Domain Adaptation from a PAC-Bayesian Random Features perspective

Master 1 MIASHS - (2022 - 2023)

Julien Bastian
UFR ASSP
Université de Lyon, Université Lumière Lyon 2

Stage réalisé au sein du Laboratoire ERIC (Lyon) et du Laboratoire Hubert Curien (St-Etienne)
Encadrants : Guillaume Metzler, Emilie Morvant & Marie-Ange Lèbre
Tuteur enseignant : Stéphane Chrétien

Contents

Abstract

Building upon recent developments in PAC-Bayesian theory for Domain Adaptation, our analysis emphasizes kernel methods, particularly the Random Fourier Features (RFF). Drawing inspiration from prior works, we interpret the pseudo-kernel, obtained via RFF, as a predictor within the PAC-Bayesian context in the specific case of Unsupervised Domain Adaptation. Thanks to this interpretation we propose a new PAC-Bayesian bound on the representation in the target domain associated with the distribution of the RFF of a pseudo kernel. We contend that the optimal distribution of these features translates to better data representation enabling classification in the target domain even in the absence of labels. We propose to learn a Support Vector Machine using the obtained data representation to perform classification on the target domain. We evaluate this algorithm and compare it with other protocols designed for Domain Adaptation in the PAC-Bayesian framework to assess its interest.

1 Introduction

While showing outstanding results lately classical machine learning relies on a strong assumptions. One of them is that the learning data and the data on which the learned model will be used come from a single unknown distribution. Therefore to evaluate the performance of the model when faced with unseen instances it is common to consider the performance on a test set drawn from this distribution to be a good estimator. However, there are many cases where the assumption of the data coming from a single unknown distribution does not hold in practice. Think for a example of supervised classification model trained on socio-economic descriptor from the french population that would ultimately be used on the Brazilian population. The performance on the french domain would not be representative of those on the Brazilian domain. Indeed, the same features could have widely different meaning. For example geographic information would bear much different meaning, the fact of living near the Amazonian forest might be very important as a descriptor in the Brazilian population while it makes absolutely no sense in France. The same could be said about living or not in the capital city since both countries have different system of centralization. The performance on a test set drawn from the french population would give little to no information about the true performance on the model, *i.e.*, its performance on the Brazilian domain.
In this kind of situation it is necessary to adapt the model learned using source data to be used on new and different target data. Formally we differentiate between two distributions also called *domains*, a source domain and a target domain, which puts us in the *domain adaptation* framework. More precisely we consider in this work *unsupervised domain adaptation*, in the case of binary classification, meaning that we have access to the label we want to predict for the source domain but not for the target one.
This is a challenging task that requires specific tools and methods, more so it is especially hard to get guarantees on the performance of a model. These guarantees usually take form of upper bound on the expected risk, also called generalization bounds, over the target domain, which quantifies the expected error on unseen data. In traditional

machine learning the guarantees usually rely heavily on the assumption that the average training error is a good estimator of the expected risk, since both measure regards data coming from a same distribution. However when the distribution of the labeled data used to compute the average training (source) and of the unseen data for which the expected risk is considered (target) differs the relation does not hold anymore. It is therefore necessary to employ specific tools to build these bounds.
Our choice of considering expected risk bounds is not only motivated by the development of theoretical guarantees. Indeed we aim to ultimately learn in a way that minimizes the generalization error, the final model will then be trust-able.

The kind of theoretical guarantees we consider in this work originate from the PAC-Bayesian theory that combines the Probably Approximately Correct (PAC) learning standpoint and Bayesian inference. In this framework the model is defined as a combination of hypotheses according to a probability distribution. In comparison with the classical generalization bounds the PAC-Bayesian ones are more flexible, can incorporate prior knowledge and are convenient to minimize by finding an optimal distribution over the hypotheses. For this work we consider specifically PAC-Bayesian developments for Domain Adaptation from [Germain et al., 2020].

The analysis we develop is focused on kernel methods [Hofmann et al., 2008] and especially on Random Fourier Features (RFF) that let one approximate a kernel by combining trigonometric hypothesis according to its Fourier transform. We use the work of [Letarte et al., 2019] who shown that the pseudo kernel obtained by combining the RFF according to an arbitrary distribution can be interpreted as a predictor and therefore be analyzed by the PAC-Bayesian framework. For binary classification the Fourier features are considered as hypotheses on the belonging of two points to the same class. The optimal distribution of these features will then be the one leading to the best representation of the data for classification. The aim of the present work is to extend this interpretation of kernel approximation and representation learning through the PAC-Bayesian theory for the case of Domain Adaptation. We therefore develop new bounds on the representation of the data points that lead to a new space of representation suitable for the task at hand.

2 Generalities

2.1 Notations and Setting

We consider binary classification, with $\mathcal{X} \subseteq \mathbb{R}^d$ the input space and $\mathcal{Y} = \{-1, +1\}$ the output set. We define an unknown distribution, or domain, $\mathcal{D}$ on $\mathcal{X} \times \mathcal{Y}$ with $(\mathbf{x}, y) \in \mathcal{D}$ an instance drawn according to this distribution. More generally we have a learning sample S of n instance drawn *iid* from $\mathcal{D}$ as $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^n \underset{iid}{\sim} \mathcal{D}$. An other necessary tool for the learning a procedure is a space of hypotheses $\mathcal{H}$ that contains classifiers h of the form $h : \mathcal{X} \longrightarrow [-1, +1]$, that we call *real-valued voters*, since there output can be any real value in the interval.

Therefore given the learning sample S and the hypotheses space $\mathcal{H}$ the objective is to

find the classifier $h \in \mathcal{H}$ giving the least classification errors on the unknown distribution $\mathcal{D}$. For this it is necessary to define a measure to assess the quality of classification called a loss function ℓ In the classification setting some functions are classical, we present here two : the $0-1$ loss and the linear loss as illustrated in Figure 1. The $0-1$ loss is defined as :

$$\ell_{0-1}(h(\mathbf{x}), y) = \mathbb{1}[sign(h(\mathbf{x})) \neq y],$$

where

$$\ell_{0-1} : [-1, +1] \times \mathcal{Y} \longrightarrow 0, 1$$

with $sign(c) = -1$ if $c < 0$, $sign(c) = +1$ if $c > 0$. It can also be computed in a more convenient way by considering, $\ell_{0-1}(h(\mathbf{x}), y) = y\ sign(h(\mathbf{x}))y$. Note that, as in the definition above, when $h(\cdot)$ does not output binary value one can take the sign of the prediction to compute the $0-1$ loss. However, in this case the linear loss, that measure the amount of error for each prediction is more appropriate, defined as :

$$\ell_{lin}(h(\mathbf{x}), y) = \frac{1}{2}(1 - h(\mathbf{x}) \cdot y).$$

where

$$\ell_{lin} : [-1, +1] \times \mathcal{Y} \longrightarrow 0, 1$$

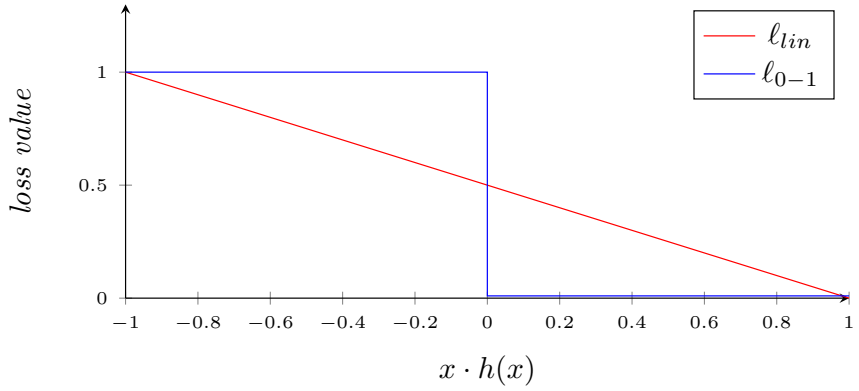


Figure 1: The linear loss ℓ_{lin} and the 0-1 loss ℓ_{0-1}

To extend the measure to the distribution $\mathcal{D}$ or the learning sample S we consider the risk associated with a loss. In the case of S we call it the empirical risk and for $\mathcal{D}$ the true risk. Given a loss ℓ, a domain $\mathcal{D}$, a hypotheses space $\mathcal{H}$, we have the true risk for all $h \in \mathcal{H}$:

$$R_{\mathcal{D}}(h) := \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{D}} \ell(h(\mathbf{x}), y).$$

Then, given loss ℓ, a training sample S of size n, a hypotheses space $\mathcal{H}$, we define the

empirical risk for all $h \in \mathcal{H}$:

$$\widehat{R}_S(h) := \frac{1}{n} \sum_{i=1}^{n} \ell(h(\mathbf{x}^i), y^i). \tag{1}$$

2.2 Finding the best classifier

Empirical Risk Minimization

From the previous definitions the best classifier $h^* \in \mathcal{H}$, is the one solving the **true risk minimization** for $\mathcal{D}$, such that,

$$h^* = \arg\min_{h \in \mathcal{H}} R_{\mathcal{D}}(h).$$

However the only information available during the learning process is the empirical risk $\widehat{R}_S(h)$ over the sample S. A classical approach is then to consider the empirical risk as a good estimator of the true one, therefore minimizing the former should amount to minimize the latter. The learning process consists then of the **empirical risk minimization**, defined for a sample S as,

$$h^* = \arg\min_{h \in \mathcal{H}} \widehat{R_S}(h).$$

However, since the learning sample may not be a perfect representation of the unknown domain $\mathcal{D}$, trusting it too much during the learning process may lead to numerous issues. For example, the case of over-fitting where by being to adapted to the training problem the classifier will perform significantly worse on new data points compared with known, train data points, i.e it will poorly generalize.

From there two important problems arise. *How to assess the true performance of the classifier ? How to learn with good guarantees on the minimization of the true risk ?.*

Generalization guarantees

To answer first issue it is possible to consider *generalization guarantees*, usually taking the form of *generalization bounds*. They generally take the form of Probably Approximately Corrects (PAC) bounds [Vapnik and Chervonenkis, 1982] [Valiant, 1984] [Haussler and Warmuth, 2018], which consider the divergence between *true* and *empirical* risk :

$$\mathbf{P}\left(\left|R_{\mathcal{D}}(h) - \widehat{R_{\mathcal{D}}}(h)\right| \leq \epsilon\right) \geq 1 - \delta,$$

with $\epsilon > 0$ and $\delta \in [0, 1]$. Or in a form giving a bound directly over the true risk $R_{\mathcal{D}}(h)$, with ϵ and δ in the same ranges, we can formulate the PAC inequality as

$$\mathbf{P}\left(R_{\mathcal{D}}(h) \leq \widehat{R_{\mathcal{D}}}(h) + \epsilon\right) \geq 1 - \delta \tag{2}$$

However those bounds can take different forms and be developed in various of ways, including but not limited to, VC-Dimension based bounds [Vapnik, 1999] [Blumer et al., 1989], Rademacher complexity-based bounds [Bartlett and Mendelson, 2002] [Koltchinskii and Panchenko, 2000], Uniform stability based bounds [Bousquet and Elisseeff, 2000] [Bousquet and Elisseeff, 2002] and PAC-Bayesian bounds [Guedj, 2019] [Alquier, 2023]. In the present study we consider the PAC-Bayesian framework introduced by [McAllester, 1998] and [Shawe-Taylor and Williamson, 1997]. This framework has the advantage of providing flexible, tight bounds and of providing a way of answering our second question by letting us find the final classifier minimizing the bounded *true risk*.

2.3 The PAC-Bayesian standpoint

2.3.1 Specificity and tools

Before diving into the PAC-Bayesian bounds we need to highlight what makes it special and define the important tools to work within this framework. In particular the final model is not anymore an unique classifier $h \in \mathcal{H}$ but a combination of all hypotheses in the hypotheses space according to a probability distribution Q. We start from a *prior distribution* P and the goal is to find the *posterior distribution* Q leading to the best classifier, in a *Bayesian* fashion. This prior distribution P can be of two types. The first type is *informed* [Lever et al., 2010] [Lever et al., 2013], meaning that it incorporates information about the problem at hand, this can be done, for example, via a sort of pre training or by taking into consideration the learning scenario. We can say it incorporates *prior knowledge* to stay within the border of the Bayesian vocabulary. In the other case which is the most widely spread due to its convenience it is, well, *not informed*.

Every $h \in \mathcal{H}$ can be considered as a predictor, or voter, but what matters is the combination of these predictors. This framework is very versatile and can be used for a wide variety of learning scenario, ranging from Support Vector Machines [Valiant, 1984] [Ambroladze et al., 2006] (SVM) to neural networks [Liao et al., 2020] [Perez-Ortiz et al., 2021] [Rivasplata et al., 2019], even linear regression can be interpreted this way [Germain et al., 2017] if one want to step outside of classification.

The learning process is thus focused on finding probability distribution giving rise to best classifier. A classical predictor to consider is the Bayes classifier or the majority vote, for a distribution Q over $\mathcal{H}$ it takes the following form,

$$B_Q(\mathbf{x}) = sign\left(\mathop{\mathbf{E}}_{h\sim Q} h(\mathbf{x})\right). \tag{3}$$

To this Bayes classifier B_Q associated with a distribution Q, we can define the *Bayes risk* over a unknown data distribution $\mathcal{D}$ as :

$$R_{\mathcal{D}}(B_Q) := \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{D}} \ell(B_Q(\mathbf{x}), y).$$

Since the output of B_Q is binary it is equivalent to use the 0-1 loss or the linear loss. More precisely the latter one is equal to the former one. Furthermore the Bayes Classifier we consider also the Gibbs Classifier associated with Q that we denote G_Q. It is a stochastic classifier in the sense that it draws a hypothesis $h \in \mathcal{H}$ according to Q and use it to make its prediction. This classifier can make different prediction for a same $\mathbf{x} \sim \mathcal{D}$ when the output of B_Q will always be the same. The risk associated with this stochastic classifier for real-valued voters is defined as,

$$\begin{aligned} R_{\mathcal{D}}(G_Q) &:= \mathop{\mathbf{E}}_{h\sim Q} R_{\mathcal{D}}(h) \\ &= \mathop{\mathbf{E}}_{h\sim Q} \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{D}} \ell_{lin}(h(\mathbf{x}), y) \\ &= \frac{1}{2}\left(1 - \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{D}} \mathop{\mathbf{E}}_{h\sim Q} h(\mathbf{x}) \cdot y\right). \end{aligned} \tag{4}$$

An important results from the PAC-Bayesian literature is that the risk of the Bayes classifier is bounded by twice the risk of the Gibbs classifier [McAllester, 2003] [Langford and Shawe-Taylor, 2002],

$$R_{\mathcal{D}}(B_Q) \leq 2 \cdot R_{\mathcal{D}}(G_Q).$$

Proof. *In appendix*

This is very useful since the second is much more convenient to manipulate. Therefore the upper bounds constructed via the PAC-Bayesian theory usually concern the Gibbs risk gives with the advantage that it immediately upper bounds the Bayes risk which is the one on which we ultimately want to have results.

2.3.2 Margin of the Bayes classifier

As studied in [Germain et al., 2015], an important tool to develop pertinent analysis is the random variable M_Q^D that outputs for a specific $(\mathbf{x}, y) \in \mathcal{D}$ the margin of Bayes classifier B_Q,

$$M_Q(\mathbf{x}, y) = \mathop{\mathbf{E}}_{h\sim Q} y \cdot h(\mathbf{x}). \tag{5}$$

Specifically we are interested in the moments of this random variable. We start with its first moment $\mu_1(M_Q^D)$ that let us relate the Gibbs risk with the margin of the majority

vote,

$$\begin{aligned}\mu_1(M_Q^D) &= \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{D}} M_Q(\mathbf{x},y) \\ &= \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{D}} \mathop{\mathbf{E}}_{h\sim Q} y\cdot h(\mathbf{x}).\end{aligned}$$

Interestingly it is let us rewrite the Gibbs risk $R(G_Q)$ with a different point of view than its original definition in Equation 4

$$R(G_Q) = \left(1 - \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{D}} \mathop{\mathbf{E}}_{h\sim Q} h(\mathbf{x})\cdot y\right) \tag{6}$$

$$= \left(1 - \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{D}} M_Q(\mathbf{x},y)\right) \tag{7}$$

$$= \left(\, 1 - \mu_1(M_Q^D)\,\right). \tag{8}$$

Then we define the second moment of M_Q^D that will allow useful useful interpretation latter, we design it $\mu_2(M_Q^D)$ and we have,

$$\begin{aligned}\mu_2(M_Q^D) &= \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{D}} \left[M_Q(\mathbf{x},y)\right]^2 \\ &= \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{D}} y^2 \left[\mathop{\mathbf{E}}_{h\sim Q} h(\mathbf{x})\right]^2 \\ &= \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{D}_\mathcal{X}} \left[\mathop{\mathbf{E}}_{h\sim Q} h(\mathbf{x})\right]^2.\end{aligned} \tag{9}$$

2.3.3 Joint error and disagreement

To allow a useful decomposition of the risk we recall the notions of*expected disagreement* and *expected joint error*. For real-valued voters the first one considers the expected amount by which the outputs of the voters are similar. For a domain $\mathcal{D}$, its marginal distribution $\mathcal{D}_\mathcal{X}$ and any probability distribution Q over the hypotheses space $\mathcal{H}$, the expected disagreement $d_{\mathcal{D}_\mathcal{X}}(Q)$ is defined as,

$$\begin{aligned}d_{\mathcal{D}_\mathcal{X}}(Q) &= \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{D}_\mathcal{X}} \mathop{\mathbf{E}}_{h\sim Q} \mathop{\mathbf{E}}_{h'\sim Q} \ell_{lin}(h(\mathbf{x}),h'(\mathbf{x})) \\ &= \frac{1}{2}\left(1 - \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{D}_\mathcal{X}} \mathop{\mathbf{E}}_{h\sim Q} \mathop{\mathbf{E}}_{h'\sim Q} h(\mathbf{x})\cdot h'(\mathbf{x})\right) \\ &= \frac{1}{2}\left(1 - \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{D}_\mathcal{X}} \left[\mathop{\mathbf{E}}_{h\sim Q} h(\mathbf{x})\right]^2\right).\end{aligned}$$

On the other hand for a domain $\mathcal{D}$ and any probability distribution Q over the hypotheses space $\mathcal{H}$, the expected joint error $e_\mathcal{D}(Q)$ gives a measure of the expected ability of two voters h, h' to make the same prediction when predicting on an instance $(\mathbf{x}, y) \sim \mathcal{D}$ is

defined as,

$$e_{\mathcal{D}}(Q) = \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{D}} \mathop{\mathbf{E}}_{h\sim Q} \mathop{\mathbf{E}}_{h'\sim Q} \ell_{lin}(h(\mathbf{x}), y) \cdot \ell_{lin}(h'(\mathbf{x}), y).$$

[Germain et al., 2015] observed that by using this quantities it is possible to decompose the Gibbs risk in such a manner that it is entirely represented by half the expected disagreement and the expected joint error such that, for a distribution Q over $\mathcal{H}$ and a domain $\mathcal{D}$, we have

$$R_{\mathcal{D}}(G_Q) = \frac{1}{2} d_{\mathcal{D}_\mathcal{X}}(Q) + e_{\mathcal{D}}(Q) \tag{10}$$

By defining the empirical disagreement and joint error for a distribution Q over $\mathcal{H}$ and a sample $S \sim \mathcal{D}^n$ of size n as,

$$\widehat{d_{D_\mathcal{X}}}(Q) = \frac{1}{n} \sum_{i=1}^{n} \mathop{\mathbf{E}}_{h\sim Q} \mathop{\mathbf{E}}_{h'\sim Q} \ell_{lin}(h(\mathbf{x}), h'(\mathbf{x})) \tag{11}$$

and

$$\widehat{e_S}(Q) = \frac{1}{n} \sum_{i=1}^{n} \mathop{\mathbf{E}}_{h\sim Q} \mathop{\mathbf{E}}_{h'\sim Q} \ell_{lin}(h(\mathbf{x}), y) \cdot \ell_{lin}(h'(\mathbf{x}), y). \tag{12}$$

We get the following distribution of the empirical risk $\widehat{R_S}(G_Q) = \mathbf{E}_{h\sim Q}\, \widehat{R_S}(h)$ similar to Equation (10),

$$\widehat{R_S}(G_Q) = \frac{1}{2} \widehat{d_{D_\mathcal{X}}}(Q) + \widehat{e_S}(Q).$$

We consider in the next subsection the PAC-Bayesian theorem that originates from [McAllester, 1998].

2.3.4 Bounds

The PAC-Bayesian generalization bounds are usually dependant on two main quantities : the empirical risk of the Gibbs classifier $\widehat{R_S}(G_Q)$, and the Kullback-Leibler (KL) divergence. The KL-divergence is defined for two probability distributions Q and P,

$$\mathrm{KL}(Q||P) = \mathop{\mathbf{E}}_{h\sim Q} \frac{P(h)}{Q(h)}.$$

It is a divergence metric that measures how much the distribution Q is different from the second one P. By considering Q and P as the posterior and prior distribution associated with the problem at hand it will act let us evaluate how much the posterior distribution diverges from the prior distribution.

Thereafter the PAC-Bayesian bound is presented as trade-off between the empirical Gibbs risk, acting as an estimator of the real Gibbs risk $R_{\mathcal{D}}(G_Q)$, and the KL-divergence between the prior and the posterior distribution. We recall below the PAC-Bayesian bound as formulated in [Germain et al., 2009b].

Theorem 1 ([Germain et al., 2009b]). *For any domain $\mathcal{D}$, any hypotheses space $\mathcal{H}$, any prior distribution P over $\mathcal{H}$, any $\epsilon \in (0,1]$, and any real number $c > 0$, with at least probability $1-\epsilon$ over the random choice $S \sim \mathcal{D}^n$ of size n, for every posterior distribution Q over $\mathcal{H}$, we have*

$$R_{\mathcal{D}}(G_Q) \leq \frac{1}{1-e^{-c}} \left[c \cdot \widehat{R_S}(G_Q) + \frac{1}{n} \left[KL(Q||P) + ln\frac{1}{\epsilon} \right] \right]$$

Note that the value of hyperparameter c that controls the trade-off between empirical risk and KL-divergence Moreover, if we formulate it a bit differently, for any domain $\mathcal{D}$, any set of hypotheses $\mathcal{H}$, any prior distribution P over $\mathcal{H}$, any $\epsilon \in (0,1]$, and any real number $c > 0$, we have

$$\underset{S\sim\mathcal{D}^n}{\mathbf{P}} \left(\forall\, Q \text{ on } \mathcal{H} : R_{\mathcal{D}}(G_Q) \leq \frac{1}{1-e^{-c}} \left[c \cdot \widehat{R_S}(G_Q) + \frac{1}{n} \left[\mathrm{KL}(Q||P) + ln\frac{1}{\epsilon} \right] \right] \right) \leq 1-\epsilon,$$

which is strictly equivalent to Theorem 1 but in a form similar to Equation (2).

2.4 Kernel methods and Random Fourier Features.

In the present work we focus on Kernel methods and in particular the PAC-Bayesian interpretation of their approximation with Random Fourier Features (RFF) introduced by [Rahimi and Recht, 2007].

2.4.1 Kernels and kernel methods

For all the following and whenever we consider kernels, we suppose $\mathcal{X}$ to be a compact subset of $\mathbb{R}$. A kernel function takes as input two feature vector and output a real coordinate,

$$k : \mathcal{X} \times \mathcal{X} \longrightarrow \mathbb{R},\ (\mathbf{x}, \mathbf{x}') \longrightarrow k(\mathbf{x}, \mathbf{x}').$$

This kernel has to verify two condition. First it must be symmetric meaning that $\forall (\mathbf{x}, \mathbf{x}') \in \mathcal{X} \times \mathcal{X}$ we have,

$$k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}', \mathbf{x}).$$

Second, it must be positive semi definite, so for any set $\{\mathbf{x}_i\}_{i=1}^n \in \mathcal{X}$ of size n and $\forall c \in \mathbb{R}^d$ we have,

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j k(\mathbf{x}_i, \mathbf{x}_j) \geq 0.$$

The above properties are necessary to apply the following theorem, known as the Mercer theorem [Mercer, 1909].

Theorem 2 (Mercer theorem [Mercer, 1909]). *Let $\mathcal{X}$ be a compact subset of $\mathbb{R}^d$ and k a positive semi-definite, symmetric and continuous function. Then there is an orthonormal basis $\{\phi_i\}_{i\in\mathbb{N}^*}$ of $L^2(\mathcal{X})$ and a nonnegative sequence $\{\lambda_i\}_{i\in\mathbb{N}^*}$, such that k has the*

representation :

$$k(x, x') = \sum_{i=1}^{\infty} \lambda_i \phi_i(\mathbf{x}) \phi_i(\mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}) \rangle$$

with $\phi(\mathbf{x}) = (\sqrt{\lambda_1}\phi_1(\mathbf{x}), \ldots, \sqrt{\lambda_i}\phi_i(\mathbf{x}), \ldots)$.

This theorem allows the representation of a kernel's output as the inner product of the transformation of the two input instances $(\mathbf{x}, \mathbf{x}') \in \mathcal{X}^2$ through a function ϕ. This function is called a feature map, as it maps a data point from $\mathcal{X}$ into a new feature space $\mathcal{Z}$,

$$\phi : \mathcal{X} \longrightarrow \mathcal{Z},$$

with the condition of $\mathcal{Z}$ being an inner product space, i.e. a space equipped with an inner product. This new representation can be very high dimensional or infinite dimensional and enable a better representation of the data. For example, in a classification task where the classes aren't linearly separable in the initial feature space they might be after the mapping in $\mathcal{Z}$.

Kernel trick

The strength of kernels is apparent in learning problems that rely on the inner product between instances. Indeed in this case, the following representation of the kernel output allowed by Mercer Theorem,

$$k(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle, \tag{13}$$

(with $\langle \cdot, \cdot \rangle$ the inner product) lets the learning be done in the feature space $\mathcal{Z}$ without having to compute the explicit coordinates of the data in that space. Only the kernel output is required to access all the necessary information, this is called the *kernel trick*. This trick has been used to "kernelize" a numerous learning protocols, the most famous one being Support Vector Machines [Valiant, 1984] but we can also cite ridge regression [McDonald, 2009] [Vovk, 2013], Principal Component Analysis (PCA) [Wold et al., 1987] [Schölkopf et al., 1997] or Gaussian processes [Rasmussen, 2003].

Different kernel functions can be used, each allowing to tackle a learning problem in a different feature space. Some of the most popular ones are,

- the polynomial kernel, for arbitrary polynomial order p :

$$k(\mathbf{x}, \mathbf{x}') = (\langle \mathbf{x}, \mathbf{x}' \rangle + b)^p$$

- the linear kernel,

$$k(\mathbf{x}, \mathbf{x}') = \langle \mathbf{x}, \mathbf{x}' \rangle,$$

- the Gaussian, or Radial Basis Function(RBF), kernel,

$$k(\mathbf{x}, \mathbf{x}') = e^{\left(-\gamma||\,\mathbf{x}-\mathbf{x}'\,||^2\right)} \tag{14}$$

  with γ a dilation hyper-parameter to tune.

Once a kernel function as been chosen and a learning sample $S \sim \mathcal{D}^n$ of size n as been drawn, it is possible to build the associated Gram matrix $\boldsymbol{K}$ of size $n \times n$ that contains the values of every couple of instances,

$$\boldsymbol{K} = (k(\mathbf{x}_i, \mathbf{x}_j))_{i,j}.$$

This matrix is different of a classical data matrix $\boldsymbol{X}$ of size $n \times d$ with d the dimension of the features, which associate to every instance the value of all of its features.

2.4.2 Random Fourier Features

The Gram matrix defined just before can become very expensive in both time and memory to deal with when the number of data points grows. This issue is one of the drawback from kernel methods and it is to answer it that the methods of Random Fourier Features (RFF) have been introduced by [Rahimi and Recht, 2007]. The idea is to approximate the value of a shift invariant kernels using properties of its Fourier transform. We denote $\boldsymbol{\delta} = \mathbf{x} - \mathbf{x}'$ the value of the difference between two data points from $\mathcal{X}$, a shift invariant kernel as therefore to verify,

$$k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x} - \mathbf{x}') = k(\boldsymbol{\delta}).$$

Put into words the kernel outputs only depends on the difference of the instances.

The idea is to take the Fourier transform of $k(\boldsymbol{\delta})$ for $\boldsymbol{\delta} \in \mathbb{R}^d$,

$$p(\boldsymbol{\omega}) = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} k(\boldsymbol{\delta}) e^{-i\,\boldsymbol{\omega}\,\boldsymbol{\delta}} d\,\boldsymbol{\delta}\,,$$

and to consider the results $p(\boldsymbol{\omega})$ as a probability distribution, which is guaranteed by Bochner's Theorem [Bochner, 1933] [Rudin, 2017] over the coefficient $\boldsymbol{\omega} \in \mathbb{R}$. Indeed we can rewrite k as its inverse Fourier transform as,

$$k(\boldsymbol{\delta}) = \int_{\mathbb{R}^d} p(\boldsymbol{\omega}) e^{-i\omega\boldsymbol{\delta}} d\boldsymbol{\delta}$$

that can be expressed as

$$\begin{aligned}
&= \mathop{\mathbf{E}}_{\boldsymbol{\omega}\sim p} e^{-i\omega\boldsymbol{\delta}} \\
&= \mathop{\mathbf{E}}_{\boldsymbol{\omega}\sim p} [\cos(\boldsymbol{\omega}(\boldsymbol{\delta})) + i\sin(\boldsymbol{\omega}(\boldsymbol{\delta}))] \\
&= \mathop{\mathbf{E}}_{\boldsymbol{\omega}\sim p} \cos(\boldsymbol{\omega}(\boldsymbol{\delta})) \\
&= \mathop{\mathbf{E}}_{\boldsymbol{\omega}\sim p} \cos(\boldsymbol{\omega}\cdot(\mathbf{x}-\mathbf{x}')).
\end{aligned}$$

Furthermore, for a specific $\boldsymbol{\omega}$ we can express $\cos(\boldsymbol{\omega}\cdot(\mathbf{x}-\mathbf{x}'))$ as a product of two expression each relative to one of the point, by mapping them into,

$$z_{\boldsymbol{\omega}}(\mathbf{x}) = (\cos(\boldsymbol{\omega}\cdot\mathbf{x}), \sin(\boldsymbol{\omega}\cdot\mathbf{x})).$$

By the sum of angles formula we obtain,

$$\langle z_{\boldsymbol{\omega}}(\mathbf{x}), z_{\boldsymbol{\omega}}(\mathbf{x}')\rangle = \cos(\boldsymbol{\omega}\cdot(\mathbf{x}-\mathbf{x}'),$$

that when combined over $\boldsymbol{\omega}$ according to p as expected value $k_p(\boldsymbol{\delta})$,

$$\begin{aligned}
\mathop{\mathbf{E}}_{\boldsymbol{\omega}\sim p} \langle z_{\boldsymbol{\omega}}(\mathbf{x}), z_{\boldsymbol{\omega}}(\mathbf{x}')\rangle &= \mathop{\mathbf{E}}_{\boldsymbol{\omega}\sim p} \cos(\boldsymbol{\omega}\cdot(\mathbf{x}-\mathbf{x}') \\
&= k(\mathbf{x},\mathbf{x}').
\end{aligned}$$

For $\boldsymbol{\omega}$ sampled according to p the product $\langle z_{\boldsymbol{\omega}}(\mathbf{x}), z_{\boldsymbol{\omega}}(\mathbf{x}')\rangle$ is an unbiased estimator of $k(\mathbf{x},\mathbf{x}')$.

The idea is thus to approximate the kernel by drawing a limited number of coefficients. We draw a sample $\boldsymbol{\omega}_{i\,i=1}^{D} \sim p^D$ of D coefficients *i.i.d*, and represent every instances as a vector $\boldsymbol{z}(\mathbf{x}) \in [-1,1]^{2D}$,

$$\boldsymbol{z}(\mathbf{x}) = \frac{1}{\sqrt{D}} \left(\cos(\boldsymbol{\omega}_1\cdot\mathbf{x}), \sin(\boldsymbol{\omega}_1\cdot\mathbf{x}), \ldots, \cos(\boldsymbol{\omega}_D\cdot\mathbf{x}), \sin(\boldsymbol{\omega}_D\cdot\mathbf{x})\right).$$

We then get $\langle \boldsymbol{z}(\mathbf{x}), \boldsymbol{z}(\mathbf{x}')\rangle \approx k(\mathbf{x},\mathbf{x}')$, which is equivalent to,

$$\langle \boldsymbol{z}(\mathbf{x}), \boldsymbol{z}(\mathbf{x}')\rangle \approx \langle \phi(\mathbf{x}), \phi(\mathbf{x}')\rangle, \tag{15}$$

with ϕ the feature map upon which the kernel is built. Thus Equation (15) tells us that the transformation $\boldsymbol{z}(\mathbf{x})$ approximates the feature map $\phi(\mathbf{x})$ while only relying on D features and having $2D$ dimensions. It provides a way of "de-kernilizing" the kernel by approximating its feature map. The learning process can be done in the new representation $\boldsymbol{z}(\mathbf{x})$ by considering the matrix $\boldsymbol{Z}$ of size $n \times 2D$ associating to each instance its representation $\boldsymbol{z}$. This is especially useful when $2D \ll n$ because the matrix $\boldsymbol{Z}$ is more convenient to handle than the Gram matrix $\boldsymbol{K}$.

2.4.3 PAC-Bayes Random Fourier Features

PAC-Bayesian interpretation of Random Fourier Features

Using the Random Fourier Features to build a kernel approximation it is possible to interpret this approximation using the PAC-Bayesian theory as developed by [Letarte et al., 2019]. Indeed when using RFF the idea is to combine cosine representation $\cos(\boldsymbol{\omega}(\delta))$ according to the probability distribution over the $\boldsymbol{\omega}$ given by the Fourier transform of the kernel. [Letarte et al., 2019] propose to formalize this representation as hypotheses dependent on a coefficient $\boldsymbol{\omega} \in \mathbb{R}$, for $(x, y) \sim \mathcal{D}$ and $(x', y') \sim \mathcal{D}$, $h : \mathbb{R} \times \mathcal{X} \times \mathcal{X} \longrightarrow [-1, 1]$,

$$h_{\boldsymbol{\omega}}(\boldsymbol{\delta}) = cos(\boldsymbol{\omega}(\boldsymbol{\delta})). \tag{16}$$

From this the kernel can be considered as a "Bayesian predictor", in particular the kernel built by combining hypotheses $h_{\boldsymbol{\omega}}(\delta)$ according to distribution p is,

$$k_p(\boldsymbol{\delta}) = \mathop{\mathbf{E}}_{\boldsymbol{\omega}\sim p} h_{\boldsymbol{\omega}}(\boldsymbol{\delta}). \tag{17}$$

It is important to note that p can be viewed as a *prior* distribution suggesting that it is possible to learn a *posterior* distribution q that should better suited for the task considered, given the approximation of the kernel called a *pseudo kernel*,

$$k_q(\boldsymbol{\delta}) = \mathop{\mathbf{E}}_{\boldsymbol{\omega}\sim q} h_{\boldsymbol{\omega}}(\boldsymbol{\delta}).$$

Indeed, since $h_{\boldsymbol{\omega}}(\boldsymbol{\delta})$, can be seen as real-valued predictors. It is possible to measure how much the output of $h_{\boldsymbol{\omega}}(\boldsymbol{\delta})$ is coherent with the label of the couple $(\mathbf{x}, \mathbf{x}')$ given by $\lambda(y, y')$ in Equation (19), by generalizing the linear loss for this case where we consider to data points,

$$\ell(h_{\boldsymbol{\omega}}(\mathbf{x}, \mathbf{x}'), \lambda) = \frac{1}{2}\big(1 - \lambda h_{\boldsymbol{\omega}}(\mathbf{x}, \mathbf{x}')\big), \tag{18}$$

with,

$$\lambda = \lambda(y, y') = \begin{cases} 1, & \text{if } y = y'. \\ -1, & \text{otherwise.} \end{cases} \tag{19}$$

We want the output of the hypothesis to be close to 1 when the two points are of the same class and close to -1 when they are not. Then combining these losses each associated with a coefficient $\boldsymbol{\omega}$, according to a distribution q over those features, we can measure the performance of $k_q(\mathbf{x}, \mathbf{x}')$,

$$\begin{aligned} \ell(k_q(\mathbf{x}, \mathbf{x}'), \lambda) &= \mathop{\mathbf{E}}_{\boldsymbol{\omega}\sim q} \ell(h_{\boldsymbol{\omega}(}\, \mathbf{x}, \mathbf{x}'), \lambda) \\ &= \frac{1}{2}\big(1 - \lambda \mathop{\mathbf{E}}_{\boldsymbol{\omega}\sim q} h_{\boldsymbol{\omega}}(\mathbf{x}, \mathbf{x}')\big) \\ &= \frac{1}{2}\big(1 - \lambda k_q(\mathbf{x}, \mathbf{x}')\big). \end{aligned}$$

To be able measure the "performance" of this representation on the scale of the entire domain $\mathcal{D}$ we define the true risk for $h_{\boldsymbol{\omega}}$ with $\boldsymbol{\omega} \in \mathbb{R}$ as,

$$R_{\mathcal{D}}(h_{\boldsymbol{\omega}}) = \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{D}} \mathop{\mathbf{E}}_{(\mathbf{x}',y')\sim\mathcal{D}} \ell(h_{\boldsymbol{\omega}}(\mathbf{x},\mathbf{x}'),\lambda),$$

and the *kernel alignment* true risk for k_q as,

$$\begin{aligned} R_{\mathcal{D}}(k_p) &= \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{D}} \mathop{\mathbf{E}}_{(\mathbf{x}',y')\sim\mathcal{D}} \mathop{\mathbf{E}}_{\boldsymbol{\omega}\sim q} \ell(h_{\boldsymbol{\omega}}(\mathbf{x},\mathbf{x}'),\lambda) \\ &= \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{D}} \mathop{\mathbf{E}}_{(\mathbf{x}',y')\sim\mathcal{D}} \ell(k_q(\mathbf{x},\mathbf{x}'),\lambda) \\ &= \mathop{\mathbf{E}}_{\boldsymbol{\omega}\sim q} R_{\mathcal{D}}(h_{\boldsymbol{\omega}}). \end{aligned} \tag{20}$$

Note that the *kernel aligment* true risk expression is actually equivalent to the Gibbs risk of Equation (4). To develop further the analysis we need their empirical counterparts for a sample $S \sim \mathcal{D}^n$ of size n, the empirical risk of a hypothesis $h_{\boldsymbol{\omega}}$ is,

$$\widehat{R_S}(h_{\boldsymbol{\omega}}) = \frac{1}{n(n-1)} \sum_{i,j=1,i\neq j}^{n} \ell(h_{\boldsymbol{\omega}}(\mathbf{x}_i,\mathbf{x}_j),\lambda_{ij})$$

with

$$\lambda_{ij} = \lambda(y_i, y_j) \tag{21}$$

and the *kernel alignment* empirical risk for k_q is,

$$\begin{aligned} \widehat{R_S}(k_p) &= \frac{1}{n(n-1)} \mathop{\mathbf{E}}_{\boldsymbol{\omega}\sim p} \sum_{i,j=1,i\neq j}^{n} \ell(h_{\boldsymbol{\omega}}(\mathbf{x}_i,\mathbf{x}_j),\lambda_{ij}) \\ &= \mathop{\mathbf{E}}_{\boldsymbol{\omega}\sim p} \widehat{R_S}(h_{\boldsymbol{\omega}}). \end{aligned}$$

PAC-Bayes Random Fourier Features bound

From Equation (20) we recall here how the PAC-Bayesian setting can be applied in this set-up. The only issue is that the hypotheses and therefore every measure associated depend on couple of instances. We simplify by specifying the hypotheses by considering them for a fixed point $(\mathbf{x}_i, y_i) \in S$,

$$h^i_{\boldsymbol{\omega}}(\mathbf{x}) = \cos(\boldsymbol{\omega}\cdot(\mathbf{x}-\mathbf{x}_i)). \tag{22}$$

Furthermore, we specify the true and empirical risk in the same fashion,

$$\begin{aligned} R^i_{\mathcal{D}}(k_p) &= \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{D}} \ell(k_p(\mathbf{x},\mathbf{x}^i),\lambda) \\ &= \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{D}} \mathop{\mathbf{E}}_{\boldsymbol{\omega}\sim p} \ell(h^i_{\boldsymbol{\omega}}(\mathbf{x}),\lambda) \\ &= \mathop{\mathbf{E}}_{\boldsymbol{\omega}\sim p} R^i_{\mathcal{D}}(h^i_{\boldsymbol{\omega}}), \end{aligned}$$

and

$$\begin{aligned} \widehat{R^i_S}(k_p) &= \frac{1}{n_s-1} \mathop{\mathbf{E}}_{\boldsymbol{\omega}\sim p} \sum_{j=1,i\neq j}^{n} \ell(h^i_{\boldsymbol{\omega}}(\mathbf{x}_j),\lambda_{ij}) \\ &= \mathop{\mathbf{E}}_{\boldsymbol{\omega}\sim p} \widehat{R^i_S}(h^i_{\boldsymbol{\omega}}) \end{aligned}$$

Thus, it is possible to use transcript the PAC-Bayesian Theorem 1 over the risk $R^i_{\mathcal{D}}(k_p)$, we recall the Theorem presented in [Letarte et al., 2019].

Theorem 3 ([Letarte et al., 2019])**.** *For any domain $\mathcal{D}$, any set of hypotheses $\mathcal{H}$, any prior distribution P over $\mathcal{H}$, any $\epsilon \in (0,1]$, and any real number $c > 0$, with at least probability $1-\epsilon$ over the random choice $S \sim \mathcal{D}^n$ of size n, for every posterior distribution Q over $\mathcal{H}$, we have*

$$R^i_{\mathcal{D}}(k_q) \leq \widehat{R^i_S}(k_q) + \frac{1}{c}\left(KL(Q||P) + \frac{c^2}{2(n-1)} + ln\frac{1}{\epsilon}\right)$$

PAC-Bayes Random Fourier Features learning

To get the optimal posterior according to Theorem 3 for a fixed $(\mathbf{x}_i, y_i) \in S$ the part that needs to be minimized is

$$\widehat{R^i_S}(k_p) + \frac{1}{c}\mathrm{KL}(q||p).$$

Which gives the expression of the optimal posterior q^* using classical PAC-Bayesian development,

$$q^*(\boldsymbol{\omega}) = \frac{1}{Z} p(\boldsymbol{\omega}) \exp\left(-\tau\ \widehat{R^i_S}(h_{\boldsymbol{\omega}})\right) \tag{23}$$

with $\tau = \frac{1}{2}c$ and Z a normalization constant

However, considering the problem for only one $(\mathbf{x}_i, y_i) \in S$ is very limited since the whole dataset would have to be considered only relative to this instance. Therefore, to tackle this issue [Letarte et al., 2019] proposed a method that consists in learning a set of pseudo kernels for a fixed set of training instances called landmarks. Formally, we consider a sample of size n_l from the source training set $L = \{(\mathbf{x}_l, y_l)\}_{l=1}^{n_l} \sim S^{n_l}$ that will be our landmarks points and a prior distribution p that is the Fourier transform of

chosen kernel. For each landmark $(x_l, y_l) \in L$ let sample D coefficients $\boldsymbol{\omega}$ according to p such that $\Omega^l = \{\boldsymbol{\omega}_m^l\}_{m=1}^D \sim p^D$. We then replace the distribution p with a uniform distribution P over every set of coefficients Ω^l giving $P(\boldsymbol{\omega}_m^l) = \frac{1}{D}$. Therefore the hypothesis associated with a landmark point $\mathbf{x}_l$, a coefficient $\boldsymbol{\omega}_m^l$, and any data point $\mathbf{x}$ is $h_m^l = \cos(\boldsymbol{\omega}_m^l \cdot(\mathbf{x}_l - \mathbf{x}))$ and the pseudo kernel for P is

$$k_P^l(\mathbf{x}_l - \mathbf{x}) = \frac{1}{D}\sum_{m=1}^{D} \cos(\boldsymbol{\omega}_m^l \cdot(\mathbf{x}_l - \mathbf{x})). \tag{24}$$

Similarly as the optimal posterior expression in Equation (23) we get the posterior distribution associated with landmark $(\mathbf{x}_l, y_l)$,

$$Q^l(\boldsymbol{\omega}^l) = \frac{1}{Z^l} P^l(\boldsymbol{\omega}) \exp\left(-\beta\sqrt{n}\ \widehat{R_S^l}(h_{\boldsymbol{\omega}}^l)\right) \tag{25}$$

We obtain the following representation of every $\mathbf{x} \in \mathcal{D}$,

$$\phi(\mathbf{x}) = \left(k_{Q^1}(\mathbf{x}_1, \mathbf{x}), \ldots, k_{Q^{n_l}}(\mathbf{x}_{n_l}, \mathbf{x})\right).$$

This n_l dimensional mapping is assumed to be pertinent for the learning problem, because for every landmark point the learned pseudo kernel is optimized with the objective of having it be as well aligned with all instances in $\mathcal{D}$ as possible. The fact that these pseudo kernels are aligned using a PAC-Bayesian generalization bound gives guarantees about how well it should perform on the whole domain $\mathcal{D}$. Meaning, that when presented with new unknown data points the learnt mapping is guaranteed to remain relevant. The algorithm PB-landmarks developed by [Letarte et al., 2019] consists of using this new representation associated with landmark points to train a SVM.

2.5 Domain Adaptation

The classical framework of machine learning relies on an important assumption which is that the training data and the ones on which the model will be used come from the same unknown distribution. However, in practice this assumption is hard to verify and it is needed to design method specifically with this issue in mind. The framework that enables this is *transfer learning* [Torrey and Shavlik, 2010] [Pan and Yang, 2010] which originates from the observation that humans have an innate ability to transfer knowledge from one domain to another. For example, if a person learns to ride a bicycle, they can apply some of the learned skills (like balance and coordination) to similar tasks such as riding a scooter. In that sense, the principle of transfer learning in MLis to mimic this ability in ML models. The objective is to make the models able to take profit of existing knowledge Formally, we consider two data distributions, the source domain $\mathcal{S}$ over $\mathcal{X}_S \times \mathcal{Y}_S$ and the target domain $\mathcal{T}$ over $\mathcal{X}_T \times \mathcal{Y}_T$. We denote the marginal distribution over $\mathcal{X}_S$ as $\mathcal{S}_X$ and over $\mathcal{X}_T$ as $\mathcal{T}_X$ For each domain we define a learning task, in $\mathcal{T}$ it is $t_\mathcal{T}$ and in $\mathcal{S}$ it is $t_\mathcal{S}$.
While the usual learning paradigm implies $\mathcal{S}_X = \mathcal{T}_X$ and $t_\mathcal{T} = t_\mathcal{S}$ transfer learning can

be separated into three different scenario :

1. **Inductive Transfer Learning :** $\mathcal{S}_X = \mathcal{T}_X$ and $t_\mathcal{T} \neq t_\mathcal{S}$
2. **Transductive Transfer Learning :** $\mathcal{S}_X \neq \mathcal{T}_X$ and $t_\mathcal{T} = t_\mathcal{S}$
3. **Unsupervised Transfer Learning :** $\mathcal{S}_X \neq \mathcal{T}_X$ and $t_\mathcal{T} \neq t_\mathcal{S}$

We consider here the second case known under the name of *Domain Adaptation* (DA) [Redko et al., 2022] [Farahani et al., 2021]. A variety of techniques have been developed to tackle this problem. Specifically we stand in a special case of DA called unsupervised DA for classification where we have access to a labeled source sample S from$\mathcal{S}$ and an **unlabeled** target sample T from $\mathcal{T}$.

Therefore when drawing leaning samples we get a labelled source sample $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n_s} \sim \mathcal{S}^{n_s}$ and a unlabeled target sample $T = \{\mathbf{x}_j\}_{j=1}^{n_t} \sim \mathcal{T}^{n_t}$. This makes the learning of a model performing on $\mathcal{T}$ hard since the training cannot rely on information from the target labels. It is therefore necessary to rely on the labels of the source domain and make the interest of transfer learning evident. Indeed, in the classical learning paradigm it would be impossible to learn a classification model solely on $\mathcal{T}$ because of the absence of labels. One could possibly implement a clustering method but it would be much less powerful since it would give no guarantee on the correspondence of the clusters with the actual classes. However, by building a unsupervised DA procedure we are able to define a learning process that will be specifically designed to perform well for **classification** on $\mathcal{T}$, potentially coming with theoretical guarantees.

Our proposed approach of a binary classifier for unsupervised domain adaptation stands in the PAC-Bayesian framework and its specification for DA as recalled in the next section.

3 PAC-Bayes and Domain Adaptation

In this section we recall the important results on the specialization to the PAC-Bayesian framework developed by/in [Germain et al., 2020] [Germain et al., 2013a] [Germain et al., 2013b].

3.1 Domain adaptation bounds

Given a source domain $\mathcal{S}$, a target one $\mathcal{T}$ and a posterior distribution Q over the hypothesis space $\mathcal{H}$ as defined in Section 2.5. We have from Equation 10,

$$R_\mathcal{S}(G_Q) = \frac{1}{2} d_{\mathcal{S}_\mathcal{X}}(Q) + e_\mathcal{S}(Q) \tag{26}$$

$$R_\mathcal{T}(G_Q) = \frac{1}{2} d_{\mathcal{T}_\mathcal{X}}(Q) + e_\mathcal{T}(Q). \tag{27}$$

This allows the following expression that decomposes the difference between target and source true risk,

$$R_{\mathcal{T}}(G_Q) - R_{\mathcal{S}}(G_Q) = \frac{1}{2}\Big(d_{\mathcal{T}_\mathcal{X}}(Q) - d_{\mathcal{S}_\mathcal{X}}(Q)\Big) + \Big(e_{\mathcal{T}}(Q) - e_{\mathcal{S}}(Q)\Big). \tag{28}$$

3.1.1 First domain adaptation bound

Before going further we recall two useful quantities to derive the first bound. First, the pseudometric of domain disagreement $dis_Q\left(\mathcal{S}_\mathcal{X}, \mathcal{T}_\mathcal{X}\right)$ proposed in [Germain et al., 2020],

$$dis_Q\left(\mathcal{S}_\mathcal{X}, \mathcal{T}_\mathcal{X}\right) = \left|d_{\mathcal{S}_\mathcal{X}}(Q) - d_{\mathcal{T}_\mathcal{X}}(Q)\right|, \tag{29}$$

that can be viewed as a measure of divergence between the domains dependant on the distribution Q. Furthermore, for a specific domain the wanted behavior is to have hypotheses predicting similar values when presented with the same data point. This corresponds to a lower value of the disagreement measure. The value of $dis_Q\left(\mathcal{S}_\mathcal{X}, \mathcal{T}_\mathcal{X}\right)$ lowers as the similarity between $d_{\mathcal{S}_\mathcal{X}}(Q)$ and $d_{\mathcal{T}_\mathcal{X}}(Q)$ increases, meaning that the hypotheses have similar predictions in both domain. The best case scenario is to have a distribution of hypotheses Q providing low disagreement value for any domain Therefore $dis_Q\left(\mathcal{S}_\mathcal{X}, \mathcal{T}_\mathcal{X}\right)$ will measure the difference between the domains marginal distribution according to Q, with the aim of it being as small as possible indicating a little divergence in the domain when considered via Q. The other needed measure is the deviation between the expected joint errors λ_Q,

$$\lambda_Q = \left|e_{\mathcal{S}}(Q) - e_{\mathcal{T}}(Q)\right|. \tag{30}$$

This one does not measure the divergence between domain but provide a way of comparing the joint performance of classifiers between the two domains according to Q, by acknowledging the difference in performance in the two domains. We can notice that we do not dispose of the value of $e_\mathcal{T}$ by a lack of labels in the target domain. The value of λ_Q thus remain inaccessible. However, $e_\mathcal{T}$ and λ_Q are very convenient to manipulate in order to derive upper bounds on $R_\mathcal{T}(G_Q)$ and with small workarounds we can get formulations that don't rely on them in the end.

The first domain adaptation bound we consider relies on this two measures and the source domain risk $R_\mathcal{S}(G_Q)$ to upper bound the target risk $R_\mathcal{T}(G_Q)$.

Theorem 4. *Let $\mathcal{H}$ be an hypothesis space, let $\mathcal{S}$ be the source domain and $\mathcal{T}$ be the target domain both over $\mathcal{X} \times \mathcal{Y}$. For any distribution Q over $\mathcal{H}$, we have,*

$$R_\mathcal{T}(G_Q) \leq R_\mathcal{S}(G_Q) + \frac{1}{2} dis_Q\left(\mathcal{S}_\mathcal{X}, \mathcal{T}_\mathcal{X}\right) + \lambda_Q,$$

This result is interesting because it allows to upper bound the target risk $R_\mathcal{T}(G_Q)$ in a very intuitive way using only the source risk $R_\mathcal{S}(G_Q)$, the domain disagreement dis_Q and the deviation between the expected joint errors λ_Q.

3.1.2 Second domain adaptation bound

The authors of [Germain et al., 2020]developed a second bound that relies most specifically on the target risk decomposition highlighted in Equation 27. To do so they proposed a new measure of divergence called β_z-divergence that we recall in Equation (31) underneath.

$$\beta_z(\mathcal{T}\,||\,\mathcal{S}) = \left[\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{S}}\left(\frac{\mathcal{T}(\mathbf{x},y)}{\mathcal{S}(\mathbf{x},y)}\right)^z\right]^{\frac{1}{z}}. \tag{31}$$

With real value $z > 0$. It takes the expectation of the ration of the density of the data in the target domain over the source one, raised to the power q. For a data point $(\mathbf{x}, y)$ its density in domain $\mathcal{T}$ is noted $\mathcal{T}(\mathbf{x}, y)$ and in domain $\mathcal{S}$ it is $\mathcal{S}(\mathbf{x}, y)$. Its value should get higher the more the distribution $\mathcal{T}$ and $\mathcal{S}$ differ and the opposite should happen if the distribution are closed. Thus, a low value should indicate a high similarity between the domains which supposedly means an easier Domain Adaptation task. At the limit $z \to \infty$ the β_z-divergence takes the following form,

$$\beta_\infty\left(\mathcal{T}\,||\,\mathcal{S}\right) = \sup_{(\mathbf{x},y)\in\text{supp}(\mathcal{S})}\left(\frac{\mathcal{T}(\mathbf{x},y)}{\mathcal{S}(\mathbf{x},y)}\right), \tag{32}$$

with $\text{supp}(\mathcal{S})$ defining the support of $\mathcal{S}$.

The important issue regarding the β_z-divergence is that it only operates for instances $(\mathbf{x}, y)$ at the intersection between $\text{supp}(\mathcal{S})$ and $\text{supp}(\mathcal{T})$, that might not contain every points in the target domain. However, it is important to access information about the entire domain $\mathcal{T}$. To indicate the distribution of the data contained in the support of $\mathcal{T}$ but not in the support of $\mathcal{S}$, $(\mathbf{x}, y) \in \text{supp}(\mathcal{T})\backslash\text{supp}(\mathcal{S})$ we define $\mathcal{T}\setminus\mathcal{S}$. Inside this area we consider the worst possible risk denoted $\eta_{\mathcal{T}\setminus\mathcal{S}}$ as,

$$\eta_{\mathcal{T}\setminus\mathcal{S}} = \mathop{\mathbf{P}}_{(\mathbf{x},y)\sim\mathcal{T}}\big((\mathbf{x},y)\notin\text{supp}(\mathcal{S})\big)\sup_{h\in\mathcal{H}} R_{\mathcal{T}}(h_{\boldsymbol{\omega}}). \tag{33}$$

Using that it is possible to formulate a new upper bound on $R_{\mathcal{T}}(G_q)$, stated in Theorem 5 below.

Theorem 5. *Let $\mathcal{H}$ be an hypothesis space, let $\mathcal{S}$ be the source domain and $\mathcal{T}$ be the target domain both over $\mathcal{X}\times\mathcal{Y}$. For any distribution Q over $\mathcal{H}$, we have,*

$$R_{\mathcal{T}}(G_Q) \le \frac{1}{2}d_{T\mathcal{X}}(Q) + \beta_z(\mathcal{T}\,||\,\mathcal{S}) \times [e_s(Q)]^{1-\frac{1}{z}} + \eta_{\mathcal{T}\setminus\mathcal{S}}.$$

The second bound provided by Theorem 5 relies on a direct reinterpretation of the decomposition of $R_{\mathcal{T}}(G_q)$ stated in Equation 27. The β_z-divergence lets us bridge between $e_s(q)$ and $e_t(q)$, by acting as re-weighting of $e_s(q)$, while $\eta_{\mathcal{T}\setminus\mathcal{S}}$ gives an information about the worst risk in $\mathcal{T}\setminus\mathcal{S}$ since $e_{\mathcal{T}\setminus\mathcal{S}}(q)$ is not estimable. The measure $\beta_z(\mathcal{T}\,||\,\mathcal{S})$ also acts as a trade-off between target disagreement and joint source error, the more different the domain, the higher β-divergence and the more the emphasis is put on the joint source error.

It is important to note that these two domain adaptation bounds, while being anchored in the PAC-Bayesian setting, are not the final form of the bounds we want. Indeed, they rely solely on values that we cannot access directly, such as the source risk $R_{\mathcal{S}}(G_Q)$ or the true joint error and disagreement. They thus are very theoretically interesting but have no practical interest to derive a learning process. Furthermore, the notion of prior and posterior distribution dear to Bayesian framework are not exploited. The next step is to get from this, interesting but unpractical, non probabilistic bounds to PAC-Bayesian bounds that use estimable quantities and will consider a prior and a posterior distribution.

3.2 PAC-Bayesian bounds for domain adaptation

We recall here the procedure of [Germain et al., 2020] to formulate PAC-Bayesian bounds from the Domain Adaptations bounds of Subsection 3.1. The value we want to use in their empirical formulation are the source risk $\widehat{R_S}(G_q)$ as stated in Equation 1, the source joint error $\widehat{e_S}(Q)$ from Equation 12, the target disagreement $\widehat{d_{T_{\mathcal{X}}}}(Q)$ from Equation 11 and the domain disagreement $\widehat{\text{dis}_Q}$. The last one is defined for a sample T drawn from a target domain $\mathcal{T}^{n_t}$ of size n_t and a sample S drawn from a source domain $\mathcal{S}^{n_s}$ of size n_s, as,

$$\widehat{\text{dis}_Q} = |\widehat{d_{S_{\mathcal{X}}}}(Q) - \widehat{d_{T_{\mathcal{X}}}}(Q)|. \tag{34}$$

The next step to derive PAC-Bayesian bound is to use the empirical measure as recalled above to upper bound their true counterpart. This is the idea behind the following Theorem 6 from [Germain et al., 2020] that reformulates the PAC-Bayesian Theorem to be expressed explicitly on a generic loss function.

Theorem 6. *[Germain et al., 2020] For any domain $\mathcal{D}$ over $\mathcal{X} \times Y$, for any set of hypotheses $\mathcal{H}$, any prior distribution P over $\mathcal{H}$, any loss function $\ell : \mathcal{H} \times \mathcal{X} \times Y \longrightarrow [0, 1]$, any real number $\alpha > 0$, any value $\epsilon > 0$ with at least probability $1 - \epsilon$ over the random choice $S \sim \mathcal{D}^n$, we have for all Q over $\mathcal{H}$,*

$$\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{D}} \mathop{\mathbf{E}}_{h\sim Q} \ell(h(\mathbf{x}), y) \leq \frac{\alpha}{1 - e^{-\alpha}} \left[\frac{1}{n} \sum_{i=1}^{n} \mathop{\mathbf{E}}_{h\sim Q} \ell(h(\mathbf{x}^i), y^i) + \frac{KL(Q||P) + ln\frac{1}{\epsilon}}{m \times \alpha} \right]$$

This theorem is important in that it let derive upper bounds over the expected joint error $e_{\mathcal{S}}(Q)$, expected disagreement $d_{\mathcal{T}_{\mathcal{X}}}$ and domain disagreement $dis_q(\mathcal{S}_{\mathcal{X}}, \mathcal{T}_{\mathcal{X}})$ from which we will profit later. Those results are formulated in Corollary 1 from [Germain et al., 2020].

Corollary 1. *[Germain et al., 2020] For any source domain $\mathcal{S}$ and target domain $\mathcal{T}$ over $\mathcal{X} \times Y$, any hypotheses space $\mathcal{H}$, any prior distribution P over $\mathcal{H}$, any $\epsilon \in (0, 1]$, any real numbers $a > 0$, $b > 0$, $c > 0$, we have with at least probability $1 - \epsilon$,*

- *With probability* $1-\epsilon$ *over the choice* $T \sim \mathcal{T}^{n_t}$

$$\forall Q \text{ on } \mathcal{H},\ d_{\mathcal{T}_\mathcal{X}}(Q) \leq \frac{c}{1-e^{-c}}\left[\widehat{d_{T_\mathcal{X}}}(Q) + \frac{2KL(Q||P) + ln\frac{1}{\epsilon}}{n_t \times c}\right]$$

- *With probability* $1-\epsilon$ *over the choice* $S \sim \mathcal{S}^{n_s}$

$$\forall Q \text{ on } \mathcal{H},\ e_\mathcal{S}(Q) \leq \frac{b}{1-e^{-b}}\left[\widehat{e_S}(Q) + \frac{2KL(Q||P) + ln\frac{1}{\epsilon}}{n_s \times b}\right]$$

- *With probability* $1-\epsilon$ *over the choice* $S \times T \sim (\mathcal{S} \times \mathcal{T})^n$ *each of size* n

$$\forall Q \text{ on } \mathcal{H},\ dis_q(\mathcal{S}_\mathcal{X}, \mathcal{T}_\mathcal{X}) \leq \frac{2a}{1-e^{-2a}}\left[\widehat{dis_q}(S,T)(Q) + \frac{2KL(Q||P) + ln\frac{2}{\epsilon}}{n \times a} + 1\right] - 1$$

Further, Corollary 1 enables us to draw connections between the true quantities and their empirical estimates. Specifically, it provides us with PAC-Bayesian bounds for the disagreement and joint error in both the source and target domains. The parameters a, b, c control the trade-off between the empirical term and the complexity term represented by the KL-divergence between prior and posterior distribution $\mathrm{KL}(Q||P)$. An important detail to note from Corollary 1 is that the upper bound on $dis_q(\mathcal{S}_\mathcal{X}, \mathcal{T}_\mathcal{X})$ requires the two samples S and T to be of equal size n. This implies important limitation in practice since the samples might be of dramatically different size, using this result would then require to not use all information available.

Utilizing these formulations, we present two critical theorems that furnish PAC-Bayesian bounds for domain adaptation. The first result in Theorem 7 derives a bound for the risk in the target domain, $R_\mathcal{T}(G_q)$, in terms of empirical quantities, such as the empirical risk in the source domain, the empirical domain disagreement, and the KL-divergence between the prior and posterior distributions. The second results in Theorem 8 provides a tighter PAC-Bayesian bound by incorporating additional terms that capture the divergence between the source and target domains and the worst-case risk in the target domain outside the source domain's support. This bound provides a more nuanced view of the target domain's risk by directly taking into account the potential differences between the source and target domains with the β-divergence. More so, it does not include the empirical source risk $\widehat{R_S}(G_q)$ making it simpler. Both Theorems are reformulated from [Germain et al., 2020]

Theorem 7. *[Germain et al., 2020] For any domains* $\mathcal{S}$ *and* $\mathcal{T}$ *over* $X \times Y$*, their marginal distributions over* $\mathcal{X}$ $\mathcal{S}_\mathcal{X}$ *and* $\mathcal{T}_\mathcal{X}$ *any set of hypotheses* $\mathcal{H}$*, any prior distribution* P *over* $\mathcal{H}$*, any* $\epsilon \in (0,1]$*, any real numbers* $\gamma > 0$ *and* $a > 0$*, with at least probability* $1-\epsilon$ *over the random choice of* $S \times T \sim (\mathcal{S} \times \mathcal{T}_\mathcal{X})^n$ *both of size* n*, for every posterior distribution* Q *on* $\mathcal{H}$*, we have*

$$R_{\mathcal{T}}(G_q) \leq \gamma' \widehat{R_S}(G_q) + a' \frac{1}{2} \widehat{dis_q} + \left(\frac{\gamma'}{\gamma} + \frac{a'}{a} \right) \left(\frac{KL(Q||P) + ln \frac{3}{\epsilon}}{n} \right) + \lambda_q + \frac{1}{2}(a' - 1),$$

with $\gamma' = \frac{\gamma}{1-e^{-\gamma}}$ *and* $a' = \frac{2a}{1-e^{-2a}}$

Proof. The results is obtained by upper-bounding $R_{\mathcal{S}}(G_q)$ with probability $\frac{\epsilon}{3}$ using the formulation of Theorem 1 and dis_q with probability $\frac{2\epsilon}{3}$ according to Corollary 1. These upper bounds are then plugged in the non-probabilistic bound of Theorem 4 □

Theorem 7 presents an interesting PAC-Bayesian bound on the target risk $R_{\mathcal{T}}(G_q)$. The bound is defined in terms of an empirical risk on the source domain, $\widehat{R_S}(G_q)$, which is a tangible quantity we can calculate given labeled data in the source domain, and of domain disagreement $\widehat{\text{dis}_q(S,T)}$, which is the empirical disagreement between the source and target domains. This disagreement term reflects the dissimilarity between the two domains and will naturally be higher when the source and target domains are substantially different. In this bound, the coefficients γ' and a' are inversely proportional to their respective parameters γ and a, and represent trade-off factors for balancing the different terms in the bound. The bound is minimal when both the empirical risk on the source domain and the empirical disagreement are low, which indicates a successful adaptation from the source to the target domain.

The main drawbacks from this bounds is, first that the term α_q can still not be estimated since it relies on $e_{\mathcal{T}}(Q)$ while its value might give crucial information. Second that as pointed earlier the bound on $\text{dis}_q(\mathcal{S}, \mathcal{T})$ requires the two samples S and T to have the same size which is a great limitation in practice. We now formulate the second theorem as an adaptation of Theorem 5.

Theorem 8. *[Germain et al., 2020] For any domains* $\mathcal{S}$ *and* $\mathcal{T}$ *over* $X \times Y$*, their marginal distributions over* $\mathcal{X}$ $\mathcal{S}_{\mathcal{X}}$ *and* $\mathcal{T}_{\mathcal{X}}$ *any set of hypotheses* $\mathcal{H}$*, any prior distribution* P *over* $\mathcal{H}$*, any* $\epsilon \in (0, 1]$*, any real numbers ¿*0 *and* $b > 0$*, with at least probability* $1 - \epsilon$ *over the random choices of* $S \sim \mathcal{S}^{n_s}$ *of size* n_s *and* $T \sim \mathcal{T}_{\mathcal{X}}^{n_t}$ *of size* n_t*, for every posterior distribution* Q *on* $\mathcal{H}$*, we have*

$$R_{\mathcal{T}}(G_q) \leq c' \frac{1}{2} \widehat{d_T}(Q) + b' \widehat{e_s}(Q) + \eta_{\mathcal{T} \setminus \mathcal{S}} + \left(\frac{c'}{n_t c} + \frac{b'}{n_s b} \right) \left(2KL(Q||P) + ln \frac{2}{\epsilon} \right).$$

With $c' = \frac{c}{1-e^{-c}}$ *and* $b' = \beta_\infty(\mathcal{T} \,||\, \mathcal{S}) \frac{b}{1-e^{-b}}$.

Proof. The result is obtained by upper bounding $d_{\mathcal{T}}(Q)$ and $e_{\mathcal{S}}(Q)$ according to Corollary 1 and plugging them in Theorem 5 □

Theorem 8 provides a tighter PAC-Bayesian bound for domain adaptation. It incorporates the β-divergence in the form $\beta_\infty(\mathcal{T} \,||\, \mathcal{S})$. This divergence is weighted by $\frac{b}{1-e^{-b}}$, reflecting how the bound penalizes large divergence between the source and target distributions. What's interesting there is that this bound is minimalist in the term in contains

compared to the former one. Indeed, the empirical source risk is entirely absent here an the important values are the target empirical disagreement $\widehat{d_T}(Q)$ and the source empirical joint error $\widehat{e_s}$ that are simple to interpret, giving the bound an intuitive final form. This bound also incorporates the term $\eta_{\mathcal{T}\setminus\mathcal{S}}$ which is a measure of the worst-case risk in the target domain that is not covered by the source domain. This term accounts for the 'blind spots' in the target domain which we are unaware of due to the lack of labeled data in those regions. However, this term is problematic since it is purely theoretical because relies on an information on the target domain $\mathcal{T}$ that can never be accessed. The same can be said about $\beta_\infty(\mathcal{T}\,||\,\mathcal{S})$ since it can't be entirely computed, however it is possible to estimate it from available or considering it as an hyperparameter. Both solutions are not theoretically adequate with the bound but lets derive pertinent learning procedures.

These theorems provide the theoretical foundation for designing algorithms that can effectively adapt from a source domain to a target domain. In the next section we recall the algorithms derived from these results, PBDA [Germain et al., 2013a] from Theorem 7 and DALC [Germain et al., 2013b] from Theorem 8

3.3 PAC-Bayesian algorithms for Domain Adaptation : PBDA & DALC

The two domain adaptation learning PBDA and DALC consists on the specialisation of the results from Theorem 7 and Theorem 8 to linear classifier and spherical Gaussian distributions. The form of this procedures is inspired by [Germain et al., 2009a]. Relying on the same PAC-Bayesian results for domain adaptation as our contribution described in Section 4, PBDA and DALC are important protocols to put in perspective with the one we derive in Subsection 4.2.

Let $\mathcal{H}$ be a hypothesis space over linear classifiers in a d-dimensional space. A linear classifier $h_{\boldsymbol{w}'}(\mathbf{x}) \in \mathcal{H}$ associated with a vector $\boldsymbol{w}' \in \mathbb{R}^d$ is defined as,

$$h_{\boldsymbol{w}'}(\mathbf{x}) = sign(\boldsymbol{w}' \cdot \mathbf{x}) \tag{35}$$

The prior distribution $\pi_{\mathbf{0}}$ defined in Equation (36) and posterior distribution $p_{\boldsymbol{w}'}$ defined in Equation (37) are restricted to be Gaussian distributions with identity covariance matrix.

$$\pi_{\mathbf{0}} = \left(\frac{1}{\sqrt{2\boldsymbol{\pi}}}\right)^d \exp\left(-\frac{1}{2}||\boldsymbol{w}'||^2\right). \tag{36}$$

$$p_{\boldsymbol{w}} = \left(\frac{1}{\sqrt{2\boldsymbol{\pi}}}\right)^d \exp\left(-\frac{1}{2}||\boldsymbol{w}' - \boldsymbol{w}||^2\right). \tag{37}$$

This distributions can be seen as multivariate Normal distribution with identity covariance matrix. The prior $\pi_{\mathbf{0}}$ being center on the d-dimensional $\mathbf{0}$ vector, $\pi_{\mathbf{0}} = \mathcal{N}(0, \mathbf{I})$, and the posterior $p_{\boldsymbol{w}}$ being center on the vector $\boldsymbol{w}$ giving, $p_{\boldsymbol{w}} = \mathcal{N}(\boldsymbol{w}, \mathbf{I})$.

3.3.1 PBDA

The first learning procedure is derived from Corollary 2, the specialization of Theorem 7 to linear classifiers.

Corollary 2. *[Germain et al., 2013a] For any domains $\mathcal{S}$ and $\mathcal{T}$ over $X \times Y$, their marginal distributions over $\mathcal{X}$ $\mathcal{S}_{\mathcal{X}}$ and $\mathcal{T}_{\mathcal{X}}$ any set of hypotheses $\mathcal{H}$, any prior distribution P over $\mathcal{H}$, any $\epsilon \in (0, 1]$, any real numbers $\gamma > 0$ and $a > 0$, with at least probability $1 - \epsilon$ over the random choice of $S \times T \sim (\mathcal{S} \times \mathcal{T}_{\mathcal{X}})^n$ both of size n, for every posterior distribution Q on $\mathcal{H}$, we have*

$$R_{\mathcal{T}}(h_{\boldsymbol{w}}(\mathbf{x})) \leq 2\gamma' \widehat{R_S}(G_{p_{\boldsymbol{w}}}) + a' \widehat{dis}_{p_{\boldsymbol{w}}}(S, T) + KL(p_{\boldsymbol{w}} || \pi_{\mathbf{0}}) + 2\lambda_{p_{\boldsymbol{w}}} + 2\left(\frac{\gamma'}{\gamma} + \frac{a'}{a}\right) \frac{||\boldsymbol{w}||^2 + ln\frac{3}{\epsilon}}{n} + (a' - 1)$$

with $\gamma' = \frac{\gamma}{1 - e^{-\gamma}}$ and $a' = \frac{2a}{1 - e^{-2a}}$

The minimization of this bound for a labelled source sample $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^n \sim \mathcal{S}^n$ and a unlabeled target sample $T = \{\mathbf{x}_j\}_{j=1}^n \sim \mathcal{T}^n$, both of size n, gives the optimal posterior distribution $p_{\boldsymbol{w}}$. A strong assumption made by the authors is that the term $\lambda_{p_{\boldsymbol{w}}}$ is negligible, allowing to not include it in the optimization problem. Specifically, the formulation of the part of Corollary 2 to optimize for hyper-parameters $\Omega > 0$ $A > 0$ is given by Equation (38).

$$\Omega \widehat{R_S}(G_{p_{\boldsymbol{w}}}) + A \widehat{dis}_{p_{\boldsymbol{w}}}(S, T) + + KL(p_{\boldsymbol{w}} || \pi_{\mathbf{0}}) \tag{38}$$

Equation (38) is optimized through gradient descent.

3.3.2 DALC

The second learning procedure is derived from Corollary 3 by specializing Theorem 8 to linear classifiers.

Corollary 3. *[Germain et al., 2013b] For any domains $\mathcal{S}$ and $\mathcal{T}$ over $X \times Y$, their marginal distributions over $\mathcal{X}$ $\mathcal{S}_{\mathcal{X}}$ and $\mathcal{T}_{\mathcal{X}}$ any set of hypotheses $\mathcal{H}$, any prior distribution P over $\mathcal{H}$, any $\epsilon \in (0, 1]$, any real numbers $c > 0$ and $b > 0$, with at least probability $1 - \epsilon$ over the random choices of $S \sim \mathcal{S}^{n_s}$ of size n_s and $T \sim \mathcal{T}_{\mathcal{X}}^{n_t}$ of size n_t, for every posterior distribution Q on $\mathcal{H}$, we have*

$$R_{\mathcal{T}}(h_{\boldsymbol{w}}(\mathbf{x})) \leq c' \widehat{d_T}(p_{\boldsymbol{w}}) + 2b' \widehat{e_S}(p_{\boldsymbol{w}}) + \eta_{\mathcal{T} \setminus \mathcal{S}} + \left(\frac{c'}{n_t c} + \frac{b'}{n_s b}\right) \left(||\boldsymbol{w}||^2 + ln\frac{2}{\epsilon}\right)$$

With $c' = \frac{c}{1 - e^{-c}}$ and $b' = \beta_\infty(\mathcal{T} \,||\, \mathcal{S}) \frac{b}{1 - e^{-b}}$.

The minimization of this bound for a labelled source sample $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n_s} \sim \mathcal{S}^{n_s}$ and a unlabeled target sample $T = \{\mathbf{x}_j\}_{j=1}^{n_t} \sim \mathcal{T}^{n_t}$ gives the optimal posterior

distribution $p_{\boldsymbol{w}}$. The part of Corollary 3 to optimize is formulated in the following form,

$$C\widehat{d_T}(p_{\boldsymbol{w}}) + 2B\widehat{e_S}(p_{\boldsymbol{w}}) + ||\boldsymbol{w}||^2, \tag{39}$$

with hyper-parameters $B > 0$ $C > 0$. As for PBDA the optimization is done using gradient descent.

4 Contributions : a novel PAC-Bayesian bound for kernel approximation in Domain Adaptation

Our contributions sit at the intersection of PAC-Bayesian interpretation of RFF and domain adaptation, as we've discussed in the Sections 3 and 2.5 Our primary achievement lies in the derivation of a novel PAC-Bayesian bound, as presented in Theorem 10. This bound relates to the alignment, in the target domain, of the pseudo kernel associated with the posterior distribution Q, that operates in this case over $\mathbb{R}$. This bounded value is what we consider here as our target true risk, explicitly defined in Equation (41). The resulting Theorem corresponds to the reinterpretation of Theorem 8 and relies on a preliminary result in Theorem 9 that follows from Theorem 4 adaptation to kernel alignment. From Theorem 10 we derive a learning procedure to find an optimal posterior distribution regarding the output of the kernel in the target domain.

We must emphasize the nature of the hypotheses we are considering, defined in Equation (16). These hypotheses function over pairs of data points and enable us to reconstruct or approximate a specific kernel, k_p as described in Equation (17). Each of these hypotheses correspond to a feature $\boldsymbol{\omega} \in \mathbb{R}$ that are the coefficients given by the Fourier transform p of the kernel k_p taken as a probability distribution. A limited number of these feature is then drawn according to p and the task is to find an optimal posterior distribution Q for the data at hand, in this case by taking into account the fact that we have a target domain $\mathcal{T}$ with an unlabelled sample and a source domain § for which the labels are available. In the same fashion as the results from [Letarte et al., 2019] that we recalled in Section 2.4.3, our approach simplifies both the analysis and the derivation of generalization bounds by centering our hypotheses on a fixed set of data points. However, we introduce a subtlety by drawing these points solely from the source domain. Precisely, for a target domain $\mathcal{T}$, a source domain $\mathcal{S}$ for a sample $S \sim \mathcal{S}^{n_s}$ of size n_s an hypothesis is associated with a coefficient $\boldsymbol{\omega} \in \mathbb{R}$ a point $(\mathbf{x}^i, y^i) \in S$ and a data point $(\mathbf{x}, y)$ indifferently from $\mathcal{S}$ or $\mathcal{T}$, we have the hypothesis $h^i_{\boldsymbol{\omega}}(\mathbf{x})$ defined as,

$$h^i_{\boldsymbol{\omega}}(\mathbf{x}) = \cos(\boldsymbol{\omega} \cdot (\mathbf{x} - \mathbf{x}^i)), \tag{40}$$

similarly as Equation (22). The true risk over the target domain $\mathcal{T}$, for the kernel $k_q = \mathbf{E}_{\boldsymbol{\omega} \sim Q}\, h_{\boldsymbol{\omega}}(\mathbf{x})$ associated with a distribution Q and a fixed data points $(\mathbf{x}^i, y^i) \in S$ is thus given by,

$$R^i_{\mathcal{T}}(k_q) = \mathop{\mathbf{E}}_{\boldsymbol{\omega} \sim Q} \mathop{\mathbf{E}}_{(\mathbf{x},y) \sim \mathcal{T}} h^i_{\boldsymbol{\omega}}(\mathbf{x}). \tag{41}$$

Moreover, we must define the joint error and disagreement necessary to develop the analysis in our specific scenario. These metrics are derived from the alignment loss given in Equation (18). The joint error for both $\mathcal{T}$ and $\mathcal{S}$ Equations (42) and (43).

$$e^i_{\mathcal{T}}(q) = \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{T}} \mathop{\mathbf{E}}_{\boldsymbol{\omega},\boldsymbol{\omega'}\sim^2} \ell(h_{\boldsymbol{\omega}}(\mathbf{x},\mathbf{x}_i),\lambda_i)\cdot\ell(h_{\boldsymbol{\omega'}}(\mathbf{x},\mathbf{x}_i),\lambda), \tag{42}$$

$$e^i_{\mathcal{S}}(q) = \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{S}} \mathop{\mathbf{E}}_{\boldsymbol{\omega},\boldsymbol{\omega'}\sim^2} \ell(h_{\boldsymbol{\omega}}(\mathbf{x},\mathbf{x}_i),\lambda_i)\cdot\ell(h_{\boldsymbol{\omega'}}(\mathbf{x},\mathbf{x}_i),\lambda). \tag{43}$$

Additionally, the disagreement for the target domain $\mathcal{T}$ is encapsulated by Equation (44).

$$d^i_{\mathcal{T}_\mathbf{X}}(q) = \frac{1}{2}\left(1 - \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{T}_\mathbf{X}}\left[\mathop{\mathbf{E}}_{\boldsymbol{\omega}\sim q} h_{\boldsymbol{\omega}}(\mathbf{x},\mathbf{x}^i)\right]^2\right). \tag{44}$$

Connecting these metrics, we can represent the risk $R^i_{\mathcal{T}}(k_q)$ as a combination of the joint error and disagreement, as seen in Equation (45),

$$R^i_{\mathcal{T}}(k_q) = \frac{1}{2}d^i_{\mathcal{T}_\mathbf{X}}(q) + e^i_{\mathcal{T}}(q) \tag{45}$$

This decomposition is pivotal to the elaboration of upper bounds on the target true risk.

Proof. *The form of the following proof is greatly inspired by the equivalent one from [Germain et al., 2015] The proof requires the margin of the Bayes classifier as defined in Subsection 2.3.2 that we specialize for the present perspective. Consider the kernel* $k_q = \mathbf{E}_{\boldsymbol{\omega}\sim Q}\, h_{\boldsymbol{\omega}}(\mathbf{x})$ *associated with a distribution* Q *and a fixed data point* $(\mathbf{x}', y') \in \mathcal{S}$ *We define the random variable* $M_Q^{'\mathcal{T}}$ *that outputs for any data-point* $(\mathbf{x}, y) \in \mathcal{T}$ *the margin of the Bayes classifier in the same fashion as Equation* (5) *,*

$$M'_Q(\mathbf{x}, y) = \mathop{\mathbf{E}}_{\boldsymbol{\omega}\sim Q} \lambda(y, y')h'_{\boldsymbol{\omega}}(\mathbf{x}), \tag{46}$$

with $\lambda(y, y')$ *defined in Equation* (21)*. From that we get the first moment* $\mu'_1(M_Q^{'\mathcal{T}})$ *of the random variable* $M_Q^{'\mathcal{T}}$*, as*

$$\mu'_1(M_Q^{'\mathcal{T}}) = \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{T}} M'_Q(\mathbf{x}, y) \tag{47}$$

and its second moment $\mu'_2(M_Q^{'\mathcal{T}})$ *as,*

$$\begin{aligned}
\mu'_2(M_Q^{'\mathcal{T}}) &= \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{T}} \left[M'_Q(\mathbf{x}, y)\right]^2 \\
&= \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{T}} \lambda(y, y')^2 \left[h'_{\boldsymbol{\omega}}(\mathbf{x})\right]^2, \quad \lambda(y, y')^2 = 1 \\
&= \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{T}_\mathcal{X}} \left[h'_{\boldsymbol{\omega}}(\mathbf{x})\right]^2
\end{aligned} \tag{48}$$

Furthermore, we can rewrite the disagreement $d'_{\mathcal{T}_\mathbf{X}}(Q)$ *in the following way,*

$$\begin{aligned} d'_{\mathcal{T}_\mathbf{X}}(Q) &= \frac{1}{2}\left(1 - \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{T}_\mathbf{X}}\left[\mathop{\mathbf{E}}_{\boldsymbol{\omega}\sim Q} h_{\boldsymbol{\omega}}(\mathbf{x},\mathbf{x}')\right]^2\right) \\ &= \frac{1}{2}\left(1 - \mu'_2(M_Q^{'\mathcal{T}})\right) \end{aligned} \tag{49}$$

and the joint error $e'_{\mathcal{T}}(Q)$ *as,*

$$e'_{\mathcal{T}}(Q) = \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{S}} \mathop{\mathbf{E}}_{\boldsymbol{\omega},\boldsymbol{\omega}'\sim^2} \ell(h_{\boldsymbol{\omega}}(\mathbf{x},\mathbf{x}'),\lambda(y,y')) \cdot \ell(h_{\boldsymbol{\omega}'}(\mathbf{x},\mathbf{x}'),\lambda(y,y')) \tag{50}$$

Using the definition of $\mu'_1(M_Q^{'\mathcal{T}})$ *from Equation* (47) *and the reinterpretation of the disagreement and joint error From Equations* (49) (50) *we get,*

$$\mu'_1(M_Q^{'\mathcal{T}}) = 1 - (2e'_{\mathcal{T}}(Q) + d'_{\mathcal{T}_\mathbf{X}}(Q)). \tag{51}$$

Also, Equation (6) *let us reinterpret* $R'_{\mathcal{T}}(k_q)$ *as,*

$$R'_{\mathcal{T}}(k_q) = (1 - \mu'_1(M_Q^{'\mathcal{T}})) \tag{52}$$

. Combining Equation (51) *and Equation* (52) *we obtain the expected result :*

$$R'_{\mathcal{T}}(k_q) = \frac{1}{2} d'_{\mathcal{T}_\mathbf{X}}(q) + e'_{\mathcal{T}}(q)$$

4.1 Novel Bounds for Domain Adaptation

4.1.1 Domain Adaptation bound on the real target kernel alignment

Building from the Theorem 5, we craft an upper bound on the target true risk for the kernel. To translate the result of Theorem 5 the measure $\eta_{\mathcal{T}\setminus\mathcal{S}}$ from Equation 33 is specified for hypotheses over a fixed data point $(x^i, y^i) \in S$,

$$\eta^i_{\mathcal{T}\setminus\mathcal{S}} = \mathop{\mathbf{P}}_{(\mathbf{x},y)\sim\mathcal{T}}\big((\mathbf{x},y) \notin supp(\mathcal{S})\big) \sup_{\boldsymbol{\omega}\in\mathbb{R}^d} R^i_{\mathcal{T}}(h_{\boldsymbol{\omega}}) \tag{53}$$

Theorem 9. *Let* $\mathcal{H}$ *be an hypothesis space, let* $\mathcal{S}$ *be the source domain and* $\mathcal{T}$ *be the target domain both over* $\mathcal{X}\times\mathcal{Y}$*. For any distribution* Q *over* $\mathbb{R}$*, we have,*

$$R^i_{\mathcal{T}}(k_q) \leq \frac{1}{2} d^i_{\mathcal{T}_\mathcal{X}}(q) + \beta_\infty(\mathcal{T}\,||\,\mathcal{S}) \times e^i_{\mathcal{S}}(q) + \eta^i_{\mathcal{T}\setminus\mathcal{S}}.$$

Proof. The following proof is an adaptation and is very similar to the original proof of Theorem 8 in [Germain et al., 2020], the objective is to get rid of $e^i_{\mathcal{T}}(q)$ from the

decomposition of $R^i_{\mathcal{T}}(k_q)$ given by Equation 45.

$$
\begin{aligned}
e^i_{\mathcal{T}}(q) &= \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{T}} \mathop{\mathbf{E}}_{\boldsymbol{\omega},\boldsymbol{\omega'}\sim q^2} \ell(h_{\boldsymbol{\omega}}(\mathbf{x},\mathbf{x}^i),\lambda(y,y^i)) \cdot \ell(h_{\boldsymbol{\omega'}}(\mathbf{x},\mathbf{x}^i),\lambda(y,y^i)) \\
&= \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{S}} \frac{\mathcal{T}(\mathbf{x},y)}{\mathcal{S}(\mathbf{x},y)} \mathop{\mathbf{E}}_{\boldsymbol{\omega},\boldsymbol{\omega'}\sim q^2} \ell(h_{\boldsymbol{\omega}}(\mathbf{x},\mathbf{x}^i),\lambda(y,y^i)) \cdot \ell(h_{\boldsymbol{\omega'}}(\mathbf{x},\mathbf{x}^i),\lambda(y,y^i)) + \\
&\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{T}} \left((I[(\mathbf{x},y)] \notin \mathrm{supp}(\mathcal{S})) \mathop{\mathbf{E}}_{\boldsymbol{\omega},\boldsymbol{\omega'}\sim q^2} \ell(h_{\boldsymbol{\omega}}(\mathbf{x},\mathbf{x}^i),\lambda(y,y^i)) \cdot \ell(h_{\boldsymbol{\omega'}}(\mathbf{x},\mathbf{x}^i),\lambda(y,y^i)) \right)
\end{aligned} \tag{54}
$$

We upper bound separately the two parts of the equation. First, using Hölder's inequality 1 we upper bound the first part as,

$$
\begin{aligned}
&\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{S}} \frac{\mathcal{T}(\mathbf{x},y)}{\mathcal{S}(\mathbf{x},y)} \left(\mathop{\mathbf{E}}_{\boldsymbol{\omega},\boldsymbol{\omega'}\sim q^2} \ell(h_{\boldsymbol{\omega}}(\mathbf{x},\mathbf{x}^i),\lambda(y,y^i)) \cdot \ell(h_{\boldsymbol{\omega'}}(\mathbf{x},\mathbf{x}^i),\lambda(y,y^i)) \right) \leq \\
&\left[\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{S}} \frac{\mathcal{T}(\mathbf{x},y)}{\mathcal{S}(\mathbf{x},y)}^q \right]^{\frac{1}{q}} + \left[(\mathop{\mathbf{E}}_{\boldsymbol{\omega}\sim Q} \left(\mathop{\mathbf{E}}_{\boldsymbol{\omega},\boldsymbol{\omega'}\sim q^2} \ell(h_{\boldsymbol{\omega}}(\mathbf{x},\mathbf{x}^i),\lambda(y,y^i)) \cdot \ell(h_{\boldsymbol{\omega'}}(\mathbf{x},\mathbf{x}^i),\lambda(y,y^i)) \right))^p \right]^{\frac{1}{p}}.
\end{aligned}
$$

For the second part we have,

$$
\begin{aligned}
&\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{T}} \left((I[(\mathbf{x},y)] \notin \mathrm{supp}(\mathcal{S})) \mathop{\mathbf{E}}_{\boldsymbol{\omega},\boldsymbol{\omega'}\sim q^2} \ell(h_{\boldsymbol{\omega}}(\mathbf{x},\mathbf{x}^i),\lambda(y,y^i)) \cdot \ell(h_{\boldsymbol{\omega'}}(\mathbf{x},\mathbf{x}^i),\lambda(y,y^i)) \right) \\
&= \left(\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{T}} ((I[(\mathbf{x},y)] \notin \mathrm{supp}(\mathcal{S}))) \right) \mathop{\mathbf{E}}_{(\mathbf{x},y)\,\mathcal{T}\setminus\mathcal{S}} \mathop{\mathbf{E}}_{\boldsymbol{\omega},\boldsymbol{\omega'}\sim q^2} \ell(h_{\boldsymbol{\omega}}(\mathbf{x},\mathbf{x}^i),\lambda(y,y^i)) \cdot \ell(h_{\boldsymbol{\omega'}}(\mathbf{x},\mathbf{x}^i),\lambda(y,y^i)) \\
&= \left(\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{T}} ((I[(\mathbf{x},y)] \notin \mathrm{supp}(\mathcal{S}))) \right) \mathop{\mathbf{E}}_{(\mathbf{x},y)\,\mathcal{T}\setminus\mathcal{S}} e^i_{\mathcal{T}\setminus\mathcal{S}}(q) \\
&= \left(\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{T}} ((I[(\mathbf{x},y)] \notin \mathrm{supp}(\mathcal{S}))) \right) \left(R^i_{\mathcal{T}\setminus\mathcal{S}}(k_q) - \frac{1}{2} d^i_{\mathcal{T}\setminus\mathcal{S}}(q)(q) \right) \\
&\leq \left(\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{T}} ((I[(\mathbf{x},y)] \notin \mathrm{supp}(\mathcal{S}))) \right) \sup_{\boldsymbol{\omega}\in\mathbb{R}^d} R^i_{\mathcal{T}\setminus\mathcal{S}}(h^i_{\boldsymbol{\omega}}) = \eta^i_{\mathcal{T}\setminus\mathcal{S}}
\end{aligned}
$$

□

4.1.2 PAC-Bayesian bound on the real target kernel alignment

Capitalizing on the preceding theorem, we translate its insights into a PAC-Bayesian bounds, taking the form of Theorem 8. To do so we leverage results from Theorem 6 to upper bound the real values $d^i_{\mathcal{T}_{\mathcal{X}}}(q)$ and $e^i_{\mathcal{S}}(q)$ by their empirical counterparts, given for a target sample $T = \{x^t\}_{t=1}^{n_t} \sim \mathcal{T}^{n_t}$ and a source sample $S = \{(x^j, y^j\}_{j=1}^{n_s} \sim \mathcal{S}^{n_s}$ by,

$$
\hat{e}^i_S(Q) = \frac{1}{n_s - 1} \mathop{\mathbf{E}}_{\boldsymbol{\omega},\boldsymbol{\omega'}\sim Q^2} \sum_{j=1, i\neq j}^{n_s} \ell(h^i_{\boldsymbol{\omega}}(\mathbf{x}^j),\lambda_{ij}) \cdot \ell(h^i_{\boldsymbol{\omega'}}(\mathbf{x}^j),\lambda_{ij}), \tag{55}
$$

and

$$\hat{d}^i_{T_{\mathbf{X}}}(Q) = \frac{1}{n_t} \underset{(\boldsymbol{\omega},\boldsymbol{\omega}')\sim Q^2}{\mathbf{E}} \sum_{t=1}^{n_t} \ell(h^i_{\boldsymbol{\omega}}(\mathbf{x}^t), h^i_{\boldsymbol{\omega}'}(\mathbf{x}^t)). \tag{56}$$

The final PAC-Bayesian bound, as stipulated in Theorem 10, presents an ensemble of quantities that are, for the most part, estimable.

Corollary 4. *For any source domain $\mathcal{S}$ and target domain $\mathcal{T}$ over $\mathcal{X} \times Y$, with $(\mathbf{x}', y') \in \mathcal{S}$ any hypotheses space $\mathcal{H}$, any prior distribution P over $\mathcal{H}$, any $\epsilon \in (0, 1]$, any real numbers $a > 0$, $b > 0$, $c > 0$, we have with at least probability $1 - \epsilon$*

- *With probability $1 - \epsilon$ over the choice $T \sim \mathcal{T}^{n_t}$*

$$\forall Q \text{ on } \mathcal{H},\ d'_{\mathcal{T}_{\mathcal{X}}}(Q) \leq \frac{c}{1 - e^{-c}} \left[\hat{d}'_{T_{\mathbf{X}}}(Q) + \frac{2KL(Q||P) + ln\frac{1}{\epsilon}}{n_t \times c} \right]$$

- *With probability $1 - \epsilon$ over the choice $S \sim \mathcal{S}^{n_s}$*

$$\forall Q \text{ on } \mathcal{H},\ e'_{\mathcal{S}}(Q) \leq \frac{b}{1 - e^{-b}} \left[\hat{e}'_S(Q) + \frac{2KL(Q||P) + ln\frac{1}{\epsilon}}{n_s \times b} \right]$$

Proof. Since we consider the measures for a fixed point $(\mathbf{x}', y') \in \mathcal{S}$ the condition of Theorem 6 are verified if we take $y = \lambda(y, y')$ □

Corollary 4 gives PAC-Bayesian upper bounds $\hat{d}^i_{T_{\mathbf{X}}}(Q)$ and $\hat{e}^i_S(Q)$ using empirical value. Those bounds are the keypoint of the transcription of Theorem9 into the PAC-Bayesian form relying on empirical quantities of Theorem 10.

Theorem 10. *For any domains $\mathcal{S}$ and $\mathcal{T}$ over $X \times Y$, their marginal distributions over $\mathcal{X}$ $\mathcal{S}_{\mathcal{X}}$ and $\mathcal{T}_{\mathcal{X}}$ any set of hypotheses $\mathcal{H}$, any prior distribution P over $\mathcal{H}$, any $\epsilon \in (0, 1]$, any real numbers $c > 0$ and $b > 0$, with at least probability $1-\epsilon$ over the random choices of $S \sim \mathcal{S}^{n_s}$ of size n_s and $T \sim \mathcal{T}^{n_t}_{\mathcal{X}}$ of size n_t, with $i \in \{1, ..., n_s\}$, for every posterior distribution Q on $\mathbb{R}^d$, we have*

$$R^i_{\mathcal{T}}(k_q) \leq c' \frac{1}{2} \hat{d}^i_{T_{\mathcal{X}}}(q) + b' \hat{e}^i_S + \eta^i_{\mathcal{T}\setminus\mathcal{S}} + \left(\frac{c'}{n_t c} + \frac{b'}{n_s b} \right) \left(2KL(q||p) + ln\frac{2}{\epsilon} \right).$$

With $c' = \frac{c}{1-e^{-c}}$ and $b' = \beta_\infty(\mathcal{T}\,||\,\mathcal{S})\frac{b}{1-e^{-b}}$. Since $\beta_\infty(\mathcal{T}\,||\,\mathcal{S})$ is a non available measure of divergence between de domains $\mathcal{S}$ and $\mathcal{T}$ it is possible to consider it as an hyperparameter during the learning phase.

Proof. The bound is obtained by upper bounding the values of $\hat{d}^i_{T_{\mathcal{X}}}(q)$ and $\hat{e}^i_S$ in Theorem 9 using Corollary 4 each with probability $1 - \frac{1}{\epsilon}$. □

Theorem 10 furnishes us with an upper bound on the risk in the target domain. It combines terms related to disagreement and error. The β-divergence holds theoretical

importance in our work, acting as a measure of the dissimilarity or divergence between the source and target domains. By offering precise insights into the extent of domain shift it allows the trade off between source joint error and target error to be relevant regarding the divergence between domains. However, in practice we are faced with a limitation : its value is not directly computable. Limitation that is possible to circumvent by considering it as a tunable hyperparameter in the learning process. Validating this hyperparameter let the model implicitly gauge the domain shift by selecting the value giving the best performances.

This PAC-Bayesian bound is the main result of our work. Combining kernel approximation using Random Fourier Features 15, with domain adaptation 2.5. All of that through the PAC-Bayesian perspective, defined for Random Fourier Features from the work of [Letarte et al., 2019] in section 3 and for domain adaptation with the results from [Germain et al., 2020] in section 3. We now derive a learning strategy minimizing the bound with the objective of obtaining a suitable new representation of the data with learnt pseudo kernel.

4.2 Pseudo kernel learning in practice

Relying on Theorem 10, the learnt representation should minimizes the target true risk for the kernel alignment, meaning that the resulting space should be coherent with the class in the target domain even in the absence of labeled data. Indeed, we recall from Section 3 that minimizing the target true risk is synonymous of maximizing the matching of the pseudo kernel output with the labels. The output is obtained by a combination of cosine hypotheses each associated with a coefficient $\boldsymbol{\omega}$. Thus the obtained value is in interval $[-1, 1]$. Meaning that for a target instance $(x^j, y^j) \sim \mathcal{T}$ we want the kernel centered on a source point $(x^i, y^i) \in S$ to output a value near 1 when $\lambda_{ij} = 1$, i.e. when $y^j = y^i$ and -1 when $\lambda_{ij} = -1$, i.e. when $y^j \neq y^i$, with λ_{ij} defined in Equations (19) (21).

Learning a posterior distribution Q that minimizes the bound amounts to finding the optimal pondering of these hypotheses by putting the emphasis on the hypotheses giving values near λ_{ij}.

The minimum bound value for a training sample $(x_i, y_i) \in S$ is obtained by minimizing

$$c' \frac{1}{2} \hat{d}^i_{T_{\mathcal{X}}}(Q) + b' \hat{e}^i_S + \left(\frac{c'}{n_t c} + \frac{b'}{n_s b} \right) \left(2\mathrm{KL}(Q||p) \right). \tag{57}$$

With $c' = \frac{c}{1-e^{-c}}$ and $b' = \beta_\infty(\mathcal{T}\,||\,\mathcal{S}) \frac{b}{1-e^{-b}}$. It is noteworthy that the inestimable quantity $\eta_{\mathcal{T}\setminus\mathcal{S}}$ disappears there making the optimization process more manageable. The only non empirical value there is then $\beta_\infty(\mathcal{T}\,||\,\mathcal{S})$, that will be, as discussed above, as an hyperparameter.

landmark based learning

Since 10 bounds the target real risk for a fixed data point $(\mathbf{x}^i, y^i) \in S$, the posterior distribution Q is optimal for this specific point so we introduce a precision by denoting this distribution by Q^i. Learning a global kernel by minimizing the bound for a specific

$(\mathbf{x}^i, y^i) \in S$ and apply it to every point would be counter productive since the distribution over the hypothesis optimal for a point will not be for others. Similarly, learning a specific kernel for each instance in S would require way too much computation Therefore we consider a learning protocol relying on a subset of source training points called *landmarks*, by computing a specific posterior distribution for each of them. A kernel is obtained for each landmark, giving a coordinate in the final feature space.

Formally, we consider a sample of size n_l from the source training set $L = \{(\mathbf{x}_l, y_l)\}_{l=1}^{n_l} \sim \mathcal{S}^{n_l}$ that will be our landmarks points and a prior distribution p that is the Fourier transform of chosen kernel. For each landmark $(x_l, y_l) \in L$ let sample D coefficients $\boldsymbol{\omega}$ according to p such that $\Omega^l = \{\boldsymbol{\omega}_m^l\}_{m=1}^D \sim p^D$. We then replace the distribution p with a uniform distribution P over every set of coefficients Ω^l giving $P(\boldsymbol{\omega}_m^l) = \frac{1}{D}$. Therefore the hypothesis associated with a landmark point $(\mathbf{x}^l, y^l)$, a coefficient $\boldsymbol{\omega}_m^l$, and any data point $\mathbf{x}$ is $h_m^l(\mathbf{x}) = \cos(\boldsymbol{\omega}_m^l \cdot (\mathbf{x}_l - \mathbf{x}))$ and the pseudo kernel for the uniform prior distribution P is

$$k_P(\mathbf{x}_l - \mathbf{x}) = \frac{1}{D} \sum_{m=1}^{D} \cos(\boldsymbol{\omega}_m^l \cdot (\mathbf{x}_l - \mathbf{x})).$$

From this formulation our objective is to learn n_l posterior distribution Q_l each associated with a landmark point $\mathbf{x}_l \in L$. To simplify this computation corresponding to the minimization of Equation (57) we consider its matrix formulation. In particular, for a landmark point $(\mathbf{x}^l, y^l)$ we define the empirical target disagreement matrix D_T^l in Equation (58) and the empirical source joint error matrix E_S^l in Equation (58).

For a target sample $T = \{x^t\}_{t=1}^{n_t} \sim \mathcal{T}^{n_t}$ and with the disagreement between two fixed coefficients $(\boldsymbol{\omega}, \boldsymbol{\omega}') \in \mathbb{R}$ defined as $d_{\boldsymbol{\omega},\boldsymbol{\omega}'}^i = \frac{1}{n_t} \sum_{j=1}^{n_t} \ell(h_{\boldsymbol{\omega}}^i(\mathbf{x}^t), h_{\boldsymbol{\omega}'}^i(\mathbf{x}^t))$, the matrix D_T^l is given by,

$$D_T^l = \begin{bmatrix} d_{\boldsymbol{\omega}_1,\boldsymbol{\omega}_1}^l & 0 \cdots \cdots \cdots \cdots \cdots 0 \\ d_{\boldsymbol{\omega}_1,\boldsymbol{\omega}_2}^l & d_{\boldsymbol{\omega}_2,\boldsymbol{\omega}_2}^l \cdots \cdots \cdots \cdots \cdots 0 \\ \vdots & \ddots \\ & \ddots \\ & \ddots \\ d_{\boldsymbol{\omega}_1,\boldsymbol{\omega}_m}^l & d_{\boldsymbol{\omega}_2,\boldsymbol{\omega}_m}^l \quad d_{\boldsymbol{\omega}_3,\boldsymbol{\omega}_m}^l \cdots \cdots d_{\boldsymbol{\omega}_m,\boldsymbol{\omega}_m}^l \end{bmatrix}. \tag{58}$$

Similarly, for a source sample $S = \{(x^s, y^s\}_{s=1}^{n_s} \sim \mathcal{S}^{n_s}$, with the source joint error for two coefficients as $e_{\boldsymbol{\omega},\boldsymbol{\omega}'}^i = \frac{1}{n_s - 1} \sum_{j=1, i \neq j}^{n_s} \ell(h_{\boldsymbol{\omega}}^i(\mathbf{x}^j), \lambda_{ij}) \cdot \ell(h_{\boldsymbol{\omega}'}^i(\mathbf{x}^j), \lambda_{ij})$, the matrix E_S^l is formulated as,

$$E_S^l = \begin{bmatrix} e_{\boldsymbol{\omega}_1,\boldsymbol{\omega}_1}^i & 0 \cdots \cdots \cdots \cdots \cdots 0 \\ e_{\boldsymbol{\omega}_1,\boldsymbol{\omega}_2}^i & e_{\boldsymbol{\omega}_2,\boldsymbol{\omega}_2}^i \cdots \cdots \cdots \cdots \cdots 0 \\ \vdots & \ddots \\ & \ddots \\ & \ddots \\ e_{\boldsymbol{\omega}_1,\boldsymbol{\omega}_m}^i & e_{\boldsymbol{\omega}_2,\boldsymbol{\omega}_m}^i \quad e_{\boldsymbol{\omega}_3,\boldsymbol{\omega}_m}^i \cdots \cdots e_{\boldsymbol{\omega}_m,\boldsymbol{\omega}_m}^i \end{bmatrix}. \tag{59}$$

This allows the formulation of the optimization problem to find the posterior distribution

Q^l, in the following form, for fixed hyperparameters values $\beta_\infty(\mathcal{T}\,||\,\mathcal{S}) > 0$, $c > 0$, $b > 0$,

$$\begin{aligned} Q_l = \min_{Q}\ & c'\frac{1}{2}Q^t D_T^l\, Q + b' Q^t E_S^l\, Q + \left(\frac{c'}{n_t c} + \frac{b'}{n_s b}\right) 2\mathrm{KL}(Q||P) \\ s.t\ & \sum_{m=1}^{D} Q(\boldsymbol{\omega}_{\boldsymbol{m}}^{\boldsymbol{l}}) = 1, \end{aligned} \tag{60}$$

With $c' = \frac{c}{1-e^{-c}}$ and $b' = \beta_\infty(\mathcal{T}\,||\,\mathcal{S})\frac{b}{1-e^{-b}}$. Indeed, we have $\hat{d}^i_{T_\mathcal{X}}(q) = Q^t D_T^l\, Q$ and $\hat{e}^i_S = Q^t E_S^l\, Q$. Due to the constrained nature of problem presented in Equation (60) we formulate the Lagrangian dual function,

$$\begin{aligned} \mathcal{L}(Q,\lambda) = c'\frac{1}{2}Q^t D_T^l\, Q + b' Q^t E_S^l\, Q \\ + \left(\frac{c'}{n_t c} + \frac{b'}{n_s b}\right) 2\mathrm{KL}(Q||P) + \lambda \mathbf{1}^t Q \end{aligned} \tag{61}$$

and the associated gradient,

$$\frac{\partial \mathcal{L}}{\partial Q} = c' Q^t D_T^l + b' Q^t E_S^l + \left(\frac{c'}{n_t c} + \frac{b'}{n_s b}\right) 2\left(1 + ln\left(\frac{Q}{P}\right)\right) - \lambda \mathbf{1}^t \tag{62}$$

With $\lambda \mathbf{1}^t$ a vector of ones of size m the number of coefficients. By lack of a closed form solution for the posterior, through the Lagrangian dual function of the problem, we choose to solve Equation (60) using gradient descent. Specifically we use a trust region method for constrained optimization with the an implementation of the method from [Byrd et al., 1999] which makes the optimization process happen only in the regions where the constraints are verified.

pseudo kernel representation

The set of learnt optimal posterior distribution $\{Q_l\}_{l=1}^{n_l}$, translate into a set of similarity features $\{k_{Q_l}\}_{l=1}^{n_l}$, each defined as,

$$k_{Q_l}(\mathbf{x}_l - \mathbf{x}) = \sum_{m=1}^{D} Q(\boldsymbol{\omega}_m^l)\cos(\boldsymbol{\omega}_m^l \cdot (\mathbf{x}_l - \mathbf{x})).$$

Each of these features, built upon a landmark point $\mathbf{x}^l$, is itself a combination of similarity features $\cos(\boldsymbol{\omega}_m^l \cdot (\mathbf{x}_l - \mathbf{x}))$ and is not a kernel anymore, motivating the denomination of *pseudo kernel*. While the cosine functions remains the pseudo-kernel's core, the pondered summation indicates a composite behavior s. Here, the mapping $\phi(\mathbf{x})$, described in Equation (63), for a point $\mathbf{x} \in \mathbb{R}^d$ encodes the similarities of the data point with respect to each landmark. The new representation thereby offers a summary of the data point's relationship with some representatives from the source domain.

$$\phi(\mathbf{x}) = \left(k_{Q_1}(\mathbf{x}_1 - \mathbf{x}), \ldots, k_{Q_{n_l}}(\mathbf{x}_{n_l} - \mathbf{x})\right), \tag{63}$$

Utilizing this pseudo-kernel space, the goal would be to train models that can more effectively generalize across domains by leveraging the landmark-based learned representations. As a result, models trained in this feature space are expected to have improved performance in domain adaptation tasks by relying on similarity features bridging the knowledge gap between the source and target domains. In the present work the model learnt using this representation is a SVM.

5 Experiments

5.1 Landmarks selection

A crucial question when learning via landmark points is how to choose them. We propose to implement here 2 methods of selection, at random and using k-medoids We stress again the fact that in each case the landmark points are selected from the source training set S_{tr}. We denote the set of source training points with label $+1$ of size n_s^+, $S^+ = \{(x_i, +1)\}_1^{n_s^+}$ and the set of source training points with label -1 of size n_s^-, $S^- = \{(x_i, -1)\}_1^{n_s^-}$

Random

The first and most straightforward way of making the selection is randomly. Meaning that we do not have any restriction on which points are chosen with the exception of class balance. For an even number of n_l landmarks we select $\frac{1}{2}n_l$ points in S^+ and $\frac{1}{2}n_l$ points in S^- with no restriction regarding their coordinates.

K-medoids

To get hopefully more stable and better results we consider a second way of selecting landmarks. We use the well-known K-means clustering algorithm which search iteratively for synthetic centers for a pre-specified number of clusters K. In this case, we consider its variant K-medoids where the K clusters are not synthetic points but an actual point from the cluster such that it is the most central. For an even number of cluster n_l the K-medoids procedure is done on S^+ with $K = \frac{1}{2}n_l$ and another K-medoids clustering is done on S^- with $K = \frac{1}{2}n_l$.

5.2 Hyper-parameters tuning

Hyper-parameters tuning is a central question when learning a model. Since they can dramatically impact the performances of the final model it is necessary to chose them carefully. One common solution in supervised classification is to make use of k-fold cross validation and its extension grid search cross validation that relies on hold a k^{th}

of the training data and evaluating the performance of the model on them with the consideration that it will be a good estimator of the performance on the test data set. However, since we are considering a domain adaptation task the performance we want to measure is on the target domain and since we are in an unsupervised set up we do not have access to the label for this domain, implying that a classical cross-validation cannot be performed.. Therefore the target performance cannot be estimated and we need to build a workaround to have a validation procedure that is pertinent in the current scenario. In the present work we implement two strategies, source cross validation, that only consider the source training set and reverse validation [Zhong et al., 2010] which is specifically designed for Domain Adaptation tasks.

Reverse validation

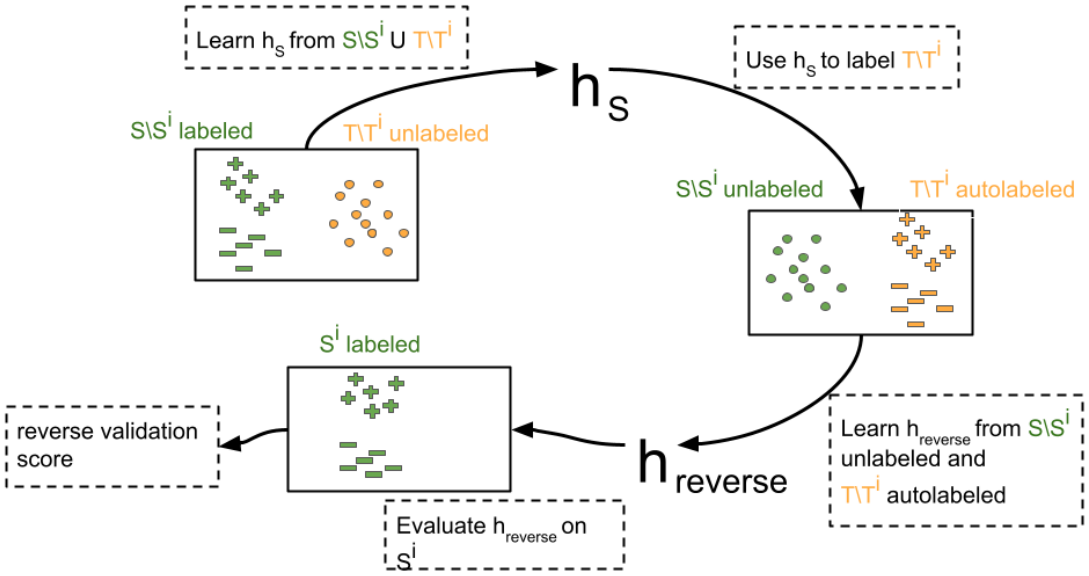


Figure 2: Illustration of the reverse validation procedure

The reverse validation [Zhong et al., 2010] also known as circular validation is a validation strategy that relies both on the source labeled training set and on the target unlabeled training set to evaluate the capacity of a model to label new data accurately in a DA setting.

Formally, given a k-folds stratification on the source labeled sample S, such that $S = S_1 \cup \ldots \cup S_k$, a k-folds stratification on the unlabeled target sample T, such that $T = T_1 \cup \ldots \cup T_k$, a tuple of hyper-parameters and a learning algorithm parameterized by these parameters. A first classifier h_S is trained using the labeled set $S \setminus Si = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n_{S\setminus S_i}}$ as source and the unlabeled set $T \setminus T_i = \{(\mathbf{x}_j)\}_{j=1}^{n_{T\setminus T_i}}$ as target. Then h_S is used to label $T \setminus Ti$ such that we get a new set $T_{labeled} = \{(\mathbf{x}_j, h_S(x_j)\}_{j=1}^{n_{T_{labeled}}}$. Finally a

reverse classifier $h_{reverse}$ is trained using $T_{labeled}$ as source and $S \setminus S_i$ as target, then its performance are evaluated on S_i, in our case using the prediction accuracy giving the reverse cross-validation risk $\hat{v}_{i,p}$ for fold i and set of hyper-parameters p. This process is repeated for each of the k folds and we select the set of parameter p minimizing :

$$\hat{V}_p = \frac{1}{k} \sum_{i=1}^{k} \hat{v}_{i,p}. \tag{64}$$

This process is evaluated in Figure 2

5.3 Protocols for comparison

The different protocol implement for comparison are

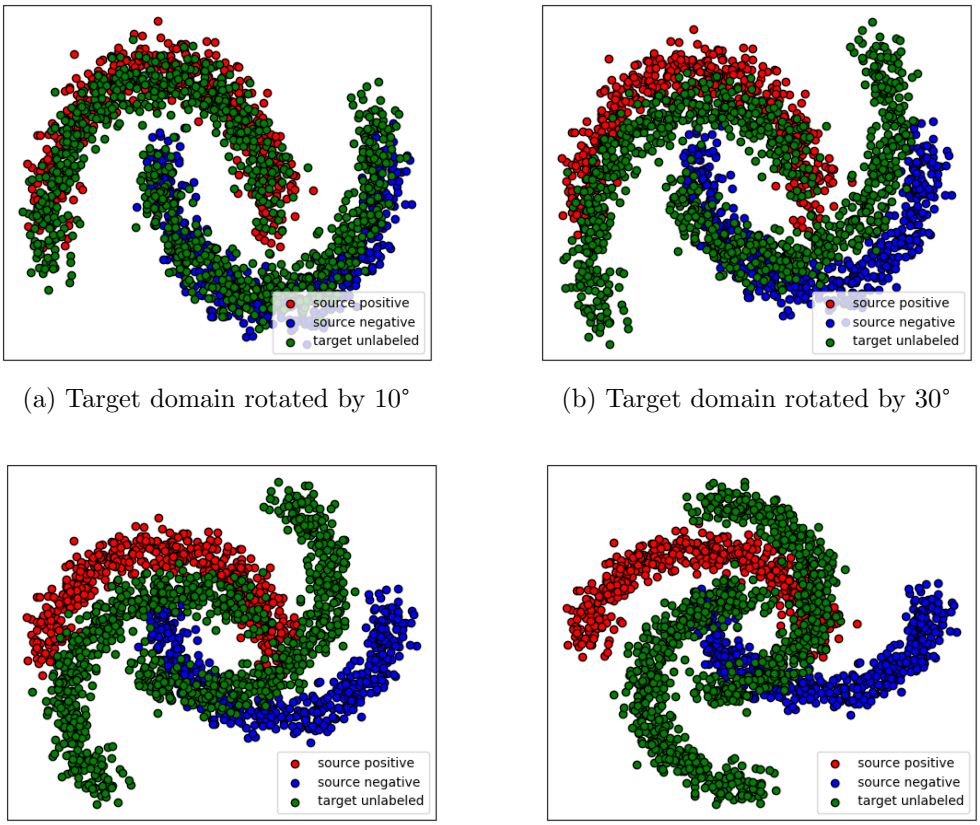
- Our protocol PBDA-landmarks described in Subsection 4.2 with a Gaussian kernel with and without reverse validation
- PBRFF-landmarks as described in Subsubsection 2.4.3 with and without reverse validation, using a Gaussian kernel
- RBF-landmarks, corresponding to the following protocol. Given a set L of n_l landmarks, the data points are mapped to the empirical kernel map composed of RBF kernels centered on each landmark, such that, for a point $\mathbf{x}$ we have its mapping $\phi(\mathbf{x})$
$$\phi(\mathbf{x}) = \big(k(\mathbf{x}_1, \mathbf{x}), \ldots, k(\mathbf{x}_{n_l}, \mathbf{x})\big).$$
In this case we use a Gaussian kernel with simple cross validation
- PBDA from 3.3.1 and DALC from 3.3.2 with reverse validation
- Transductive Support Vector Machine (TSVM) [Vapnik and Chervonenkis, 1982] a transductive version of SVM performent with Domain Adaptation tasks with simple cross validation.
- SVM with simple cross validation.

SVM and TSVM are the baseline protocols for comparison, since the first one is a classical classification algorithm with no specificity for DA and the second one is an adaptation of the first supposedly better at DA tasks. Since, we consider a Gaussian kernel for every protocol relying on kernels we select the parameter γ corresponding to the dilatation of the kernel via cross validation on the SVM. Every other parameter is selected specifically for each learning protocol and with different ranges for the toy experiment in Subsection 5.4 and the experiment on real datas in Subsection 5.5.

5.4 Toy experiments

We first experiment on the two intertwining moons dataset illustrated in figure 3 with rotation degrees $d \in \{10, 20, 30, 40, 50, 70, 90\}$ by doing 10 runs for each d and averaging

the results by rotation degree. The source test set is generated first by sampling 3000 points at 0° degree of rotation. Then for each d a target test set of also of 3000 points and at rotation degree d is generated. Finally at each run we generate a source and a target training set each composed of 300 points respectively with 0 and d degrees of rotation. Each d leads to an increasingly difficult DA task, since the target set will be more and more distant with the source set as illustrated in Figure 3.



(a) Target domain rotated by 10°

(b) Target domain rotated by 30°

(c) Target domain rotated by 50°

(d) Target domain rotated by 90°

Figure 3: Illustration of domain adaptation tasks corresponding with different degrees of rotation from 10° to 90°

At the initialization of each learning procedure a new set of landmarks is selected from the source training set, such that we have the same number of landmarks by label.

The γ parameter, that will be shared by every kernel method, is selected via source validation in the range $\gamma \in \{10^{-7}, 10^{-6}, ..., 10^{2}\}$. For every method that relies on learning a SVM we select the penalisation constant $C \in \{10^{-1}, 10^{1}, 10^{3}\}$. When landmarks are involved we have $n_l \in \{2, 4, 8, 16\}$ and if the model uses random Fourier features their number D is selected from $D \in \{4, 8, 16\}$. The ranges of the hyper-parameters

specific to algorithms are the following,

PB-Landmarks : from Equation (25), the trade off parameter $\beta \in \{10^{-3}, 10^{1}, 10^{3}\}$.

PBDA-landmarks : The parameters β_∞, c, b from optimization problem in Equation (60), we get them in the ranges $\beta_\infty \in \{10^{-3}, 10^{1}, 10^{3}\}$ and c, $b \in \{10^{-2}, 10^{0}, 10^{1}\}$.

DALC : from Equation (39) $C \in [10^{-2}, 10^{0}, 10^{1}]$ and $B \in [10^{-2}, 10^{0}, 10^{1}]$.

PBDA : from Equation (38) $\Omega \in [10^{-2}, 10^{0}, 10^{1}]$ and $A \in [10^{-2}, 10^{0}, 10^{1}]$.

5.4.1 Results

Table 1 contains the accuracy on the target dataset for the two intertwining moons at every rotation degree with random landmark selection. The proposed approach without reverse validation gives the best result in average but does not win by a lot in comparison with PB-landmarks even though it is not designed for DA. Its performance surpass the results of DALC and PBDA for low degrees of rotation but fail to keep this trend for higher degrees. This might indicate that the model fails to capture information when the two domains are too different by maybe focusing too much on the source domain. This is confirmed by the result on the source dataset in Table 2. Indeed, DALC and PBDA show very poor result on source data while our method shows almost no classification error. This behaviour indicates a strong fitting to the source domain at the detriment of the target one even though the objective was to learn a common space of representation. Still, being good on the source domain is appreciable in practice since the model might be used on both domains. The stability of PBDA-Landmarks on source points is thus a good point.
The results for K-medoids landmark selection in Table 3 for the target domain and Table 4 for the source one point in the same direction. The only specificity here is that the best performing protocol is PBDA-Landmark but this time with reverse cross validation. This landmark selection procedure does not improve the performance in this learning scenario.

Table 1: Mean target test error over 10 runs

Degree	SVM	TSVM	RBF-landmarks	PB-landmarks	PBDA-landmarks	PB-landmarks^{RCV}	PBDA-landmarks^{RCV}	DALC	PBDA
10	0.004 ± 0.002	0.146 ± 0.001	**0.008 ± 0.008**	0.012 ± 0.008	0.013 ± 0.012	0.010 ± 0.005	0.012 ± 0.008	0.154 ± 0.003	0.158 ± 0.003
20	0.039 ± 0.015	0.122 ± 0.001	0.056 ± 0.033	0.067 ± 0.038	0.056 ± 0.037	**0.056 ± 0.030**	0.059 ± 0.032	0.187 ± 0.015	0.205 ± 0.004
30	0.158 ± 0.016	0.104 ± 0.001	0.185 ± 0.060	0.185 ± 0.078	0.175 ± 0.073	**0.151 ± 0.067**	0.175 ± 0.082	0.263 ± 0.014	0.275 ± 0.007
40	0.336 ± 0.030	0.141 ± 0.002	0.357 ± 0.064	0.348 ± 0.102	0.311 ± 0.101	0.302 ± 0.105	**0.289 ± 0.110**	0.326 ± 0.011	0.337 ± 0.005
50	0.493 ± 0.021	0.206 ± 0.001	0.497 ± 0.075	0.498 ± 0.107	0.431 ± 0.132	0.520 ± 0.091	0.458 ± 0.082	**0.382 ± 0.009**	0.389 ± 0.007
70	0.709 ± 0.035	0.343 ± 0.002	0.672 ± 0.112	0.665 ± 0.111	0.584 ± 0.153	0.594 ± 0.144	0.641 ± 0.128	0.501 ± 0.005	**0.497 ± 0.006**
90	0.812 ± 0.057	0.484 ± 0.001	0.746 ± 0.108	0.743 ± 0.118	0.698 ± 0.120	0.744 ± 0.065	0.735 ± 0.083	0.625 ± 0.027	**0.589± 0.006**
mean	0.364	0.220	0.360	0.359	**0.324**	0.339	0.338	0.348	0.350

Table 2: Mean source test error over 10 runs

Degree	SVM	TSVM	RBF-landmarks	PB-landmarks	PBDA-landmarks	PB-landmarks^{RCV}	PBDA-landmarks^{RCV}	DALC	PBDA
10	0.000 ± 0.000	0.174 ± 0.001	0.074 ± 0.000	0.001 ± 0.001	**0.001 ± 0.000**	0.001 ± 0.000	0.001 ± 0.001	0.613 ± 0.003	0.618 ± 0.002
20	0.000 ± 0.000	0.174 ± 0.001	0.074 ± 0.000	0.001 ± 0.000	0.001 ± 0.000	**0.000 ± 0.000**	0.001 ± 0.001	0.611 ± 0.005	0.618± 0.002
30	0.000 ± 0.000	0.173 ± 0.001	0.074 ± 0.000	0.001 ± 0.000	0.001 ± 0.001	**0.000 ± 0.000**	0.001 ± 0.000	0.614 ± 0.004	0.619 ± 0.002
40	0.000 ± 0.001	0.172 ± 0.002	0.074 ± 0.000	0.001 ± 0.000	0.001 ± 0.000	**0.000 ± 0.000**	0.001 ± 0.001	0.614 ± 0.004	0.618 ± 0.002
50	0.000 ± 0.000	0.172 ± 0.002	0.074 ± 0.000	0.001 ± 0.000	0.001 ± 0.000	**0.000 ± 0.000**	0.001 ± 0.000	0.615 ± 0.004	0.618 ± 0.004
70	0.000 ± 0.001	0.170 ± 0.002	0.074 ± 0.000	0.001 ± 0.000	0.001 ± 0.001	**0.001 ± 0.000**	0.000 ± 0.000	0.619 ± 0.002	0.617 ± 0.003
90	0.000 ± 0.000	0.169 ± 0.001	0.074 ± 0.000	0.001 ± 0.000	0.001 ± 0.01	0.005 ± 0.009	**0.000 ± 0.000**	0.634 ± 0.021	0.611 ± 0.002
mean	0.000	0.172	0.074	0.001	0.001	0.001	**0.001**	0.617	0.617

Table 3: Mean target test error over 10 runs

Degree	SVM	TSVM	RBF-landmarks	PB-landmarks	PBDA-landmarks	PB-landmarks^{RCV}	PBDA-landmarks^{RCV}	DALC	PBDA
10	0.004 ± 0.002	0.146 ± 0.001	0.007 ± 0.002	0.012 ± 0.009	**0.005 ± 0.004**	0.006 ± 0.004	0.008 ± 0.003	0.154 ± 0.003	0.158 ± 0.003
20	0.039 ± 0.015	0.122 ± 0.001	0.071 ± 0.013	0.063 ± 0.039	0.043 ± 0.024	**0.038 ± 0.009**	0.053 ± 0.042	0.187 ± 0.015	0.205 ± 0.004
30	0.158 ± 0.016	0.104 ± 0.001	0.228 ± 0.037	0.170 ± 0.064	0.162 ± 0.082	0.161 ± 0.074	**0.154 ± 0.089**	0.263 ± 0.014	0.275 ± 0.007
40	0.336 ± 0.030	0.141 ± 0.002	0.405 ± 0.049	0.306 ± 0.091	0.309 ± 0.087	**0.293 ± 0.082**	0.304 ± 0.102	0.326 ± 0.011	0.337 ± 0.005
50	0.493 ± 0.021	0.206 ± 0.001	0.546 ± 0.055	0.432 ± 0.116	0.481 ± 0.079	0.483 ± 0.093	0.410 ± 0.124	**0.382 ± 0.009**	0.389 ± 0.007
70	0.709 ± 0.035	0.343 ± 0.002	0.727 ± 0.085	0.630 ± 0.128	0.635 ± 0.080	0.695 ± 0.074	0.643 ± 0.103	0.501 ± 0.005	**0.497 ± 0.006**
90	0.812 ± 0.057	0.484 ± 0.001	0.788 ± 0.093	0.740 ± 0.090	0.712 ± 0.102	0.756 ± 0.074	0.719 ± 0.149	0.625 ± 0.027	**0.589± 0.006**
mean	0.364	0.220	0.396	0.336	0.335	0.347	**0.327**	0.348	0.350

Table 4: Mean source test error over 10 runs

Degree	SVM	TSVM	RBF-landmarks	PB-landmarks	PBDA-landmarks	PB-landmarks^{RCV}	PBDA-landmarks^{RCV}	DALC	PBDA
10	**0.000 ± 0.000**	0.174 ± 0.001	0.000 ± 0.000	0.002 ± 0.002	0.001 ± 0.002	0.001 ± 0.000	0.002 ± 0.002	0.613 ± 0.003	0.618 ± 0.002
20	**0.000 ± 0.000**	0.174 ± 0.001	0.000 ± 0.000	0.002 ± 0.002	0.001 ± 0.001	**0.000 ± 0.000**	0.001 ± 0.002	0.611 ± 0.005	0.618± 0.002
30	**0.000 ± 0.000**	0.173 ± 0.001	0.000 ± 0.000	0.002 ± 0.002	0.001 ± 0.001	**0.000 ± 0.000**	0.001 ± 0.002	0.614 ± 0.004	0.619 ± 0.002
40	0.000 ± 0.001	0.172 ± 0.002	**0.000 ± 0.000**	0.002 ± 0.002	0.001 ± 0.001	**0.000 ± 0.000**	0.002 ± 0.002	0.614 ± 0.004	0.618 ± 0.002
50	**0.000 ± 0.000**	0.172 ± 0.002	0.000 ± 0.000	0.002 ± 0.002	0.001 ± 0.001	**0.000 ± 0.000**	0.001 ± 0.002	0.615 ± 0.004	0.618 ± 0.004
70	0.000 ± 0.001	0.170 ± 0.002	**0.000 ± 0.000**	0.002 ± 0.002	0.001 ± 0.001	0.001 ± 0.002	**0.000 ± 0.000**	0.619 ± 0.002	0.617 ± 0.003
90	**0.000 ± 0.000**	0.169 ± 0.001	0.000 ± 0.000	0.002 ± 0.002	0.001 ± 0.001	0.002 ± 0.002	**0.000 ± 0.000**	0.634 ± 0.021	0.611 ± 0.002
mean	**0.000**	0.172	0.000	0.002	0.001	0.001	0.001	0.617	0.617

5.5 Amazon product review dataset

For a final evaluation of our proposed learning procedure we consider the Amazon product review dataset [Blitzer et al., 2007]. It is composed of the reviews of user on four types of products books, DVDs, electronics, and kitchen appliances from Amazon.com. Those reviews are associated with five stars ratings that are binarized with the preprocessing used in [Chen et al., 2011], a review is considered positive if it has 3 or more stars ($y = +1$) and negative in the other case ($y = -1$). Moreover, this preprocessing makes the data go from 100000 features, representing the bag of words encoding of the reviews, to approximately 40000 by only keeping the ones that appear at least ten times for each learning task.

We consider two data set-up, that correspond to two domain adaptation tasks. In the first one we take the books review as the source domain and the DVDs reviews as the target domain. In the second one we take the books review as the source domain and the kitchen appliances reviews as the target domain. In both case the learning is done with 2000 labeled train source data and 2000 unlabeled train target data. Then for the DVDs ratings we consider 3586 target instances to test the performance and for the kitchen appliances ratings we have 5945 instances for test. For both domain 400 validation data are drawn from the train data to run the appropriate validation procedure for each algorithm may it be classical cross validation or reverse validation. The limitation to only two domain adaptation task is due to a lack of available time to run the experiments caused to the complexity of the task

Only the results for random landmark selection are reported for the present learning scenario. The γ parameter is fixed in this scenario at value 0.1 because of limitation in computation capacity. For the same reason the penalisation constant C used for every protocol relying on a SVM is fixed at value 1. When landmarks are involved we have $n_l \in \{2, 4, 8, 16\}$ and if the model uses random Fourier features their number D is selected from $D \in \{4, 8, 16\}$. The ranges of the hyper-parameters specific to algorithms are the following,

PB-Landmarks : from Equation (25), the trade off parameter $\beta \in \{10^{-3}, 10^{1}, 10^{3}\}$.

PBDA-Landmarks : The parameters β_∞, c, b from optimization problem in Equation (60), we get them in the ranges $\beta_\infty \in 1$ and c, $b \in \{10^{-2}, 10^{0}, 10^{1}\}$.

DALC : from Equation (39) $C \in [10^{-2}, 10^{0}, 10^{1}]$ and $B \in [10^{-2}, 10^{0}, 10^{1}]$.

PBDA : from Equation (38) $\Omega \in [10^{-2}, 10^{0}, 10^{1}]$ and $A \in [10^{-2}, 10^{0}, 10^{1}]$.

5.5.1 Results

For this learning scenario the target accuracy given in Table 5 is disappointing. PBDA-Landmarks performs drastically worse than PBDA and DALC is even worse than PB-Landmarks. This indicates that little to no adaptation to target dataset is done and our protocol has no interest over a similar one like PB-Landmarks even though it is not designed for Domain Adaptation at all.

Table 5: Mean target test error

Dataset	SVM	TSVM	DALC	PBDA	PB-landmarks$^{(RCV)}$	PBDA-landmarks$^{(RCV)}$
b to d	0.492	0.496	0.173	**0.172**	0.420	0.458
b to k	0.503	0.503	**0.245**	**0.245**	0.465	0.452
mean	0.497	0.499	0.209	**0.208**	0.442	0.455

6 Conclusion

We developed a new PAC-Bayesian bound by bridging between the PAC-Bayesian approach of Kernel approximation via Random Fourier Features developped by [Letarte et al., 2019] and the results of [Germain et al., 2020] in the specification of the PAC-Bayesian framework to Domain Adaptation. This bound guarantees the alignment of a learned pseudo-kernel in the unlabeled target domain, assuring the quality of the representation of target data points even in the absence of labels. Motivated by the limitation of the bound to a single source data point we derived a landmark learning procedure consisting in learning a pseudo-kernel for every point in a fixed group of source instances. These pseudo-kernels enables a new representation of the data from both the source and the target domain that minimizes the real risk of the kernel alignment in the target domain.

The comparison of our method with PAC-Bayesian domain adaptation protocols PBDA [Germain et al., 2013a] and DALC [Germain et al., 2013b] shown satisfying result on the toy problem of the two intertwining moons. However when faced with real data in the sentiment analysis task for Amazon reviews [Blitzer et al., 2007] the model failed, showing performance even worse then a protocol not designed for Domain Adaptation in the case of PB-Landmarks [Letarte et al., 2019]. This results might be partially improved with a better hyper-parameter tuning making the learning more adapted for the problem at hand. However, the fact that the performances of PBDA-Landmarks underclass the ones of PB-Landmarks indicate a failure in the process.

To go further on the analysis developed here it might be interesting to evaluate the evolution of the performances associated with increasing number of landmarks. Indeed, the number of landmarks and features might have been underestimated by a great amount for the real data experiment since the dataset consists of thousands of points and we drawn at most 16 landmarks. The same could be done with the number of features used in the kernel approximation, the range used here might insufficient to capture the

richness of the dataset descriptors.

Another key point of interest is the dependence of the proposed bound to a fixed data point. If this limitation were lifted it would be possible to learn a single kernel and to apply it with to every target data-point without relying on landmark points. The advantage of this would be that the learning process could take advantage of the entire dataset instead of relying on limited number of points. This issue was solved by [Letarte et al., 2019] using U-statistics and f-divergence and a similar approach might be designed with the bound developped here.

References

[Alquier, 2023] Alquier, P. (2023). User-friendly introduction to pac-bayes bounds.

[Ambroladze et al., 2006] Ambroladze, A., Parrado-Hernández, E., and Shawe-Taylor, J. (2006). Tighter pac-bayes bounds. *Advances in neural information processing systems*, 19.

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

[Blitzer et al., 2007] Blitzer, J., Dredze, M., and Pereira, F. (2007). Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pages 440–447.

[Blumer et al., 1989] Blumer, A., Ehrenfeucht, A., Haussler, D., and Warmuth, M. K. (1989). Learnability and the vapnik-chervonenkis dimension. *Journal of the ACM (JACM)*, 36(4):929–965.

[Bochner, 1933] Bochner, S. (1933). Monotone funktionen, stieltjessche integrale und harmonische analyse. *Mathematische Annalen*, 108:378–410.

[Bousquet and Elisseeff, 2000] Bousquet, O. and Elisseeff, A. (2000). Algorithmic stability and generalization performance. *Advances in Neural Information Processing Systems*, 13.

[Bousquet and Elisseeff, 2002] Bousquet, O. and Elisseeff, A. (2002). Stability and generalization. *The Journal of Machine Learning Research*, 2:499–526.

[Byrd et al., 1999] Byrd, R. H., Hribar, M. E., and Nocedal, J. (1999). An interior point algorithm for large-scale nonlinear programming. *SIAM Journal on Optimization*, 9(4):877–900.

[Chen et al., 2011] Chen, M., Weinberger, K. Q., and Blitzer, J. (2011). Co-training for domain adaptation. *Advances in neural information processing systems*, 24.

[Farahani et al., 2021] Farahani, A., Voghoei, S., Rasheed, K., and Arabnia, H. R. (2021). A brief review of domain adaptation. *Advances in data science and information engineering: proceedings from ICDATA 2020 and IKE 2020*, pages 877–894.

[Germain et al., 2017] Germain, P., Bach, F., Lacoste, A., and Lacoste-Julien, S. (2017). Pac-bayesian theory meets bayesian inference.

[Germain et al., 2013a] Germain, P., Habrard, A., Laviolette, F., and Morvant, E. (2013a). A pac-bayesian approach for domain adaptation with specialization to linear classifiers. In Dasgupta, S. and McAllester, D., editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 738–746, Atlanta, Georgia, USA. PMLR.

[Germain et al., 2013b] Germain, P., Habrard, A., Laviolette, F., and Morvant, E. (2013b). A pac-bayesian approach for domain adaptation with specialization to linear classifiers. In Dasgupta, S. and McAllester, D., editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 738–746, Atlanta, Georgia, USA. PMLR.

[Germain et al., 2020] Germain, P., Habrard, A., Laviolette, F., and Morvant, E. (2020). Pac-bayes and domain adaptation. *Neurocomputing*, 379:379–397.

[Germain et al., 2009a] Germain, P., Lacasse, A., Laviolette, F., and Marchand, M. (2009a). Pac-bayesian learning of linear classifiers. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 353–360, New York, NY, USA. Association for Computing Machinery.

[Germain et al., 2015] Germain, P., Lacasse, A., Laviolette, F., Marchand, M., and Roy, J.-F. (2015). Risk bounds for the majority vote: From a pac-bayesian analysis to a learning algorithm.

[Germain et al., 2009b] Germain, P., Lacasse, A., Marchand, M., Shanian, S., and Laviolette, F. (2009b). From pac-bayes bounds to kl regularization. *Advances in neural information processing systems*, 22.

[Guedj, 2019] Guedj, B. (2019). A primer on pac-bayesian learning. *arXiv preprint arXiv:1901.05353*.

[Haussler and Warmuth, 2018] Haussler, D. and Warmuth, M. (2018). The probably approximately correct (pac) and other learning models. *The Mathematics of Generalization*, pages 17–36.

[Hofmann et al., 2008] Hofmann, T., Schölkopf, B., and Smola, A. J. (2008). Kernel methods in machine learning.

[Koltchinskii and Panchenko, 2000] Koltchinskii, V. and Panchenko, D. (2000). Rademacher processes and bounding the risk of function learning. In *High dimensional probability II*, pages 443–457. Springer.

[Langford and Shawe-Taylor, 2002] Langford, J. and Shawe-Taylor, J. (2002). Pac-bayes & margins. *Advances in neural information processing systems*, 15.

[Letarte et al., 2019] Letarte, G., Morvant, E., and Germain, P. (2019). Pseudo-bayesian learning with kernel fourier transform as prior. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 768–776. PMLR.

[Lever et al., 2010] Lever, G., Laviolette, F., and Shawe-Taylor, J. (2010). Distribution-dependent pac-bayes priors. In *International Conference on Algorithmic Learning Theory*, pages 119–133. Springer.

[Lever et al., 2013] Lever, G., Laviolette, F., and Shawe-Taylor, J. (2013). Tighter pac-bayes bounds through distribution-dependent priors. *Theoretical Computer Science*, 473:4–28.

[Liao et al., 2020] Liao, R., Urtasun, R., and Zemel, R. (2020). A pac-bayesian approach to generalization bounds for graph neural networks. *arXiv preprint arXiv:2012.07690*.

[McAllester, 2003] McAllester, D. (2003). Simplified pac-bayesian margin bounds. In *Learning Theory and Kernel Machines: 16th Annual Conference on Learning Theory and 7th Kernel Workshop, COLT/Kernel 2003, Washington, DC, USA, August 24-27, 2003. Proceedings*, pages 203–215. Springer.

[McAllester, 1998] McAllester, D. A. (1998). Some pac-bayesian theorems. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 230–234.

[McDonald, 2009] McDonald, G. C. (2009). Ridge regression. *Wiley Interdisciplinary Reviews: Computational Statistics*, 1(1):93–100.

[Mercer, 1909] Mercer, J. (1909). Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 209:415–446.

[Pan and Yang, 2010] Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

[Perez-Ortiz et al., 2021] Perez-Ortiz, M., Rivasplata, O., Guedj, B., Gleeson, M., Zhang, J., Shawe-Taylor, J., Bober, M., and Kittler, J. (2021). Learning pac-bayes priors for probabilistic neural networks. *arXiv preprint arXiv:2109.10304*.

[Rahimi and Recht, 2007] Rahimi, A. and Recht, B. (2007). Random features for large-scale kernel machines. *Advances in neural information processing systems*, 20.

[Rasmussen, 2003] Rasmussen, C. E. (2003). Gaussian processes in machine learning. In *Summer school on machine learning*, pages 63–71. Springer.

[Redko et al., 2022] Redko, I., Morvant, E., Habrard, A., Sebban, M., and Bennani, Y. (2022). A survey on domain adaptation theory: learning bounds and theoretical guarantees.

[Rivasplata et al., 2019] Rivasplata, O., Tankasali, V. M., and Szepesvari, C. (2019). Pac-bayes with backprop. *arXiv preprint arXiv:1908.07380*.

[Rudin, 2017] Rudin, W. (2017). *Fourier analysis on groups*. Courier Dover Publications.

[Schölkopf et al., 1997] Schölkopf, B., Smola, A., and Müller, K.-R. (1997). Kernel principal component analysis. In *International conference on artificial neural networks*, pages 583–588. Springer.

[Shawe-Taylor and Williamson, 1997] Shawe-Taylor, J. and Williamson, R. C. (1997). A pac analysis of a bayesian estimator. In *Proceedings of the tenth annual conference on Computational learning theory*, pages 2–9.

[Torrey and Shavlik, 2010] Torrey, L. and Shavlik, J. (2010). Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pages 242–264. IGI global.

[Valiant, 1984] Valiant, L. G. (1984). A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142.

[Vapnik, 1999] Vapnik, V. (1999). *The nature of statistical learning theory*. Springer science & business media.

[Vapnik and Chervonenkis, 1982] Vapnik, V. N. and Chervonenkis, A. Y. (1982). Necessary and sufficient conditions for the uniform convergence of means to their expectations. *Theory of Probability and Its Applications*, 26:532–553.

[Vovk, 2013] Vovk, V. (2013). Kernel ridge regression. In *Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik*, pages 105–116. Springer.

[Wold et al., 1987] Wold, S., Esbensen, K., and Geladi, P. (1987). Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52.

[Zhong et al., 2010] Zhong, E., Fan, W., Yang, Q., Verscheure, O., and Ren, J. (2010). Cross validation framework to choose amongst models and datasets for transfer learning. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2010, Barcelona, Spain, September 20-24, 2010, Proceedings, Part III 21*, pages 547–562. Springer.

Appendix

Lemma 1. *Let* $\mathbf{a}$ *and* $\mathbf{b}$ *be vectors in* $\mathbb{R}^n$*, and for any positive real numbers* p *and* q *such that* $\frac{1}{p} + \frac{1}{q} = 1$*, Hölder's Inequality states:*

$$\|\mathbf{a} \cdot \mathbf{b}\|_1 \leq \|\mathbf{a}\|_p \|\mathbf{b}\|_q$$

Where the p-norm $\|\mathbf{a}\|_p$ *is defined as:*

$$\|\mathbf{a}\|_p = \left(\sum_{i=1}^{n} |a_i|^p \right)^{\frac{1}{p}}$$