JMLR: Workshop and Conference Proceedings 29:1–16, 2013 ACML 2013

Using Hyperbolic Cross Approximation to measure and compensate Covariate Shift

Thomas Vanck VANCK@MATH.TU-BERLIN.DE
Institut für Mathematik, Technische Universität Berlin

Jochen Garcke GARCKE@INS.UNI-BONN.DE
Institut für Numerische Simulation, Universität Bonn and Fraunhofer SCAI, Sankt Augustin

Editor: Cheng Soon Ong and Tu Bao Ho

Abstract

The concept of covariate shift in supervised data analysis describes a difference between the training and test distribution while the conditional distribution remains the same. To improve the prediction performance one can address such a change by using individual weights for each training datapoint, which emphasizes the training points close to the test data set so that these get a higher significance. We propose a new method for calculating such weights by minimizing a Fourier series approximation of distance measures, in particular we consider the total variation distance, the Euclidean distance and Kullback-Leibler divergence. To be able to use the Fourier approach for higher dimensional data, we employ the so-called hyperbolic cross approximation. Results show that the new approach can compete with the latest methods and that on real life data an improved performance can be obtained.

Keywords: Covariate Shift, Fourier Series Approximation, Hyperbolic Cross, Curse of Dimensionality

1. Introduction

In a standard machine learning setting it is assumed that the test data is essentially drawn from the same distribution as the training data, i.e.

$$p(x, y) = p(y|x)p(x) \text{ and } p(x) = p_{tr}(x) = p_{te}(x).$$

In practice, however, the training distribution p_{tr} can differ significantly from the test distribution p_{te}, while the functional relationship $p(y|x)$ remains the same. Such a situation, where $p_{tr}(x) \neq p_{te}(x)$, is known as covariate shift and can arise from several circumstances. For example, in non stationary cases the distribution of the covariate might change over time. In parameter optimization scenarios a good extrapolation into before unseen regions of the parameter domain is often needed. Or the actual (geographical) location of some measurements may have an impact, as is the case for the earthquake dataset considered in this paper.

The covariate shift is the result of some kind of bias that influences the input variables x. Hence, the test datapoints are drawn from regions that are not covered, or by far not as densely, by the training samples. Therefore, a model learned on the training data might not be well suited for prediction of the test labels. To rectify this problem one can put

more weight on training datapoints that lie close to the test data X_{te}, assuming that these better represent the structure of the test data. Note that in supervised learning such a situation where, besides the training data, additional samples are available for whose only the locations x are given is known as semi-supervised learning (Chapelle et al., 2006).

A common approach to handle such a situation is importance sampling. If one knew the training and test distribution one could give weights directly by calculating the importance sampling weight function

$$w(x) = \frac{p_{te}(x)}{p_{tr}(x)}. \tag{1}$$

Since this information is unavailable it is necessary to estimate these weights.

Several methods have recently been proposed for inferring individual weights for each training datapoint. Bickel et al. (2009) proposed a kernel logistic regression classifier for covariate shift. Another method is the so-called Kernel Mean Matching (KMM) (Huang et al., 2007) algorithm, where all moments in the original space are mean matched. KLIEP (Kullback Leibler Importance Estimation Procedure) has been put forward by Sugiyama et al. (2008), it minimizes the Kullback Leibler divergence for retrieving optimal weights. Furthermore this procedure was extended by Tsuboi et al. (2009) to large-scale problems. Least-squares importance fitting (uLSIF) is another recent method which was stated by Kanamori et al. (2009). For recent surveys on the state of the art on covariate shift as well as the more general dataset shift see Quionero-Candela et al. (2009); Sugiyama and Kawanabe (2012); Moreno-Torres et al. (2012). All these methods assume some overlap of the samples from the two distributions p_{tr} and p_{te}. In cases where the training and test distribution have nothing in common, i.e. the samples are disjunct, it will not be possible to derive reasonable information for the calculation of the weights just from the samples without strong additional assumptions about the type of distributions involved.

In this paper we propose a new approach for estimating the weights. We use a Fourier approximation of a distance measure to estimate the divergence of distributions. In a certain sense the measuring of the divergence becomes less data centered since an explicit discretization of the underlying error function is involved. The Fourier based approach does not depend on a specific distance measure, nor on a specific point set for empirically estimating the distance measure. We will minimize the total variation distance, the Kullback-Leibler divergence and the Euclidean distance. The training and test data are then used during the estimation of the Fourier coefficents of the resulting distance function. It can be seen that the resulting constrained optimization problem is convex and can be solved with standard methods. Furthermore, we give some evidence that under certain circumstances the application of the Fourier series will lead to a better weight estimation in comparison to other approaches. Note that a Fourier series approximation for high dimensional functions quickly runs into the curse of dimensionality due to the exponential growth of the number of coefficients. To overcome this we will apply the hyperbolic cross approach (Babenko, 1960; Smolyak, 1963; Knapek, 2000) which enables us to apply a Fourier series approximation to high dimensional functions by simultaneously keeping an acceptable degree of accuracy.

The paper is structured in the following way: In section 2 we introduce our new method and section 3 gives insights why the new method is beneficial. The extension to higher dimensional data based on the hyperbolic cross approximation is given in section 4. Finally,

sections 5 and 6 state the employed weighted regression and classification algorithms, the experimental setup and the results obtained on diverse datasets.

2. New Fourier Based Approach

We will now motivate and derive our new approach for the calculation of importance weights for the training data. Mathematically speaking we would like to minimize the distance of the test distribution p_{te} and the training distribution p_{tr} which is reweighted by w

$$\min_w D(p_{te}(x) \| w(x) p_{tr}(x)). \tag{2}$$

Expression (2) can be minimized using different distance measures. Typically one chooses divergence measures from the classes of Csiszár or Bregman divergences, which are then empirically evaluated on some points $\{x_i\}_{i=1}^N$. Here one often uses training $\{x_i^{tr}\}_{i=1}^{N_{tr}}$ or test datapoints $\{x_i^{te}\}_{i=1}^{N_{te}}$ for the evaluation points in the distance estimation.

Let us consider the class of Csiszár divergences, defined as $D_h(p||q) = \sum_{i=1}^N q_i h(\frac{p_i}{q_i})$, where h is a real-valued convex function satisfying $h(1) = 0$ and we define $p_i := p(x_i)$, $q_i := q(x_i)$. Different h yield different divergences. For the following exposition we set $h(u) = |u - 1|$ and considering that $q_i > 0 \; \forall i$ we get the total variation distance

$$D_h(p||q) = \sum_{i=1}^N q_i \left| \frac{p_i}{q_i} - 1 \right| = \sum_{i=1}^N |p_i - q_i|. \tag{3}$$

Substituting (3) into (2) we get

$$\min_w D_h(p_{te}(x) \| w(x) p_{tr}(x)) = \min_w \sum_{i=1}^N |p_{te}(x_i) - w(x_i) p_{tr}(x_i)|. \tag{4}$$

Note that in contrast to many other approaches, our methodology does not depend on a specific choice of the points $\{x_i\}_{i=1}^N$ and we are able to use any point set in the distance estimation (4). Nevertheless, for the sake of comparison with other approaches, we use either training or test datapoints in (4) for our experiments in Section 6.

Observe that the Fourier based approach which we will describe in the following can be directly applied to different divergence measures. For example, a generalisation of the total variation distance, the so called Matsusita or Hellinger distance, i.e. $h(u) = |u^\gamma - 1|^{\frac{1}{\gamma}}$ which yields $\sum_{i=1}^N |p_i^\gamma - q_i^\gamma|^{\frac{1}{\gamma}}$, could be used. We later state our approach with the Kullback-Leibler divergence and the Euclidean distance, respectively.

2.1. Choice of the Weight Function

The optimization problem (2) states the problem of finding an optimal weight function $w(x)$ which minimizes the distance of the two functions p_{te} and $w \cdot p_{tr}$. The exact solution would be the quotient of the density functions, i.e. $w(x) = \frac{p_{te}(x)}{p_{tr}(x)}$, which of course is not available. Therefore one can only compute an approximation $\hat{w}$ of w. For the discrete representation

of $\hat{w}$ we will, as in Sugiyama et al. (2008); Kanamori et al. (2009), use a linear combination of Gaussian kernels

$$\hat{w}(x, \alpha) = \sum_{j=1}^{Z} \alpha_j \exp\left(-\frac{\|x - \zeta_j\|^2}{2\sigma^2}\right). \tag{5}$$

It is comprised of $Z \in \mathbb{N}$ exponential functions each of which centered at a ζ_j. In our experiments we will use the test data as the center points ζ_j, as in Sugiyama et al. (2008); Kanamori et al. (2009). There it is argued that using test points as the Gaussian centers is preferable, since kernels may be needed where the target function $w(x)$ is large, which is the case where the training density $p_{tr}(x)$ is small and the test density $p_{te}(x)$ is large. Note that the ratio (1) implies positive weights, which is the case for any x and any $\alpha \geq 0$ in $\hat{w}(x, \alpha)$. Other weight function representations are possible, but to concentrate on the effect of the new Fourier based distance estimation and to be able to better compare with other approaches we consider the linear combination of Gaussian kernels in this work.

Inserting (5) into (4) now yields

$$\min_{w} D_h(p_{te}(x) \| w(x) p_{tr}(x)) \approx \min_{\alpha \geq 0} \sum_{n=1}^{N} |p_{te}(x_n) - \hat{w}(x_n, \alpha) p_{tr}(x_n)|. \tag{6}$$

Note that this minimization problem still employs the probability densities directly. In the next step we will now approximate this term using a Fourier series approximation.

2.2. Fourier Series Approximation

Our new approach makes use of Fourier series approximation, with which we discretize the employed distance measure, taking a more function centric view as opposed to the more common data centric view. Section 3 provides a discussion of the advantages of this new approach, while section 4 explains the case of more than one dimension.

Let now f be a continuous periodic function with period $T > 0$ and partially continuous derivatives; then the Fourier series is defined as

$$f(x) = \sum_{k=-\infty}^{\infty} c_k e^{i\frac{2\pi k}{T}x}, \qquad c_k = \frac{1}{T}\int_{t}^{t+T} f(x) e^{-i\frac{2\pi k}{T}x} dx, \tag{7}$$

where i denotes the imaginary unit and $t \in \mathbb{R}$ is an arbitrary point. For a suitably smooth function we can approximate this expression in a controlled fashion by a truncated Fourier series with $|k| \leq K$

$$f(x) \approx \sum_{k=-K}^{K} c_k e^{i\frac{2\pi k}{T}x}, \tag{8}$$

where K is chosen to achieve a given error, see section 4 for more details on the approximation properties.

We consider now the error function between the two densities

$$f(x) := p_{te}(x) - \hat{w}(x_n, \alpha) p_{tr}(x).$$

We assume that the given data is bounded to a certain region, i.e. $X_{tr} \cup X_{te} \subset [t, t+T] \subset \mathbb{R}$, for suitable chosen t, T. Assuming periodicity of f on that interval implies that we make the same small error on the boundary, which is our aim in the minimization. Furthermore, the interesting region is the inner part where the two samples overlap, near the boundary of the domain the densities will be small in any case, which, if necessary, can even be enforced by having a reasonable gap between the given data and the actual boundary of the interval. Therefore we can reasonably assume a continuous periodic extension of the Fourier series of f and avoid the Gibbs phenomen, i.e. potential overshoots on the boundary, in practice.

We now apply the Fourier series approximation to our problem (6). Due to its definition we can replace the densities by the empirical samples in the formula (7) for the coefficients c_k after splitting the integral into two

$$c_k(\alpha) = \frac{1}{T}\int_t^{t+T} p_{te}(x) e^{-i\frac{2\pi k}{T}x} dx - \frac{1}{T}\int_t^{t+T} \hat{w}(x,\alpha) p_{tr}(x) e^{-i\frac{2\pi k}{T}x} dx \tag{9}$$

$$\approx \frac{1}{TN_{te}} \sum_{l=1}^{N_{te}} e^{-i\frac{2\pi k}{T}x_l^{te}} - \frac{1}{TN_{tr}} \sum_{l=1}^{N_{tr}} \hat{w}(x_l^{tr}, \alpha) e^{-i\frac{2\pi k}{T}x_l^{tr}}. \tag{10}$$

In the last part of this equation we approximate the two integrals by taking the empirical expectation based on the training and test data, respectively. Therefore, we no longer explicitly need the unknown densities but use their known samples.

2.3. Optimization Problem

The original problem (2) is about finding an appropriate weight function. Employing (5) for given parameter σ and center points $(\zeta_j)_{j=1}^Z$ and using the Fourier approximation (8) for a suitably chosen K we obtain the following optimization problem

$$\min_{\alpha \geq 0} \sum_{n=1}^{N} |p_{te}(x_n) - \hat{w}(x_n, \alpha) p_{tr}(x_n)| \approx \min_{\alpha \geq 0} \sum_{n=1}^{N} \left| \sum_{k=-K}^{K} c_k(\alpha) e^{i\frac{2\pi k}{T}x_n} \right|. \tag{11}$$

Due to the linearity of this problem we can express it in matrix notation. Defining the matrix $A \in \mathbb{R}^{N \times Z}$ as $A = [A_1 | \dots | A_N]$, where the $A_n \in \mathbb{R}^Z$ are column vectors comprised, after inserting (5) for $\hat{w}$, of the entries

$$(A_n)_j = \sum_{k=-K}^{K} \frac{1}{TN_{tr}} \sum_{l=1}^{N_{tr}} e^{-\frac{\|x_l^{tr} - \zeta_j\|^2}{2\sigma^2}} e^{-i\frac{2\pi k}{T}x_l^{tr}} e^{i\frac{2\pi k}{T}x_n}, \qquad j = 1, \dots, Z.$$

Additionally we get a vector $b \in \mathbb{R}^N$, defined as

$$b_n = \sum_{k=-K}^{K} \sum_{l=1}^{N_{te}} \frac{1}{TN_{te}} e^{-i\frac{2\pi k}{T}x_l^{te}} e^{i\frac{2\pi k}{T}x_n}, \qquad n = 1, \dots, N.$$

The problem (11) can now be stated as a L_1 minimization problem with side conditions in a compact notation by employing A and b

$$\min_{\alpha \geq 0} \|A\alpha - b\|_1 .$$

2.4. Normalization Constraints

It is possible that a solution to the optimization problem (11) might not yield appropriate weights. Often only a small fraction of αs will be larger than zero, which leads to a situation where only a few training datapoints will get importance. To compensate, we employ an approach which is similar to the one introduced in Sugiyama et al. (2008). From (1) we have $p_{te}(x) = w(x)p_{tr}(x)$, and taking the integral on both sides yields the natural side condition

$$1 = \int p_{te}(x)dx = \int w(x)p_{tr}(x)dx \approx \frac{1}{N_{tr}} \sum_{n=1}^{N_{tr}} \hat{w}(x_n^{tr}, \alpha),$$

again using the empirical samples and the approximation $\hat{w}$. We augment (11) and get a new constrained optimization problem[1]

$$\min_{\alpha \geq 0} \|A\alpha - b\|_1 \text{ s.t. } \frac{1}{N_{tr}} \sum_{n=1}^{N_{tr}} \hat{w}(x_n^{tr}, \alpha) = 1. \tag{12}$$

2.5. Kullback-Leibler Divergence

An advantage of the Fourier approach is that it can directly be applied to different divergence measures. To demonstrate this flexibility we will use as a second Csiszár divergence the Kullback-Leibler divergence, which also allows us to compare with KLIEP (Sugiyama et al., 2008). Roughly following the KLIEP derivation gives

$$\begin{aligned} \mathrm{KL}(p_{te} \| w p_{tr}) &= \sum_{n=1}^{N} p_{te}(x_n) \log \left(\frac{p_{te}(x_n)}{w(x_n) p_{tr}(x_n)} \right) \\ &= \sum_{n=1}^{N} p_{te}(x_n) \log \left(\frac{p_{te}(x_n)}{p_{tr}(x_n)} \right) - \sum_{n=1}^{N} p_{te}(x_n) \log \left(w(x_n) \right). \end{aligned}$$

Since the first part does not depend on w, it suffices to minimize

$$\arg\min_{w} \mathrm{KL}(p_{te} \| w p_{tr}) \approx \arg\min_{\alpha \geq 0} - \sum_{n=1}^{N} p_{te}(x_n) \log \left(\hat{w}(x_n, \alpha) \right), \tag{13}$$

where we employ the approximation $\hat{w}$ of w. Using the same normalization approach as above, the final optimization problem becomes

$$\min_{\alpha \geq 0} \sum_{n=1}^{N} \sum_{k=-K}^{K} c_k(\alpha) e^{i \frac{2\pi k}{T} x_n} \text{ s.t.} \sum_{n=1}^{N_{tr}} \frac{\hat{w}(x_n^{tr}, \alpha)}{N_{tr}} = 1, \tag{14}$$

where

$$c_k(\alpha) = \frac{1}{T} \int_t^{t+T} -p_{te}(x) \log \left(\hat{w}(x, \alpha) \right) e^{-i \frac{2\pi k}{T} x} dx \approx \frac{-1}{T N_{te}} \sum_{l=1}^{N_{te}} \log \left(\hat{w}(x_l^{te}, \alpha) \right) e^{-i \frac{2\pi k}{T} x_l^{te}}.$$

1. We used the YALL1 Basic solver from http://yall1.blogs.rice.edu/.

Although the approach is very similar to the one suggested by Sugiyama et al. (2008), we get a different optimization problem[2] due to the Fourier approximation and also estimate the divergence in a different fashion. Note that the KL divergence is a special case of the generalized KL divergence or I-Divergence which is from the class of Bregman divergences. The Fourier approach could also be applied for these.

2.6. Euclidean Distance

The third distance measure that we will investigate is the Euclidean distance, which belongs to the class of Bregman divergences, and was also used for uLSIF (Kanamori et al., 2009). Bregman divergences are defined by

$$D_\phi(p\|q) = \phi(p) - \phi(q) - \phi'(q)(p-q),$$

where ϕ is a strictly convex real-valued function and $\phi'(q)$ denotes the derivative with respect to q. Setting $\phi(\cdot) = ||\cdot||_2^2$ we get

$$D_{||\cdot||_2^2}(p\|q) = ||p||_2^2 - ||q||_2^2 - 2q(p-q) = ||p-q||_2^2.$$

Employing the data, the weight function $\hat{w}$ and applying the Fourier approximation we get the following optimization problem:

$$\min_{\alpha \geq 0} \|A\alpha - b\|_2^2 \quad \text{s.t.} \quad \frac{1}{N_{tr}} \sum_{n=1}^{N_{tr}} \hat{w}(x_n^{tr}, \alpha) = 1, \tag{15}$$

where A and b are defined as in (2.3).

3. Benefits of the Fourier Approximation

The following illustrative example shows the behavior of our approach. The weight function $\hat{w}$ is chosen according to (5). For the sake of comparison, we use the Kullback-Leibler divergence and the Euclidean distance here.

We note that by estimating the divergence measure using the Fourier approximation we achieve a smoothing of the weights. This becomes especially useful when a small bandwidth parameter σ is chosen for the weight function $\hat{w}$. As Figure 1 illustrates the weights learned by the Fourier methods are much smoother and stable than the weights learned by KLIEP (Sugiyama et al., 2008) or uLSIF (Kanamori et al., 2009) which involve a much higher volatility. In the case of Figure 1 we applied the bandwidth parameter σ that was chosen by KLIEP also to the Fourier methods for the sake of comparison. Although we did not apply our method of parameter selection, the Fourier methods outperform KLIEP, in the sense of a less volatile weight function. Note that the parameters for uLSIF have been determined by its own parameter estimation method.

The comparison of KLIEP and KL-Fourier is of special interest here because this is a direct comparison of two very similar methods which clearly shows the advantages of the

2. We here used IPOpt from https://projects.coin-or.org/Ipopt, which is also used for the Euclidean distance.

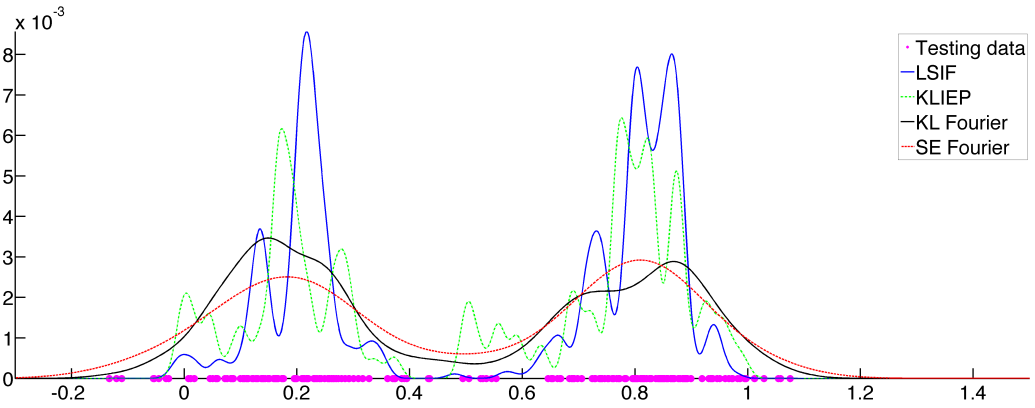

Figure 1: Plot of the learned weight functions $\hat{w}$ for a 1D toy example. Regions of low test data (magenta) density imply small weights while high density regions imply large weights. KLIEP (green) and uLSIF (blue) compute much more volatile weight functions than the Fourier methods. Total variation Fourier was omitted in this plot for clarity of the diagram and due to similarity to the other shown Fourier results.

smoothing of the Fourier approximation method. The reason for this smoothing is that the Fourier approximation only takes low frequencies into account that contain the relevant information for learning good weights. High frequencies are ignored, which usually pay more attention to noisy data that does not positively contribute to the learning of weights. Therefore we are able to learn more appropriate weights.

Another property of our approach is that we are able to estimate the distance measure on any point set. The training or the test data are just one way, and not the only set of locations where to estimate the distances (12), (14), or (15). Merely the computation of the Fourier coefficients requires the training and test data. This is possible since we firstly empirically estimate the distance and secondly apply the Fourier approximation. This is for example different to KLIEP where the error is calculated on the test data and cannot be straightforwardly computed on the training data, or uLSIF where the training and test data points need to be employed in a specific way. One can argue, that to compute suitable weights for the purpose of weighted regression, a divergence estimation on the training data is beneficial since the weights are employed for the training data in the regression algorithm and therefore for those points the distance should be small. This hypothesis is supported in section 6, where we calculated the distance using the training or the test data for each Fourier method.

4. Hyperbolic Cross Approximation

Until now, for a simplified exposition, we have only considered a one dimensional Fourier series. The straightforward d-dimensional generalisation of a Fourier approximation for

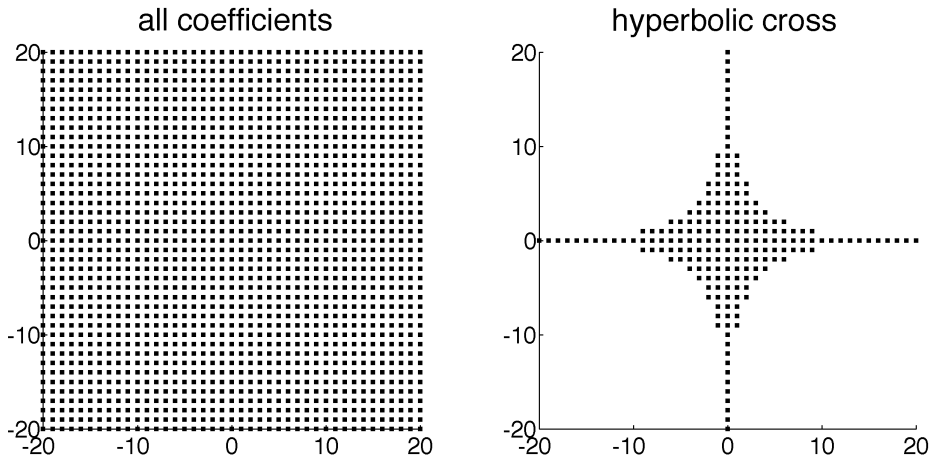

Figure 2: Used Fourier coefficients in a two dimensional example. Each point denotes a frequency combination (k_1, k_2). Left: Standard Fourier series approximation for $|k| \leq 20$. Right: Hyperbolic cross approximation FI_{20}.

$f : \mathbb{R}^d \mapsto \mathbb{R}$ by a tensor product approach

$$f(\boldsymbol{x}) = \sum_{k_1=-K}^{K} \dots \sum_{k_d=-K}^{K} c_{k_1,\dots,k_d} e^{i2\pi \sum_{l=1}^{d} \frac{k_l}{T_l} x_l}$$

$$c_{k_1,\dots,k_d} = \frac{1}{\prod_{j=1}^{d} T_j} \int_{t_1}^{t_1+T_1} \dots \int_{t_d}^{t_d+T_d} f(\boldsymbol{x}) e^{-i2\pi \sum_{l=1}^{d} \frac{k_l}{T_l} x_l} d\boldsymbol{x},$$

with $\boldsymbol{x} = (x_1, \dots, x_d)$, runs into the curse of dimensionality: The number of $c_{k_1,\dots,k_d}$ terms grows exponentially with the number of dimensions, i.e. one would have $(1+K)^d$ coefficients. The direct application of a Fourier series approximation to higher dimensional problems is infeasible.

We use instead the so-called *hyperbolic cross approximation*, where under certain assumptions on f, coefficients that make a small contribution to the representation can be identified and omitted (Babenko, 1960; Smolyak, 1963; Knapek, 2000).

Let us define sets of employed coefficients by

$$FI_K := \left\{ \boldsymbol{k} \in \mathbb{Z}^d : \prod_{i=1}^{d} (1 + |k_i|) \leq (1 + K) \right\},$$

where $\boldsymbol{k} := (k_1, \dots, k_d)$, and $K \in \mathbb{N}$. The resulting selection of Fourier coefficients is known as the hyperbolic cross. For illustration we consider the two dimensional case and show for $K = 20$ in Figure 2 the index set for the full Fourier series approximation, i.e. $|\boldsymbol{k}|_\infty \leq K$, and the hyperbolic cross for FI_{20}. In higher dimension the reduction in the number of

Fourier coefficients will be even stronger noticable and is of several orders of magnitude, going from an impossible computation for the full Fourier approximation of degree K to a possible one using the hyperbolic cross approximation based on the set FI_K.

Looking at the number of coefficients in FI_K the advantage in higher dimensions becomes clear:

$$|FI_K| = \mathcal{O}\left((1+K)\left(\log(1+K)\right)^{d-1}\right),$$

instead of $(1+K)^d$ for the standard Fourier approximation, see e.g. Zung (1983); Knapek (2000).

To consider the approximation properties of the hyperbolic cross Fourier approximation we need to introduce generalisations of Sobolev spaces. For $-\infty < s < \infty$ we define

$$\mathcal{H}^s_{mix}(\mathbb{T}^d) := \left\{ f(\boldsymbol{x}) = \sum_{\boldsymbol{k}\in\mathbb{Z}^d} c_k e^{i\boldsymbol{k}\boldsymbol{x}} : \|f(\boldsymbol{x})\|_{\mathcal{H}^s_{mix}} < \infty \right\}$$

$$\|f(\boldsymbol{x})\|^2_{\mathcal{H}^s_{mix}} := \sum_{\boldsymbol{k}\in\mathbb{Z}^d} \prod_{i=1}^{d} (1+|k_i|)^{2s} |c_k|^2,$$

where $\mathbb{T}^d := [0,1]^d$ is the n-dimensional torus which is the same as the n-dimensional cube where opposite faces are identified. Therefore the space $\mathcal{H}^s_{mix}$ is comprised of all functions whose Fourier coefficients $c_{\boldsymbol{k}}$ decay sufficiently fast in the prescribed manner. The space $\mathcal{H}^s_{mix}$ is called Sobolev space with dominating mixed smoothness. Note that $\mathcal{H}^s_{mix} \subset \mathcal{H}^s \subset \mathcal{H}^{s/d}_{mix}$ for $s \geq 0$ and that $\mathcal{H}^s_{mix}(\mathbb{T}^d) = \mathcal{H}^s(\mathbb{T}^1) \otimes \cdots \otimes \mathcal{H}^s(\mathbb{T}^1)$, where $\mathcal{H}^s(\mathbb{T}^1)$ is the standard Sobolev space.

We now can state the approximation properties (proof e.g. in Knapek (2000))

Lemma 1 *Let $t \in \mathbb{N}$, $t < s$, $s \geq 0$, $u \in \mathcal{H}^s_{mix}$, $f(x) = \sum_{\boldsymbol{k}} c_{\boldsymbol{k}} e^{i\boldsymbol{k}\boldsymbol{x}}$ and $f_K(x) = \sum_{\boldsymbol{k}\in FI_K} c_{\boldsymbol{k}} e^{i\boldsymbol{k}\boldsymbol{x}}$, then it holds that*

$$\|f - f_K\|_{\mathcal{H}^t} \leq (1+K)^{t-s} \|f\|_{\mathcal{H}^s_{mix}}.$$

Using a hyperbolic cross we achieve for $f \in \mathcal{H}^s_{mix}$ the same order of approximation as the standard Fourier approximation. However, the number of coefficients is significantly reduced from $\mathcal{O}(1+K)^d$ to $\mathcal{O}\left((1+K)\left(\log(1+K)\right)^{d-1}\right)$, the use of a Fourier series approximation in higher dimensions becomes feasible.

A question is if we can expect that $p(x) - \hat{w}(x)q(x) \in \mathcal{H}^s_{mix}$, which resolves to the question of the smoothness of p and q, since $\hat{w}$ is sufficiently smooth by definition. This is a problem-specific question and in particular depends on the unknown quantities p and q, so one can neither answer this in general, nor for a specific data set a priori. But we can give indications that the assumption $p, q \in \mathcal{H}^s_{mix}$ is warranted, if one expects reasonably smooth probability distributions at all. Firstly, let us note that the mixed Sobolev spaces have an intrinsic tensor product structure with distinguished dimensions, each of which we can relate to a specific attribute of the data set in its d-dimensional domain. This is in contrast to the standard Sobolev space $\mathcal{H}^s$ which only considers isotropic smoothness and has no distinguished dimensions, e.g. the coordinate system could be rotated without changing the

function space. Secondly, note that the spaces $\mathcal{H}^s_{mix}$ are the underlying function spaces for regression and classification approaches based on sparse grids, whose very good empirical performance was shown in recent years (Garcke, 2006; Pflüger, 2010).

5. Weighted Support Vector Regression (WSVR)

The calculated weights assign each training datapoint an amount of importance. High values denote important datapoints, whereas low values stand for less important datapoints. Classification and regression methods need to incorporate this information so that the prediction in regions of heavily weighted training datapoints is more accurate. To make use of this weighting, it is necessary to modify classification and regression methods such that they can employ a weight for each given training datapoint. A modified support vector machine for classification can be found in Huang et al. (2007).

Analogously, we state for regression problems a modified version of a support vector regression (SVR) problem

$$\begin{aligned} \min_{\theta,b,\xi,\xi^*} \frac{1}{2}\|\theta\|^2 + C\sum_{n=1}^{N} \hat{w}(x_n)(\xi_n + \xi_n^*) \\ \text{subject to: } y_n - \theta^t\phi(x_n) - b \le \epsilon + \xi_n \quad \xi_n \ge 0 \\ \theta^t\phi(x_n) + b - y_n \le \epsilon + \xi_n^* \quad \xi_n^* \ge 0. \end{aligned}$$

Here θ and b denote the model parameters and $\hat{w}(x_n)$ are the estimated importance weights. For each datapoint the slack variable ξ and ξ^* is multiplied by $\hat{w}(x_n)$. This implies higher values for large weights and lower values for small weights respectively. Therefore the slack at datapoints with large weights will tend to be lower than those multiplied by small weights, thus causing a lower tolerance to errors on important datapoints. The dual version is

$$\begin{aligned} \max_{a,a^*} y^t(a-a^*) - \epsilon\sum_{n=1}^{N}(a_n + a_n^*) - \frac{1}{2}(a-a^*)^t\boldsymbol{\kappa}(a-a^*) \\ \text{subject to: } 0 \le a \le \hat{w}(x_n)C \quad a \ge 0 \\ 0 \le a^* \le \hat{w}(x_n)C \quad a^* \ge 0 \end{aligned}$$

where $\boldsymbol{\kappa}$ is the empirical kernel map. We will use the Gaussian kernel in the following.

6. Experiments

In the experimental section we are going to show that the new approach can compete with current methods for compensating the covariate shift. First we compare the Fourier based approach, where the distance is estimated either on the training (Tr) or the test data (Te), to other methods on some benchmark datasets, and then show results on a real world dataset. We use the total variation distance (TV), the Kullback-Leibler divergence (KL), and the squared Euclidean distance (SE).

6.1. Benchmark Datasets

For the datasets with a synthetically generated covariate shift we followed the dataset creation approach described in Sugiyama et al. (2008) for reasons of comparison. We normalized the dataset to $[0,1]^d$ and created 100 datasets of 100 training datapoints and 500 test datapoints each.

The test samples are obtained by choosing a datapoint (x_n, y_n) randomly and accepting it with a sampling factor of $\min(1, 4(x_n^{(l)}))^2$, where $x_n^{(l)}$ is the lth element of x_n. For each of the 100 datasets the dimension $l \in \{1, \ldots, d\}$ is chosen randomly but kept fixed. Every randomly chosen x_n is removed from the pool even if it was not accepted. The training dataset is sampled uniformly from the remaining data. During the learning the methods will only use the training data $(\{x_n^{tr}, y_n^{tr}\}_{n=1}^{N_{tr}})$ and the test datapoints without labels $(\{x_n^{te}\}_{n=1}^{N_{te}})$. The test labels $(y_n)_{n=1}^{N_{te}}$ are used for performance measurements.

6.2. Parameter Estimation

In our experiments we estimated a set of best parameters for a SVR and a SVM without weights (uniform) with classic cross-validation. Then we calculated the weights once with the Fourier based approach and once with the KLIEP, uLSIF and the Kernel Mean Matching (KMM) method. We then employed these weights to the weighted SVR and weighted SVM (as described in Huang et al. (2007)) with RBF kernels and estimated a new set of best parameters by using IWCV (Importance Weighted Cross-Validaton) (Sugiyama et al., 2007). IWCV works like classic cross-validation but additionally weights each fold, such that errors in regions of importance get an higher impact on the cross-validation error.

Our new Fourier based method uses two types of parameters. The parameter K, which denotes the length of the Fourier series, will be fixed to 10 here which gives a reasonable approximation. In general K should be viewed as a hyperparameter to be suitably selected, but note that in our experiments larger K did not result in significantly different performance, whereas with smaller K the results degrade as one would expect. In other words, our experiments indicate that a large enough K can be easily selected. The other parameter is σ, the kernel width in the weight function (5). We will now suggest a method for estimating a good σ parameter.

The idea is that an appropriate parameter combination will minimize the expressions (12), (14), and (15). For given σ the corresponding αs have been determined by minimizing (12), (14), and (15). We will now choose the lowest value of the objective functions obtained during the optimization for different σ parameters.

To get a more stable result, we use a method that is similar to cross-validation, but will not use any label information. Given the original datasets, X_{train} and X_{test}, we split the test dataset into five parts, $(X_{test}^j)_{j=1}^5$. Each split X_{test}^j should contain enough samples of test data since they can normally be obtained quite easily. Each of the $j = \{1, \ldots, 5\}$ folds is constructed by $X_j := X_{test} \setminus X_{test}^j$. Now for a fixed parameter σ we will minimize expressions (12), (14), and (15) for each dataset combination $\{X_{train}, X_j\}$. We will calculate the mean of these five minima and choose the parameter that corresponds to the lowest average.

Minimizing the difference of the distribution of the covariates (12), (14), and (15) are independent of the labels of the test data. Therefore, we can explicitly make use of the

Table 1: Results for regression benchmark datasets. The results are obtained by taking the average of 100 mean test errors. The values in the parentheses denote the standard deviation. All errors have been normalized by the uniform result (no weights).

Dataset Dimension	kin-8fh 8	kin-8fm 8	kin-8nh 8	kin-8nm 8	abalone 7	avg -
Uniform	1.00	1.00	1.00	1.00	1.00	1.00
Fourier:TV (Tr)	0.93 (0.062)	0.93 (0.059)	0.95 (0.043)	0.91 (0.090)	0.94 (0.046)	0.93
Fourier:TV (Te)	0.95 (0.061)	0.94 (0.055)	0.95 (0.041)	0.93 (0.078)	0.94 (0.056)	0.94
Fourier:SE (Tr)	0.94 (0.077)	0.92 (0.055)	0.95 (0.047)	0.94 (0.081)	0.92 (0.091)	0.93
Fourier:SE (Te)	0.95 (0.063)	0.93 (0.061)	0.96 (0.051)	0.96 (0.090)	0.95 (0.072)	0.95
Fourier:KL (Tr)	0.93 (0.060)	0.94 (0.050)	0.95 (0.044)	0.95 (0.095)	0.91 (0.081)	0.93
Fourier:KL (Te)	0.94 (0.078)	0.96 (0.059)	0.95 (0.047)	0.95 (0.068)	0.93 (0.085)	0.94
KLIEP	0.92 (0.069)	0.92 (0.063)	0.95 (0.038)	0.97 (0.041)	0.94 (0.071)	0.94
uLSIF	0.98 (0.071)	0.94 (0.044)	0.96 (0.051)	0.96 (0.072)	0.95 (0.067)	0.95
KMM	0.97 (0.071)	0.94 (0.074)	0.96 (0.059)	0.95 (0.056)	0.93 (0.041)	0.95

Table 2: Results for classification benchmark datasets. As in table 1 results are obtained by taking the average of 100 mean test errors. The values in the parentheses denote the standard deviation. All errors have been normalized by the uniform result (no weights).

Dataset Dimension	twonorm 20	waveform 21	ringnorm 20	image data 18	average -
Uniform	1.00	1.00	1.00	1.00	1.00
Fourier:TV (Tr)	0.92 (0.079)	0.91 (0.055)	0.96 (0.091)	0.92 (0.092)	0.92
Fourier:TV (Te)	0.97 (0.072)	0.96 (0.061)	0.96 (0.081)	0.93 (0.080)	0.95
Fourier:SE (Tr)	0.89 (0.083)	0.90 (0.045)	0.95 (0.081)	0.90 (0.082)	0.91
Fourier:SE (Te)	0.93 (0.068)	0.95 (0.052)	0.98 (0.083)	0.94 (0.069)	0.95
Fourier:KL (Tr)	0.91 (0.067)	0.89 (0.056)	0.96 (0.096)	0.93 (0.076)	0.92
Fourier:KL (Te)	0.93 (0.089)	0.93 (0.045)	0.97 (0.074)	0.93 (0.079)	0.94
KLIEP	0.93 (0.069)	0.95 (0.033)	0.96 (0.077)	0.93 (0.064)	0.94
uLSIF	0.93 (0.076)	0.91 (0.048)	0.95 (0.071)	0.92 (0.079)	0.92
KMM	0.97 (0.045)	0.98 (0.040)	0.99 (0.071)	0.97 (0.083)	0.97

locality of the test data here. We therefore get a simple method for estimating adequate parameters.

Table 3: Results for the earthquake dataset (Allen and Wald, 2009). Weighted SVR significantly improves the prediction on the test data.

Uniform	1.00	uLSIF	0.96	Fourier: TV (Tr)	0.91	Fourier: KL (Tr)	0.87	Fourier: SE (Tr)	0.93
KLIEP	0.96	KMM	0.93	Fourier: TV (Te)	0.92	Fourier: KL (Te)	0.92	Fourier: SE (Te)	0.93

6.3. Experimental Results

For the experiments we created artificial covariate shift data as described in section 6.1. We used data from the DELVE repository and the abalone dataset for regression. For classification experiments we obtained the IDA datasets available on mldata.org. For each of the datasets we created 100 subdatasets and set the test data as the center points of the weight function (5). For all datasets we calculated the mean test error and normalized it by the mean test error of the uniform SVR or SVM, respectively. Note that the computing times of the Fourier methods and KLIEP were roughly the same, whereas uLSIF was slightly faster.

The results in Tables 1 and 2 show that employing weights improves the prediction performance. Observe that the Fourier approach measuring the distance on the training data is always better than the corresponding one using the test data. A reason for the slightly poorer results on the test data might be due to the fact that for SVR and SVM we are interested in calculating weights for the training data. Therefore it seems to be preferable to use the training data for the distance estimation to achieve on these a small distance between the test and reweighted training distribution.

The best method varies over the data sets, but on average the Fourier based approaches measuring the distance on the training data are better than KLIEP, uLSIF, and KMM for both the regression and the classification data. When one compares the results using KL, one observes that the Fourier based approach when measuring the distance (13) on the test data is on average comparable to KLIEP, which also estimates the distance on the test data, whereas measuring the distance on the training data slightly improves the results.

The second experiment is performed on a real world dataset (Allen and Wald, 2009). The dataset is again a regression dataset and it is comprised of measurements recorded during earthquakes in California and Japan. The features describe values such as magnitude or distance to the center. A categorical feature describes the type of the earthquake, we augmented the dataset and assigned a separate dimension for each category, which turns one dimension into three. The label to predict is the so called PGA (Peak Ground Acceleration) value.

We learned on the California data and applied the achieved model for prediction on the Japan earthquake data. Again we used Gaussian kernels in the normal SVR with no weights (uniform) and the weighted SVR method described in section 5. As in the previous experiments we normalized the results by the normal unweighted (or uniform) result. For the Fourier approach, the chosen weight parameters have been estimated by the modified cross validation procedure described in section 6.2. It turns out that learning a weighted SVR improves the prediction result on the Japan dataset, as shown by Table 3. It seems natural to assume that due to the geographical differences, especially location of the

measurements, there occurs a natural shift in the data, but that the implications remain the same for the PGA value. Our experiments show that the application of weights to the regression method considerably improves the results, where the Fourier based approaches show even more error reduction than the uLSIF, KMM, and KLIEP methods.

7. Conclusion

In this work we introduced a new method for measuring and compensating the covariate shift. We derived a new formulation for finding appropriate importance weights by using a Fourier approximation of the divergence measure between the test distribution and the reweighted training distribution which does not make explicit use of the density functions and takes a more function centric view than other data centered approaches. Higher dimensional problems can be treated by using a hyperbolic cross approximation in Fourier space. An advantage is that it enables the calculation of less volatile and therefore better weights especially in cases of small bandwidth parameters σ. Furthermore, the new approach gives a flexible framework since it can handle different divergence measures and can use any point set for the empirical estimation of the divergence. Besides investigating further divergence measures we in particular are interested in the influence of the choice of the points where the divergence is measured, besides training and test points one here could also think about using Smolyak quadrature points (Smolyak, 1963) or Quasi-Monte-Carlo sequences (Dick et al., 2013).

Note that currently all attributes are treated equally, but the hyperbolic cross approach can be extended to have different resolutions in each dimension, which corresponds to dimension-dependent smoothness properties. In such a case a dimension-adaptive choice of the Fourier resolution in the different dimensions can be achieved in a similar fashion to that described in Gerstner and Griebel (2003). Such an approach would allow the treatment of even higher dimensional problems.

Finally, the approach for compensating covariate shift is not limited to the current choice of a linear combination of (Gaussian) kernels for the weight function. An interesting possibility would be the use of a sparse grid-based approach (Garcke, 2006; Pflüger, 2010), where the same underlying idea of a sparse tensor product construction and Sobolev spaces with dominating mixed smoothness as for the hyperbolic cross approximation exists.

References

T.I. Allen and D.J. Wald. Evaluation of ground-motion modeling techniques for use in global shakemap—a critique of instrumental ground-motion prediction equations, peak ground motion to macroseismic intensity conversions, and macroseismic intensity predictions in different tectonic settings. *U.S. Geological Survey Open-File Report 2009—1047*, 2009.

K.I. Babenko. Approximation by trigonometric polynomials in a certain class of periodic functions of several variables. *Sov. Math., Dokl.*, 1:672–675, 1960.

S. Bickel, M. Brückner, and T. Scheffer. Discriminative learning under covariate shift. *Journal of Machine Learning Research*, 10, 2009.

O. Chapelle, B. Schölkopf, and A. Zien. *Semi-Supervised Learning*. MIT Press, 2006.

J. Dick, F. Kuo, and I. Sloan. High-dimensional integration: The quasi-Monte Carlo way. *Acta Numerica*, 22:133–288, April 2013. doi: 10.1017/S0962492913000044.

J. Garcke. Regression with the optimised combination technique. In W. Cohen and A. Moore, editors, *Proceedings of the 23rd ICML '06*, pages 321–328, 2006.

T. Gerstner and M. Griebel. Dimension–Adaptive Tensor–Product Quadrature. *Computing*, 71(1):65–87, 2003.

J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, and B. Schölkopf. Correcting sample selection bias by unlabeled data. In *NIPS 19*, pages 601–608, 2007.

T. Kanamori, S. Hido, and M. Sugiyama. A least-squares approach to direct importance estimation. *The Journal of Machine Learning*, 10:1391–1445, 2009.

S. Knapek. Hyperbolic cross approximation of integral operators with smooth kernel. Technical Report 665, SFB 256, Univ. Bonn, 2000. URL http://wissrech.ins.uni-bonn.de/research/pub/knapek/fourier.ps.gz.

J. Moreno-Torres, T. Raeder, R. Alaiz-Rodríguez, N. Chawla, and F. Herrera. A unifying view on dataset shift in classification. *Pattern Recognition*, 45(1):521–530, January 2012. doi: 10.1016/j.patcog.2011.06.019.

D. Pflüger. *Spatially Adaptive Sparse Grids for High-Dimensional Problems*. Verlag Dr. Hut, 2010.

J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. Lawrence. *Dataset Shift in Machine Learning*. The MIT Press, 2009.

S. A. Smolyak. Quadrature and interpolation formulas for tensor products of certain classes of functions. *Dokl. Akad. Nauk SSSR*, 148:1042–1043, 1963. Russian, Engl.: Soviet Math. Dokl. 4:240–243, 1963.

M. Sugiyama and M. Kawanabe. *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*. MIT Press, Cambridge, Mass., 2012.

M. Sugiyama, M. Krauledat, and K.-R. Müller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8:985–1005, 2007.

M. Sugiyama, S. Nakajima, H. Kashima, P. Bünau, and M. Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *NIPS 20*, pages 1433–1440, 2008.

Y. Tsuboi, H. Kashima, S. Hido, S. Bickel, and M. Sugiyama. Direct density ratio estimation for large-scale covariate shift adaptation. *Journal of Information Processing*, 17:138–155, 2009.

Din' Zung. The approximation of classes of periodic functions of many variables. *Russian Mathematical Surveys*, 38(6):117–118, December 1983.