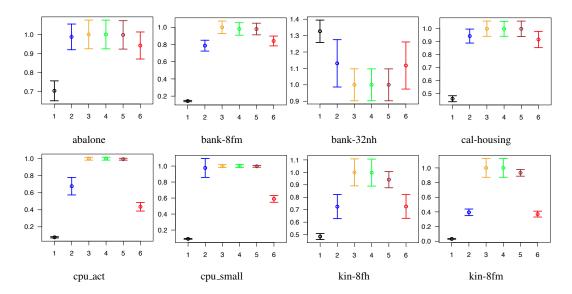


Fig. 10. Results with KLIEP-paper biasing scheme: Relative MSE performance of (1): Optimal (in black); (2): KMM (in blue); (3): KLIEP (in orange); (4): Uniform (in green); (5): Two-Stage (in brown); and (6): DM (in red). Errors are normalized so that the average MSE of Uniform is 1.

demonstrated that the smooth approximation for (15) provided an efficient iterative algorithm capable of training an order of magnitude more examples than previously possible. However, the performance of the algorithm in terms of accuracy was only tested for adaptation problems and it was not compared to other state-of-the-art algorithms.

For all these datasets, we created biased samples by a procedure similar to that of the KLIEP paper. We rescaled each feature to the interval [0,1]. Thus, each input vector x was in $[0,1]^d$ where d is the dimension of the input space. We randomly chose a labeled sample (x_k,y_k) from the pool of samples and accepted it with probability $\Pr[x_k] = \min(1,4(x_k(c))^2)$, where $x_k(c)$ is the c-th coordinate of x_k and c is randomly determined but fixed in each trial of the experiments. We repeated this procedure until we accepted m=200 target samples for \widehat{Q} . From the rest we uniformly chose n=200 test samples for \widehat{P} . Excluding both the training and test points already selected, we finally uniformly chose 1,000 samples on which the final performance was evaluated.

Each experiment was repeated ten times and we report mean values plus standard deviations. The results are summarized in Figure 10. The name of the dataset is indicated beneath each figure. All values are scaled so the mean value of Uniform is 1. KMM does not admit a principled cross-validation technique but KLIEP uses the likelihood function J to determine the Gaussian width [34]. Thus, to favor the KLIEP algorithm, we first used the parameters derived from J for all algorithms (KLIEP, KMM, Two-Stage, and DM). However, in our experiments, that choice of parameters led to results very similar to those obtained by cross-validation on the training set. The use of the KLIEP measure, J, also proved problematic in practice as the optimal value for σ was sometimes found to be at the extreme values of either

very small or very wide Gaussians. Also, note that the theory presented in Section 3 suggests a principled method for selecting the best parameter σ for DM based on the quantity $\mathcal{L}_{\widehat{Q}}(h,f) + 4r[\mathrm{disc}(\widehat{P},\widehat{Q})(M/\lambda)]^{1/2}$ following Theorem 3 in the case where $\delta_H(f) = 0$, and similarly by using Theorem 4. Selecting σ based on this criterion further favors DM in the experimental comparisons and we do not include these results here.

For all experiments the maximum weight of KMM was capped at $B=1{,}000$. For the Two-Stage algorithm, we picked the best performance obtained from a grid search over a wide range of values for the regularization parameter, $1/2\sigma_v^2 \in 2^{[-5,5]}$.

We also carried out a number of experiments with a different bias technique, that oversamples "easy-to-learn". This situation is what we often face in applications: the most useful examples are more rare in the biased sample. For each data set, we first split the data into a training set U and test set T. We extracted 200 points from T defining the empirical distribution \hat{P} . To define the training sample \hat{Q} , we sampled 200 points from U with a bias method that we now describe. To locate the "easy-to-learn" examples we sampled 200 points from T and trained a KRR model h on that sample. The prediction on a sample in U, h(x), is used to form its sample probability $p(x) \propto \frac{1}{(h(x)-y)^4}$. This sampling strategy favors examples with small residual errors.

The results of four experiments with this biasing scheme are summarized in Figure 11. As can be seen from Figure 10 and 11, DM outperforms KMM with statistical significance on a number of the datasets and otherwise matches its performance. KLIEP performs very similarly to Uniform. This in fact is not surprising. As stated by the authors (see footnote 10 in [34]), KLIEP admits no advantage over Uniform when all the training examples are used. However, the authors do not prescribe a technique or justification for selecting a subset of the examples.

8 Conclusion

We presented several new theoretical guarantees for domain adaptation in regression and proved that the empirical discrepancy minimization can be cast as an SDP when using kernels. We gave an efficient algorithm for solving that SDP by exploiting techniques 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 set of results for domain adaptation in regression when using kernel-based regularization algorithms, including theoretical guarantees, an efficient algorithmic solution, and extensive empirical results.

Our results suggest that the discrepancy plays a critical role in adaptation. The minimum discrepancy value found by our algorithm can be used as a *rejection* criterion for some domain adaptation problems since for relatively large values of this quantity, our theoretical pointwise loss guarantees become weaker. This corresponds

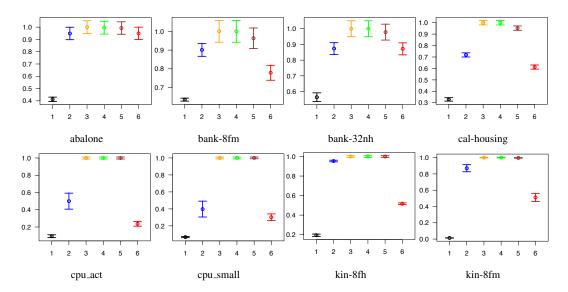


Fig. 11. Results with "easy-to-learn" biasing scheme: Relative MSE performance of (1): Optimal (in black); (2): KMM (in blue); (3): KLIEP (in orange); (4): Uniform (in green); (5): Two-Stage (in brown); and (6): DM (in red). Errors are normalized so that the average MSE of Uniform is 1.

to significantly more challenging adaptation tasks for which there may not be an effective learning solution, at least one based on the family of kernel-based regularization algorithms.

Finally, the notion of discrepancy and its extensions similarly play a critical role in the analysis of learning with drifting distributions [22], a scenario that is closely related to those of domain adaption and sample bias correction.

Acknowledgments

We thank Steve Boyd, Michael Overton, and Katya Scheinberg for discussions about the optimization problem addressed in this work. We also thank Yishay Mansour and Afshin Rostamizadeh for previous discussions and collaborations on the topic of domain adaptation.

References

- [1] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira. Analysis of representations for domain adaptation. *Advances in Neural Information Processing Systems (NIPS)*, 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] S. Bickel, M. Brückner, and T. Scheffer. Discriminative learning under covariate shift. *Journal of Machine Learning Research*, 10:2137–2155, 2009.
- [4] J. Blitzer, M. Dredze, and F. Pereira. Biographies, Bollywood, Boom-boxes and

- Blenders: Domain Adaptation for Sentiment Classification. In *Proceedings of the Association for Computational Linguistics (ACL)*, 2007.
- [5] J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Wortman. Learning bounds for domain adaptation. *Advances in Neural Information Processing Systems (NIPS)*, 2008.
- [6] O. Bousquet and A. Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, 2002.
- [7] T. Cormen, C. Leiserson, and R. Rivest. *Introduction to Algorithms*. The MIT Press, 1992.
- [8] C. Cortes and M. Mohri. Domain adaptation in regression. In *ALT*, pages 308–323, 2011.
- [9] C. Cortes and V. Vapnik. Support-Vector Networks. *Machine Learning*, 20(3), 1995.
- [10] C. Cortes, M. Mohri, M. Riley, and A. Rostamizadeh. Sample selection bias correction theory. In *ALT*, pages 38–53, 2008.
- [11] C. Cortes, Y. Mansour, and M. Mohri. Learning bounds for importance weighting. In *Advances in Neural Information Processing Systems (NIPS)*, pages 442–450, 2010.
- [12] M. Dredze, J. Blitzer, P. P. Talukdar, K. Ganchev, J. Graca, and F. Pereira. Frustratingly Hard Domain Adaptation for Parsing. In *Proceedings of the Conference on Natural Language Learning (CoNLL)*, 2007.
- [13] M. Dredze, K. Crammer, and F. Pereira. Confidence-Weighted Linear Classification. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2008.
- [14] R. M. Dudley. Real Analysis and Probability. Wadsworth, Belmont, CA, 1989.
- [15] G. H. Golub and C. F. van Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 3rd edition, 1996.
- [16] J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, and B. Schölkopf. Correcting sample selection bias by unlabeled data. In *Advances in Neural Information Process*ing Systems (NIPS), volume 19, pages 601–608, 2006.
- [17] J. Jiang and C. Zhai. Instance Weighting for Domain Adaptation in NLP. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 264–271, 2007.
- [18] 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.
- [19] A. S. Lewis. The convex analysis of unitarily invariant matrix functions. *Journal of Convex Analysis*, 2(2):173–183, 1995.
- [20] Y. Mansour, M. Mohri, and A. Rostamizadeh. Domain adaptation: Learning bounds and algorithms. In *Proceedings of the Conference on Learning Theory (COLT)*, Montréal, Canada, 2009. Omnipress.
- [21] A. M. Martínez. Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class. *IEEE Transactions Pattern Analysis*, 24(6), 2002.
- [22] M. Mohri and A. Muñoz. New analysis and algorithm for learning with drifting distributions. In *Proceedings of ALT 2012*, volume 7568, pages 124–138, Lyon, France, 2012. Springer, Heidelberg, Germany.
- [23] 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.
- [24] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Springer, 2003.

- [25] Y. Nesterov. Smooth minimization of non-smooth functions. *Math. Program.*, 103: 127–152, May 2005.
- [26] Y. Nesterov. Smoothing technique and its applications in semidefinite optimization. *Math. Program.*, 110:245–259, 2007.
- [27] Y. Nesterov and A. Nemirovsky. *Interior Point Polynomial Methods in Convex Programming: Theory and Appl.* SIAM, 1994.
- [28] M. L. Overton. On minimizing the maximum eigenvalue of a symmetric matrix. *SIAM J. Matrix Anal. Appl.*, 9(2), 1988.
- [29] S. D. Pietra, V. D. Pietra, R. L. Mercer, and S. Roukos. Adaptive language modeling using minimum discriminant estimation. In *Proceedings of the workshop on Human Language Technologies (HLT)*, pages 103–106, 1992.
- [30] R. Rosenfeld. A Maximum Entropy Approach to Adaptive Statistical Language Modeling. *Computer Speech and Language*, 10:187–228, 1996.
- [31] C. Saunders, A. Gammerman, and V. Vovk. Ridge Regression Learning Algorithm in Dual Variables. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 515–521, 1998.
- [32] H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244, 2000.
- [33] I. Steinwart. On the influence of the kernel on the consistency of support vector machines. *Journal of Machine Learning Research*, 2:67–93, 2002.
- [34] M. Sugiyama, S. Nakajima, H. Kashima, P. von Bünau, and M. Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *Advances in Neural Information Processing Systems (NIPS)*, 2008.
- [35] M. Sugiyama, T. Suzuki, S. Nakajima, H. Kashima, P. von Bünau, and M. Kawanabe. Direct importance estimation for covariate shift adaptation. *Annals of the Institute of Statistical Mathematics*, 60:699–746, 2008.
- [36] V. N. Vapnik. Statistical Learning Theory. J. Wiley & Sons, 1998.
- [37] Y. Yu and C. Szepesvári. Analysis of kernel mean matching under covariate shift. In *ICML*, 2012.

Appendix

A Proof of Theorem 2

The theorem and its proof can be viewed as generalizations of known results for stability [6]. Stability guarantees provide a bound on the difference of loss between the hypotheses obtained by training on two samples differing by one point. Here, we seek a similar bound but for hypotheses trained on two *weighted* samples, each weighted according to a different distribution.

To do so, we first derive an upper bound on the norm of the difference of these hypotheses by using the (generalized) Bregman divergences of the objective functions they minimize. We use in particular the fact that the objective function admits a zero subgradient for a minimizing hypothesis. Next we show how that upper

bound can be analyzed in terms of the discrepancy and $\eta_H(f_P, f_Q)$. Finally, we use μ -admissibility to related the difference of loss to the norm of the difference of hypotheses.

Proof. The proof makes use of a generalized Bregman divergence, which we first introduce. For a convex function $F: \mathbb{H} \to \mathbb{R}$, we denote by $\partial F(h)$ the subgradient of F at $h: \partial F(h) = \{g \in \mathbb{H}: \forall h' \in \mathbb{H}, F(h') - F(h) \geq \langle h' - h, g \rangle\}$. $\partial F(h)$ 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 , $\partial (F_1 + F_2)(h) = \{g_1 + g_2 : g_1 \in \partial F_1(h), g_2 \in \partial F_2(h)\}$. 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:

$$\forall h', h \in \mathbb{H}, B_F(h'||h) = F(h') - F(h) - \langle h' - h, \delta F(h) \rangle. \tag{A.1}$$

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 $h \to \|h\|_K^2$. 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 $F_{(\widehat{Q},f_Q)}$ and $\widehat{R}_{(\widehat{Q},f_Q)}$ compatible so that $B_{F_{(\widehat{Q},f_Q)}} = B_{\widehat{R}_{(\widehat{Q},f_Q)}} + \lambda B_N$, we define $\delta \widehat{R}_{(\widehat{Q},f_Q)}$ from $\delta F_{(\widehat{Q},f_Q)}$ by: $\delta \widehat{R}_{(\widehat{Q},f_Q)}(h) = \delta F_{(\widehat{Q},f_Q)}(h) - \lambda \nabla N(h)$ for all $h \in \mathbb{H}$. Furthermore, we choose $\delta F_{(\widehat{Q},f_Q)}(h)$ to be 0 for any point h where $F_{(\widehat{Q},f_Q)}$ is minimal and let $\delta F_{(\widehat{Q},f_Q)}(h)$ be an arbitrary element of $\partial F_{(\widehat{Q},f_Q)}(h)$ for all other hs. We proceed in a similar way to define the Bregman divergences for $F_{(\widehat{P},f_P)}$ and $\widehat{R}_{(\widehat{P},f_P)}$ so that $B_{F_{(\widehat{P},f_P)}} = B_{\widehat{R}_{(\widehat{P},f_P)}} + \lambda B_N$.

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

$$B_{F_{(\widehat{Q},f_Q)}}(h'\|h) + B_{F_{(\widehat{P},f_P)}}(h\|h') \ge \lambda \Big(B_N(h'\|h) + B_N(h\|h')\Big).$$

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

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

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

$$2\lambda \|h' - h\|_K^2 \le \left(\mathcal{L}_{\widehat{P}}(h, f_P) - \mathcal{L}_{\widehat{O}}(h, f_Q)\right) - \left(\mathcal{L}_{\widehat{P}}(h', f_P) - \mathcal{L}_{\widehat{O}}(h', f_Q)\right).$$

Let h_0 be an arbitrary element of H. The right-hand side can be decomposed as follows in terms of h_0 :

$$2\lambda \|h' - h\|_{K}^{2} \leq \left(\mathcal{L}_{\widehat{P}}(h, f_{P}) - \mathcal{L}_{\widehat{P}}(h, h_{0})\right) - \left(\mathcal{L}_{\widehat{P}}(h', f_{P}) - \mathcal{L}_{\widehat{P}}(h', h_{0})\right) + \left(\mathcal{L}_{\widehat{P}}(h, h_{0}) - \mathcal{L}_{\widehat{Q}}(h, h_{0})\right) - \left(\mathcal{L}_{\widehat{P}}(h', h_{0}) - \mathcal{L}_{\widehat{Q}}(h', h_{0})\right) + \left(\mathcal{L}_{\widehat{Q}}(h, h_{0}) - \mathcal{L}_{\widehat{Q}}(h, f_{Q})\right) - \left(\mathcal{L}_{\widehat{Q}}(h', h_{0}) - \mathcal{L}_{\widehat{Q}}(h', f_{Q})\right).$$
(A.2)

By the μ -admissibility of the loss, the following inequalities hold:

$$\begin{split} & \left(\mathcal{L}_{\widehat{P}}(h, f_P) - \mathcal{L}_{\widehat{P}}(h, h_0) \right) - \left(\mathcal{L}_{\widehat{P}}(h', f_P) - \mathcal{L}_{\widehat{P}}(h', h_0) \right) \leq 2\mu \mathop{\mathbf{E}}_{x \sim \widehat{P}}[|f_P(x) - h_0(x)|] \\ & \left(\mathcal{L}_{\widehat{Q}}(h, h_0) - \mathcal{L}_{\widehat{Q}}(h, f_Q) \right) - \left(\mathcal{L}_{\widehat{Q}}(h', h_0) - \mathcal{L}_{\widehat{Q}}(h', f_Q) \right) \leq 2\mu \mathop{\mathbf{E}}_{x \sim \widehat{P}}[|f_Q(x) - h_0(x)|]. \end{split}$$

Since h_0 is in H, the other terms can be bounded in terms of the discrepancy:

$$\left(\mathcal{L}_{\widehat{P}}(h, h_0) - \mathcal{L}_{\widehat{Q}}(h, h_0)\right) - \left(\mathcal{L}_{\widehat{P}}(h', h_0) - \mathcal{L}_{\widehat{Q}}(h', h_0)\right) \le 2\operatorname{disc}(\widehat{P}, \widehat{Q}).$$

Thus,

$$2\lambda \|h' - h\|_{K}^{2} \leq 2 \left(\operatorname{disc}(\widehat{P}, \widehat{Q}) + \mu \mathop{\mathbb{E}}_{x \sim \widehat{P}}[|f_{P}(x) - h_{0}(x)|] + \mu \mathop{\mathbb{E}}_{x \sim \widehat{Q}}[|f_{Q}(x) - h_{0}(x)|]\right).$$

Since the inequality holds for all h_0 , we can write

$$\lambda \|h' - h\|_{K}^{2} \leq \operatorname{disc}(\widehat{P}, \widehat{Q}) + \mu \min_{h_{0} \in H} \Big\{ \max_{x \in \operatorname{supp}(\widehat{P})} |f_{P}(x) - h_{0}(x)| + \max_{x \in \operatorname{supp}(\widehat{Q})} |f_{Q}(x) - h_{0}(x)| \Big\}.$$

That is

$$2\lambda \|h' - h\|_K^2 \le 2\operatorname{disc}(\hat{P}, \hat{Q}) + 2\mu \eta_H(f_P, f_Q).$$
 (A.3)

By the reproducing property, for any $x \in \mathcal{X}$, $(h'-h)(x) = \langle h'-h, K(x,\cdot) \rangle$, thus, for any $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, $\left| L(h'(x),y) - L(h(x),y) \right| \leq \mu |h'(x) - h(x)| \leq \mu r \|h'-h\|_K$. Upper bounding the right-hand side using (A.3) directly yields the statement (9).

Note that the same proof can be used to derive a bound on the difference of the expected losses of the two hypotheses by using the following steps:

$$\begin{aligned} |\mathcal{L}_{P}(h', f_{P}) - \mathcal{L}_{P}(h, f_{P})| &\leq \mu \mathop{\mathbb{E}}_{x \sim P}[|h'(x) - h(x)|] \\ &\leq \mu \mathop{\mathbb{E}}_{x \sim P}[|\langle h'(x) - h(x), K(x, \cdot) \rangle|] \\ &\leq \mu \|h' - h\|_{K} \mathop{\mathbb{E}}_{x \sim P}\left[\sqrt{K(x, x)}\right]. \end{aligned}$$

The resulting upper bound only differs from that of the theorem by the expectation term $E_{x\sim P}\left[\sqrt{K(x,x)}\right]$ versus $\max_x \sqrt{K(x,x)}$. For a fixed kernel K, these are both constant terms and cannot be minimized.

B Proof of Theorem 3

Proof. We can proceed as in the proof of Theorem 2 and use inequality A.2. Thus, for any $h_0 \in H$, we can write:

$$2\lambda \|h' - h\|_{K}^{2} \leq \left(\mathcal{L}_{\widehat{P}}(h, f_{P}) - \mathcal{L}_{\widehat{P}}(h, h_{0})\right) - \left(\mathcal{L}_{\widehat{P}}(h', f_{P}) - \mathcal{L}_{\widehat{P}}(h', h_{0})\right) + \left(\mathcal{L}_{\widehat{P}}(h, h_{0}) - \mathcal{L}_{\widehat{Q}}(h, h_{0})\right) - \left(\mathcal{L}_{\widehat{P}}(h', h_{0}) - \mathcal{L}_{\widehat{Q}}(h', h_{0})\right) + \left(\mathcal{L}_{\widehat{Q}}(h, h_{0}) - \mathcal{L}_{\widehat{Q}}(h, f_{Q})\right) - \left(\mathcal{L}_{\widehat{Q}}(h', h_{0}) - \mathcal{L}_{\widehat{Q}}(h', f_{Q})\right).$$

Now, by definition of the squared loss, we can write:

$$\mathcal{L}_{\widehat{P}}(h, f_{P}) - \mathcal{L}_{\widehat{P}}(h, h_{0}) = \underset{x \sim \widehat{P}}{\mathbb{E}} \left[(h_{0}(x) - f_{P}(x))(2h(x) - f_{P}(x) - h_{0}(x)) \right]$$

$$\mathcal{L}_{\widehat{P}}(h', f_{P}) - \mathcal{L}_{\widehat{P}}(h', h_{0}) = \underset{x \sim \widehat{P}}{\mathbb{E}} \left[(h_{0}(x) - f_{P}(x))(2h'(x) - f_{P}(x) - h_{0}(x)) \right].$$

Taking the difference of these two equalities yields

$$\left(\mathcal{L}_{\widehat{P}}(h, f_P) - \mathcal{L}_{\widehat{P}}(h, h_0)\right) - \left(\mathcal{L}_{\widehat{P}}(h', f) - \mathcal{L}_{\widehat{P}}(h', h_0)\right) \\
= 2 \mathop{\mathbb{E}}_{x \sim \widehat{P}}\left[\left(h_0(x) - f_P(x)\right)(h(x) - h'(x))\right].$$

Similarly, we obtain:

$$\left(\mathcal{L}_{\widehat{Q}}(h, h_0) - \mathcal{L}_{\widehat{Q}}(h, f_Q)\right) - \left(\mathcal{L}_{\widehat{Q}}(h', h_0) - \mathcal{L}_{\widehat{Q}}(h', f_Q)\right) \\
= -2 \mathop{\mathrm{E}}_{x \sim \widehat{Q}} \left[(h_0(x) - f_Q(x))(h(x) - h'(x)) \right].$$

Since h_0 is in H, by definition of the discrepancy, the following holds:

$$\left(\mathcal{L}_{\widehat{P}}(h, h_0) - \mathcal{L}_{\widehat{O}}(h, h_0)\right) - \left(\mathcal{L}_{\widehat{P}}(h', h_0) - \mathcal{L}_{\widehat{O}}(h', h_0)\right) \le 2\operatorname{disc}(\widehat{P}, \widehat{Q}).$$

Thus, we have $2\lambda \|h' - h\|_K^2 \le 2\operatorname{disc}(\widehat{P}, \widehat{Q}) + 2\Delta$, where

$$\Delta = \mathop{\mathbf{E}}_{x \sim \widehat{P}} \left[(h_0(x) - f_P(x))(h(x) - h'(x)) \right] - \mathop{\mathbf{E}}_{x \sim \widehat{Q}} \left[(h_0(x) - f_Q(x))(h(x) - h'(x)) \right].$$

By the reproducing property, the identity $h(x) - h'(x) = \langle h - h', K(x, \cdot) \rangle_K$ holds for any $x \in \mathcal{X}$. In view of that, Δ can be expressed and bounded as follows:

$$\Delta = \left\langle h - h', \mathop{\mathbb{E}}_{x \sim \widehat{P}}[(h_0(x) - f_P(x))K(x, \cdot)] - \mathop{\mathbb{E}}_{x \sim \widehat{Q}}[(h_0(x) - f_Q(x))K(x, \cdot)] \right\rangle$$

$$\leq \|h - h'\|_K \left\| \mathop{\mathbb{E}}_{x \sim \widehat{P}}[(h_0(x) - f_P(x))K(x, \cdot)] - \mathop{\mathbb{E}}_{x \sim \widehat{Q}}[(h_0(x) - f_Q(x))K(x, \cdot)] \right\|_K.$$

Since the inequality holds for all $h_0 \in H$, we can write $\Delta \leq \|h - h'\|_K \delta_H(f_P, f_Q)$. Thus, we can write

$$2\lambda \|h' - h\|_K^2 \le 2\operatorname{disc}(\widehat{P}, \widehat{Q}) + 2\|h - h'\|_K \delta_H(f_P, f_Q).$$

Solving the second-order inequality for $||h' - h||_K$ yields the inequality

$$||h' - h||_K \le \frac{1}{2\lambda} \left(\delta_H(f_P, f_Q) + \sqrt{\delta_H(f_P, f_Q)^2 + 4\lambda \operatorname{disc}(\widehat{P}, \widehat{Q})} \right).$$
 (B.1)

For any $(x, y) \in \mathcal{X} \times \mathcal{Y}$, using the definition of the squared loss and the reproducing property, we can write

$$|L(h'(x), y) - L(h(x), y)| = |(h'(x) - y)^{2} - (h(x) - y)^{2}|$$

$$= |(h'(x) - h(x))(h'(x) - y + h(x) - y)|$$

$$\leq 2\sqrt{M}|h'(x) - h(x)|$$

$$= 2\sqrt{M}|\langle h' - h, K(x, \cdot) \rangle| \leq 2\sqrt{M}r||h' - h||_{K}.$$
(B.2)

Upper bounding $||h' - h||_K$ using (B.1) yields the statement of theorem.

C Proof of Theorem 4

Proof. The proof of the theorem is similar to that of Theorem 2 modulo the use of inequality (B.2): instead of bounding |L(h'(x),y)-L(h(x),y)| for all $(x,y)\in \mathcal{X}\times\mathcal{Y}$, here, we seek to upper bound $\sqrt{\mathcal{L}_P(h',f)}-\sqrt{\mathcal{L}_P(h,f)}$. Our analysis is based on the following:

$$\begin{aligned} \left| \mathcal{L}_{P}(h',f) - \mathcal{L}_{P}(h,f) \right| &= \left| \underset{x \sim P}{\mathbb{E}} [L(h'(x),f(x))] - \underset{x \sim P}{\mathbb{E}} [L(h(x),f(x))] \right| \\ &= \left| \underset{x \sim P}{\mathbb{E}} [(h'(x) - f(x))^{2} - (h(x) - f(x))^{2}] \right| \\ &= \left| \underset{x \sim P}{\mathbb{E}} [(h'(x) - h(x))(h'(x) - f(x) + h(x) - f(x))] \right| \\ &\leq \underset{x \sim P}{\mathbb{E}} [|(h'(x) - h(x))(h'(x) - f(x) + h(x) - f(x))|] \\ &\leq \|h' - h\|_{\infty} \underset{x \sim P}{\mathbb{E}} [|(h'(x) - f(x) + h(x) - f(x))|]. \end{aligned}$$

The first factor in the right-hand side, $||h'-h||_{\infty}$, can be bounded by $r||h'-h||_{K}$ since, by the reproducing property, for all $x \in \mathcal{X}$, $(h'-h)(x) = \langle h'-h, K(x,\cdot) \rangle$. The second factor can be bounded as follows using Jensen's inequality:

$$\begin{split} & \underset{x \sim P}{\mathbb{E}}[|(h'(x) - f(x) + h(x) - f(x))|] \\ & \leq \underset{x \sim P}{\mathbb{E}}[|(h'(x) - f(x))|] + \underset{x \sim P}{\mathbb{E}}[|h(x) - f(x))|] \\ & \leq \sqrt{\underset{x \sim P}{\mathbb{E}}[|(h'(x) - f(x))|^2]} + \sqrt{\underset{x \sim P}{\mathbb{E}}[|(h(x) - f(x))|^2]} \\ & = \sqrt{\mathcal{L}_P(h', f)} + \sqrt{\mathcal{L}_P(h, f)}. \end{split}$$

Thus, (C.1) becomes

$$\left| \mathcal{L}_P(h',f) - \mathcal{L}_P(h,f) \right| \le r \|h' - h\|_K \left(\sqrt{\mathcal{L}_P(h',f)} + \sqrt{\mathcal{L}_P(h,f)} \right).$$

Dividing both sides by $\left(\sqrt{\mathcal{L}_P(h',f)} + \sqrt{\mathcal{L}_P(h,f)}\right)$ gives

$$\left| \sqrt{\mathcal{L}_P(h',f)} - \sqrt{\mathcal{L}_P(h,f)} \right| \le r \|h' - h\|_K.$$

Using (B.1) to upper bound $||h'-h||_K$ as in the proof of Theorem 2 concludes the proof.

D Proof of Theorem 7

Proof. Let $\|\mathbf{M}^*\|_2$ be the optimum of the SDP (23), $G_p(\mathbf{M}'^*)$ 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 (24), \mathbf{z}^* is a solution of the original SDP (23) with relative accuracy ϵ :

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

 G_p admits 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 6, $\frac{G_p(\mathbf{M}(z_k))-G_p(\mathbf{M}'^*)}{G_p(\mathbf{M}'^*)}\leq \frac{4(2p-1)r}{(k+1)(k+2)}$. Choosing p such that $2p<4\frac{(1+\epsilon)\log r}{6}$ gives

$$\frac{G_p(\mathbf{M}(z_k)) - G_p(\mathbf{M}'^*)}{G_p(\mathbf{M}'^*)} \le \frac{16r(1+\epsilon)\log r}{(k+1)(k+2)\epsilon}.$$

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:

$$k^* = \sqrt{(16r(1+\epsilon)\log r)/\epsilon^2} = 4\sqrt{(1+\epsilon)r\log r}/\epsilon.$$

This concludes the proof.