New Importance Sampling Based Algorithms for Compensating Dataset Shifts

vorgelegt von
Diplom-Mathematiker
Thomas Vanck
geboren in Wittlich

Von der Fakultät II - Mathematik und Naturwissenschaften
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Naturwissenschaften
Dr.rer.nat.

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender: Prof. Dr. Martin Henk
Gutachter: Prof. Dr. Jochen Garcke
Gutachter: Prof. Dr. Reinhold Schneider

Tag der wissenschaftlichen Aussprache: 23. Mai 2016

Berlin 2016

Mein Dank richtet sich in erster Linie an Herrn Prof. Dr. Jochen Garcke, der bereit war, mich als Promotionsstudent über die Jahre zu betreuen und zu unterstützen. Des Weiteren möchte ich mich bei Herrn Prof. Dr. Reinhold Schneider für die Bereitschaft bedanken, als Gutachter zu fungieren. Ein abschließendes Dankeschön geht an den Vorsitzenden Herrn Prof. Dr. Martin Henk.

Contents

This work contains texts, results and images from several of my publications. However, I cannot simply use *my* own creations here in *my* thesis. Instead, due to some ridiculous legal stuff I have to state (for one of my publications) at least the following sentence:

Alright, having that said, let's dive in!

Chapter 1

Introduction

Machine Learning is a field of research concerned with developing new (parameter dependent) algorithms for solving problems that are commonly denoted as regression and/or classification problems. The goal in both problem settings is to learn a functional relationship between some input data x and some associated output y without explicitly knowing the real dependence. This is achieved by learning a mathematical model $\hat{f}$ of the real (unknown) function f. Situations in which the output variable y is continuous are called regression problems and if y is comprised of discrete values it is referred to as classification. In fact, classification can be regarded as a discrete regression problem, and can, therefore, be considered a special case of regression. In general, the model $\hat{f}$ depends on a limited number of parameters which are inferred from a given set of data. After inference, the parameters are applied to the model to perform a prediction of new data. Commonly, it is assumed that two D-dimensional datasets are given, the first set $X^P \subset \mathbb{R}^{D \times M}, M \in \mathbb{N}$ is used for the prediction and will be referred to as primary dataset as it is of primary interest (it is also called target or testing data). The second dataset $X^S \subset \mathbb{R}^{D \times N}, N \in \mathbb{N}$ is for inferring the parameters and will be named secondary data because it takes the role of a helper dataset that is not of primary concern for the prediction. It is therefore often called auxiliary, source or training data. In addition to the data, one also gives the so called labels or dependent variables $Y^P \subset \mathbb{R}^M$ and $Y^S \subset \mathbb{R}^N$ respectively. The labels encode the value that one wants to predict. The secondary labels are used for the parameter inference of the prediction model and the primary labels for assessing the quality of the actual model prediction. The inference or learning procedure in a standard machine learning assumes that both dataset samples are equally/identically and independently distributed (i.i.d.) and, therefore, sampled according to the same distribution [1]. This means that for a given distribution $p : \mathbb{R}^{D+1} \to \mathbb{R}^+$, the

[1] In machine learning literature the terms density and distribution are used interchangeably.

data is distributed as $X^P, Y^P \sim p$ and $X^S, Y^S \sim p$. Such an assumption is often justified and furthermore provides a very consistent and convenient analytical framework for the analysis and the development of a diversified set of algorithms. Moreover, this model is an ideal simplification and intuitive in most cases. The main problem is that in certain cases, the assumption of identically distributed data can be violated and both dataset samples do not necessarily follow the same distribution. The reason for that is a change in the underlying distribution, i.e. the distribution p^P for the primary data differs from the secondary distribution p^S.

A well known example of a violation of this assumption is the stock market. The price of a stock, in general, depends on several influences, like the actual financial and operative situation of a company or shareholders speculations on the future of the company, etc.. These influences and their impacts fully characterize the underlying distribution p but are normally unknown and unobservable in most cases. Further, if these influences and/or their impacts change, the characteristics of the distribution p also change. Therefore, if one has given secondary data X^S, Y^S from a period where the data generating process followed a distribution p^S the current primal data X^P, Y^P usually follows another distribution $p^P \neq p^S$. This change in distribution is also known as non-stationary stochastic process [69] and as a result the old data X^S, Y^S is shifted towards the new data X^P, Y^P. Such kind of shift (or shifts), which are known as dataset shifts, have a huge impact on the model, since the model inferred on the S data is no longer applicable to the data of interest, i.e. the P data. One can think of different approaches for solving this problem. For instance, in the stock market example the influences are extremely difficult to model and therefore the stock price is commonly modeled as a brownian motion [55] (a.k.a Wiener process) which is actually an oversimplification. Since the brownian motion is by construction a stationary process, one often applies an additional (often deterministic) function for modeling the non-stationarity (for instance an Ornstein-Uhlenbeck process [81]). This implies that the characteristics of non-stationarity are exactly known. Similarly, in most other real world scenarios, the non-stationary cannot be modeled due to the same or comparable reasons as for the stock market example.

In contrast to the deterministic approach, statistical methods simplify such problems by providing general models that depend on a limited number of parameters. The advantages of such approaches are that they are flexible and relatively easy to apply. The disadvantage is that one has to accept a certain degree of inaccuracy due to the lack of exactness. However, in many situations statistical approaches provide satisfiable results and are a good compromise between complexity, accuracy and costs. Yet, statistical methods are fundamentally different from the deterministic setting. Instead of estimating a model of a drift function, as in the deterministic setting, a statistical method tries to figure out a function that gives information about which datapoints from a given state are applicable to another state such that a modified model only considers those points of the dataset that could as well correspond to the other state of the process. Following this idea and ignoring the estimation of a model of a transformation or a drift function, the problem gets simplified such that a more flexible framework is considered.

Therefore, statistical approaches are well suited for the dataset shift problem, since the goal is to provide a good model for the prediction of the data of primary interest and not the determination of the actual hidden transformation process. Considering only the secondary data that helps to improve the inference of a prediction model for the P data, it can be achieved by stating a function that assesses which secondary datapoints are appropriate and which ones are not. One idea for getting this function could be the determination of the difference inbetween the states of the data generating process. The resulting function can then be applied to an existing model such that the old model is transformed to account for the shift between the secondary and the new data and therefore better predicts the new shifted data with the help or in the context of the secondary data. Since the states of the process are characterized by the distributions p^P and p^S one can figure out this function by applying these distributions. That means, that the difference between the distributions p^P and p^S needs to be somehow measured. Special types of functionals, called divergence measure, provide an analytical framework for measuring the difference between two given distributions. The result of such a divergence measure is a positive number that can be understood as a similarity or dissimilarity score between these distributions. The score then provides a guideline for inferring the drift compensating function which is then applied to the existing model. Ideally, the resulting modified model is less prone to the drift which, as a consequence, provides a better prediction.

While this approach is reasonable the problem, in general, the two distributions p^P and p^S are unknown. If both were known, one would already have an exact mathematical model of the underlying data generating process. Instead, normally, one only gives the dataset samples X^P, Y^P and X^S, Y^S that correspond to each state or distribution of the process. However, since the data X^P, Y^P and X^S, Y^S reflect the characteristics of the primal and secondary distribution p^P and p^S, respectively, it is possible to conduct properties of each distribution by applying the corresponding dataset. Thus, instead of applying a divergence measure directly, an empirical approximation based on the given data is used for the estimation of the similarity score between the two distributions. The so obtained similarity scores can then be used to determine a function w whose values can be interpreted as some sort of weights. That means for a given new datapoint (x^*, y^*) from the domain of definition, this weight function $w(x^*, y^*)$ returns a number that states the similarity of the datapoint (x^*, y^*) to the dataset of interest, i.e. the

P data X^P, Y^P. A high value of a weight $w(x^*, y^*)$ denotes a high similarity to the P data while a low value a dissimilarity. By applying this function w to the problem at hand, it becomes possible to figure out those points of the secondary data X^S, Y^S that are similar to the P data. Therefore, one can extend the number of data available for inferring a new model on the P data by adding similar S datapoints to the inference procedure. This is especially of advantage in situations where only very few P data is available such that one can improve the inference of parameters for the prediction model of the P data. Approaches that apply this strategy are commonly referred to as instance based approaches since they treat each datapoint or instance of the S data individually. Instance based approaches are very versatile and can be applied in a lot of dataset shift situations. This thesis presents two new algorithms for compensating two different types of dataset shifts.

The first algorithm improves an existing set of instance based algorithms by applying a Fourier series approximation. Although instance based methods are very intuitive approaches, they still have some drawbacks that become evident when only a very low number of data points at some process states is sampled. Given a very low number of samples, the risk of an inappropriate approximation of the divergence measure grows. The weight function for compensating the shift might become too volatile such that it no longer provides a proper solution. This volatility is due to the noise in the given data which has an higher impact since only a limited number of datapoints is given. In order to compensate this obstacle, one can apply the Fourier series to the divergence measure which yields a new kind of approximation. It turns out that the volatility of the weight function is mainly encoded in the higher frequencies spectrum. Therefore, by truncating the length of the Fourier series, one can eliminate these higher frequencies or noise and only capture those frequencies that have a systematic impact. This kind of technique can be understood as some sort of filtering that leads to a smoother weight function also in situations where only very few dataset samples are given. While providing better approximation properties for some dataset shift settings, the method unfortunately is prone to the curse of dimensionality. The reason for that is the Fourier series itself. When considering D-dimensional problems, the Fourier series approximation of length K requires the calculation of $(2K+1)^D$ coefficients which grows exponentially. Even at a very low number of dimensions like for instance $D = 10$ this approach can be computationally too demanding such that calculations might become too expensive to perform. To compensate this problem a structure called the Hyperbolic cross can be applied which is a special selection scheme for the Fourier coefficients that reduces the amount of coefficients to an acceptable number. By applying this selection scheme, ideally those coefficients are applied that contribute significantly to the actual approximation while the other low contributing ones are omitted. As a result, the final approximation is very accurate and simultaneously computationally feasible. This way, one can also benefit from the smoothing property in higher dimensional spaces.

The second new contribution of this thesis are two new algorithms for compensating a special type of dataset shifts, called Source Component Shift. This is a situation where not only the samples of the features or covariates X are shifted but also the corresponding labels or dependent variables Y. The new algorithms can be applied to regression problems and produces good results when the number of P data samples is very low. Both algorithms also belong to the class of instance based approaches and assign large weight factors to S data that is similar to the P data. Therefore, these algorithms augment the P data by adding only those S datapoints that are similar to the P data in the sense of a divergence measure. Both algorithms are very general such that they do not require a special structure of the data.

This thesis is structured in the following way: Chapter 2 gives an introduction of the topic of importance sampling which plays a central role for stating the weight function. This provides a basis for the introduction of divergence measures in the next section of the chapter. Chapter 3 gives a brief introduction of Fourier series approximation and basically is a chapter that briefly summarizes the properties and benefits. In addition, the Hyperbolic cross is introduced and its benefits are highlighted. Chapter 4 explains the types of dataset shifts considered in the machine learning setting and how it can be compared to the standard machine learning setting. Each type shift will be defined in mathematical terms and illustrated by figurative examples. Given the definitions of different types of dataset shifts, chapter 5 is about types of machine learning approaches for compensating those shifts. Different types of such approaches for compensating shifts are presented which are commonly referred to as Transfer Learning approaches since they transfer knowledge from one dataset to another. Since instance based transfer learning approaches are paramount for this thesis, this type of transfer learning technique is discussed in details in the subsequent chapter 6. The beneficial properties of divergence measures that have been explained in chapter 2 are investigated in more detail in reference to the derivation of an appropriate approximation of the weight function. Further, the characteristics of the weight function approximation are explained and illustrated such that the reader will get a feeling which critical properties should be paid attention to when considering this type of approximation. Based on these considerations, the current state of scientific work of instance based approaches is introduced such that it becomes clear how the new methods fit into the current state of the art methods. Finally, modifications of standard learning algorithms are derived such that they can employ weight coefficients and therefore become able to compensate dataset shifts. Chapter 7 introduces a new approach for compensating a dataset shift called covariate shift. The new approach is derived by applying the Fourier series approximation to different divergence measures. In order to make it applicable to higher dimensional problems the Hyperbolic cross is implemented. Several properties of the new approaches are investigated and an extensive experimental section shows the benefits compared to existing methods. The last chapter 8 introduces two new types of algorithms for compensating source component shifts. The chapter provides detailed derivations of both algorithms and an extensive analytical and experimental analysis.

Chapter 2

Importance Sampling and Divergence Measures

The results presented in this thesis rely on several mathematical concepts which are explained in the next few chapters. The first concept is called *Importance Sampling.* Importance Sampling is a technique that states the similarity of two given distributions p and q as a function of x (called importance function) such that one can get a similarity value in $\mathbb{R}^+$ at each point. The second part of this chapter will discuss *divergence measures.* Divergence measures are functions that can be understood as a class of functions for measuring the similarity of two given probability functions. They can be used in combination with importance sampling such that they extend the importance function to a measure of the similarity of two given distributions p and q on the whole domain of definition. Therefore, the main goal of this chapter is the introduction of a set of mathematical tools for measuring similarities for a given set of distributions.

2.1 Importance Sampling

The first section will start with the technique called *Importance Sampling.* Importance sampling is basically a transformation mechanism that can be useful in situations in which data samples generated by a distribution p are preferably investigated under another distribution q of interest. Thus, importance sampling can be an ideal choice when working with non-deterministic quantities. Typical applications are variance reduction and/or the calculation of Monte Carlo estimators.

2.1.1 Introduction to Importance Sampling

A common approach in statistics is the calculation of Monte Carlo estimators. Monte Carlo denotes the approximation of an expectation by the sample mean of a function of simulated random variables. Mathematically speaking, given a set $\mathcal{X} \subset \mathbb{R}^D$, a function $f : \mathcal{X} \mapsto \mathbb{R}$, and a density $p(x)$ one would like to calculate:

$$\mathbb{E}_p[f] = \int_{\mathcal{X}} f(x)p(x)dx \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i) =: \bar{f}$$

where $\{x_i\}_{i=1}^N$ is data drawn according to p. The approximation of this integral $\bar{f}$ is called the mean or Monte Carlo estimate. Since the result of the Monte Carlo estimate $\bar{f}$ depends on the samples $\{x_i\}_{i=1}^N$ and its corresponding size it is in itself a random variable. Therefore, the expression $\bar{f}$ has also a variance which is given by:

$$\mathrm{Var}\left[\bar{f}\,\right] = \mathrm{Var}\left[\frac{1}{N}\sum_{i=1}^{N} f(x_i)\right] \approx \frac{N\mathrm{Var}\,[f]}{N^2} = \frac{1}{N}\int_{\mathcal{X}} (f(x) - \mathbb{E}\,[f])^2\, p(x)dx. \qquad (2.1)$$

This variance can be interpreted as the risk of the mean estimator. A large variance implies a high uncertainty about the estimation of the real expectation. Therefore it would be desirable if one could reduce this risk. Importance sampling is a technique for reducing the variance of such estimators by sampling data points from another distribution that accounts for events that have a higher impact on the calculation of the Monte Carlo estimate and are therefore considered important events. The variance is especially high if these important events only happen infrequently. Since then, these events would be rare and have a high impact on the outcome. A more frequent consideration of these rare events could reduce the variance and hence improve the reliability of the calculation of the expectation. A classical example for the application of importance sampling in combination with such rare events can be found in the finance sector. There, the price calculation (a.k.a. pricing) is a common problem where the price of a financial derivative, for instance a knock out or barrier option, has to be estimated. A barrier option only generates a payout if a certain goal, like price barrier, is reached - otherwise the payout is zero. Having several trials of samples of the underlying (for instance stock price samples simulated by a brownian motion) the price is then given by calculating the Monte Carlo estimate of $\sum{}^{\text{payouts}}/_{\text{trials}}$, where the payout is either positive or zero and the trials are the number of several simulated samples. The price is therefore the expected (fair) gain one can achieve with the derivative. Commonly, the payout is very often a rare event because the touching of such an event, i.e. the price barrier, might happen only very scarcely. This can lead to a situation with several simulated trials of samples which, due to the low probability of a rare event, do contain zero or only very few samples that reflect such a payout event. Typically, this happens if the current price of the underlying (the stock) of the derivative is distant to a payout event such that the price or value of the option will be low due to the low probability of a payout. However, these rare events are very critical for the correct pricing of the option. If, for instance, only 100 trials are given, there is a chance that all sampled payouts are 0 which results in a price of 0 for the option. This cannot be, since the chance of making a profit with this option is not 0. Therefore, when having only very few data, the risk or variance of the estimate is in this case too high. On the other hand, the acquisition of more data solves the problem because the estimation becomes better. Unfortunately, acquiring more data usually goes along with higher computational costs which often is neither possible nor acceptable. Therefore, instead, another method for the mean estimation is needed that is reliable and simultaneously employs only very few data. Importance sampling accounts for this by calculating the mean under a different distribution q that more often generates these rare events:

$$\begin{aligned}\mathbb{E}_p\left[f\right] &= \int_{\mathcal{X}} f(x)p(x)dx \\ &= \int_{\mathcal{X}} f(x)\frac{q(x)}{q(x)}p(x)dx \\ &= \int_{\mathcal{X}} f(x)\frac{p(x)}{q(x)}q(x)dx = \mathbb{E}_q\left[\frac{p}{q}f\right].\end{aligned}$$

Interpreting the latter expression means that the data is now generated by another distribution q that more frequently generates the rare events. However, if the data would only be sampled from q one would get a biased estimate. To prevent this, a correction or reweighting is performed by applying $\frac{p(x)}{q(x)}$ to each realization of $f(x)$. By construction, these reweighting factors exactly compensate for this bias which makes importance sampling an unbiased method. Therefore, if one would know the exact distribution of the payouts one could better estimate the option price by applying importance sampling. In fact, in practical applications the outcome of the empirical estimate essentially only depends on such important or rare events. Therefore, due to the higher sampling rate of those events the relevant information is observed much more often which in return increases the confidence of the estimation. Proofs that the importance sampling approach reduces the variance of an estimator make use of the Cauchy-Schwarz inequality and are given for instance in [82]. As shown, by choosing a good sampling distribution q it becomes possible to reduce the risk of the estimator. Unfortunately, an unwise choice of q on the other hand can also increase the risk which should be taken into account when selecting another sampling distribution q. Consequently, densities for importance sampling should be constructed in a way such that they preferably pick important samples.

2.1.2 Importance Sampling Example

The following illustrative example gives insights on the benefits of importance sampling methods by showing how the application improves the accuracy of an estimator. The following event or indicator function (see also figure 2.1) is given:

$$\mathbb{1}_A(x) = \begin{cases} 1 \text{ if } x \in A = [7, 8] \\ 0 \text{ otherwise.} \end{cases} \tag{2.2}$$

The distribution of the variable x is considered to be normally distributed by $p \sim \mathcal{N}(\mu, \eta^2)$ with $\mu = 5$ and $\eta = \sqrt{2}$ (blue function plot in 2.1). Then the exact analytical mean of the event function is given by:

$$\mathbb{E}_p[\mathbb{1}_A(x)] = \int_{-\infty}^{\infty} \mathbb{1}_A(x) dp(x) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\eta^2}} \mathbb{1}_A(x) e^{-\frac{(x-\mu)^2}{2\eta^2}} dx. \tag{2.3}$$

Pretending that the exact analytical mean value can not be calculated due to an unknown distribution p it is assumed to have a finite set of samples $X = \{x_1, \ldots, x_N\}$ that have been drawn from the original or canonical sampling distribution p. Then, the empirical mean is given by its Monte Carlo estimate:

$$\bar{X} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}_A(x_i). \tag{2.4}$$

By the strong law of large numbers, this sample average converges to the true mean as N tends to infinity. But since in practise only a finite sample of observations X from the distribution p is given, one is facing a situation which implies a certain variance on the sample mean which is given by (2.1). In unfortunate situations, the variance of the estimator can be very high due to the fact that important events occur very rarely. The empirical variance of expression (2.4) is calculated by considering various empirical means $\{\bar{X}^1, \ldots \bar{X}^M\}$ that have themselves been calculated by considering multiple sample sets $X^j = \{x_1^j, \ldots, x_N^j\}_{j=1}^M$ from the distribution p. The upper right plot in figure 2.1 shows the empirical distribution of various sample means as a histogram. The broader the distribution, the larger the variance and the more uncertainty about the reliability of the approximation of the real mean is present.

As previously explained, an appropriate application of importance sampling will reduce this uncertainty. However, in order to benefit from this technique the choice of sampling function q is crucial. Considering the event function (2.2) one faces the problem that the event 1 of the indicator function is relatively rare to happen when sampling from the canonical distribution p. Yet, this event has a huge impact on the final result and is therefore considered important. Therefore, frequent observations of this event

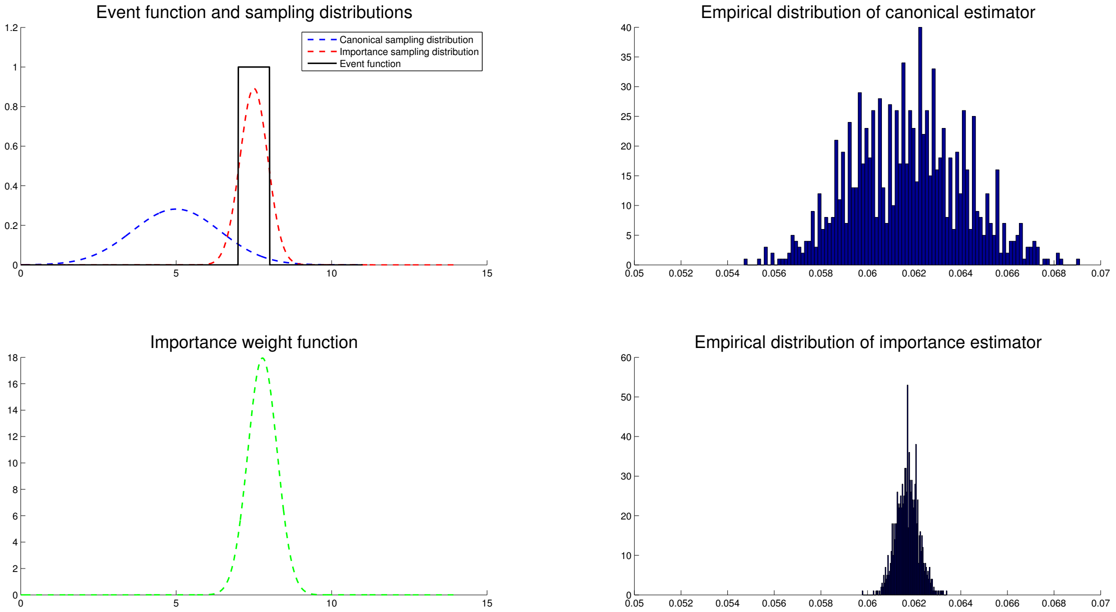


FIGURE 2.1: Illustrative example of the benefits of importance sampling. Upper left plot: The blue normal distribution denotes the canonical sampling distribution. Upper right plot: The histogram of various calculated sample means according to (2.4). Estimator (2.4) has relatively large variance. Lower right plot: The histogram of the reweighted empirical mean (2.4) according to the importance sampling weights given in the lower left plot (green dashed line). The variance is significantly reduced which enables a much more accurate estimation of the empirical mean.

would be desirable because it would enable the calculation of the empirical mean more precisely. Following the idea of importance sampling one should choose a new sampling distribution q which more often generates the event 1. Thus, to get more rare events, an appropriate sampling distribution could be $q \sim \mathcal{N}(\mu_{IS}, \sigma_{IS}^2)$ with mean $\mu_{IS} = 7.5$ and variance $\sigma_{IS} = \sqrt{.2}$ (see red function plot in figure 2.1). A sample X^{IS} according of this distribution would much more often produce samples that correspond to the rare events of the event function 2.2. However, in order to prevent a biased estimator for the sample mean w.r.t. to the canonical distribution p one additionally has to calculate the importance weights for the reweighting of each sample:

$$\mathbb{E}_p\left[\mathbb{1}_A\right] = \int_{-\infty}^{\infty} \mathbb{1}_A(x)p(x)dx = \int_{-\infty}^{\infty} \mathbb{1}_A(x)\frac{p(x)}{q(x)}q(x)dx = \mathbb{E}_q\left[\frac{p}{q}\mathbb{1}_A\right].$$

Due to the considerations in section 2.1.1 this is an unbiased estimator of the original sample mean under p and therefore converges to the exact mean when N tends to infinity. The main advantage of this approach is that as a result one gets a much lower variance in the estimation of the mean which in turn means that either less samples are needed or a more confident estimation of the mean becomes possible by applying the same number of samples to the calculation. The other plots at the right in figure 2.1 show the distribution/histogram of 1000 sample means once taken by distribution p (upper right) and once by sampling according to q and sample reweighting by the importance function (lower right). As can be seen from the figure the application of importance sampling reduces the variance significantly resulting in a much narrower and peaked distribution.

2.1.3 The Importance Function

Of special interest for this work are the importance weights $\frac{p(x)}{q(x)}$ for unbiasing the sample mean. The quotient assigns an individual weight to each value x from the domain and can therefore be considered being a weight function, defined by $w(x) := \frac{p(x)}{q(x)}$. This weight function $w : \mathcal{X} \subset \mathbb{R}^D \mapsto [0, \infty)$ is also known as the *Radon-Nikodym derivative* [62] . The *Radon-Nikodym theorem* states that for two given measures (in this particular case the densities) p and q with p being absolute continuous[2] w.r.t. q $(p \ll q)$ on a measure space Ω and its corresponding σ-algebra [46] $\mathcal{F}$ there exists a function w (the so called Radon Nikodym derivative) such that for a set $A \in \mathcal{F}$ the equality $p(A) = \int_A wdq$ holds. The derivative is often denoted by $w = {}^{dp}/_{dq}$. In the importance sampling setting the Radon Nikodym derivative $w(x)$ is usually called *importance function* and can be interpreted as giving a similarity score of two given probability measures p and q at each point x. For instance, having two identical distributions $p \equiv q$ the score will be constant $w(x) \equiv 1$. Therefore, a value of 1 denotes that at a point x the distributions are equal while any other value different from 1 indicates a difference in both distributions. The differences can either be in $[0, 1)$ or $(1, \infty)$. The first case where $\{x \in \mathcal{X} | w(x) < 1\}$ means that w.r.t. the distribution p the other distribution q samples such an x too frequently. As a result, in order to conform the distribution p, this particular x needs to be corrected by a "reduction factor" smaller than 1. The second case $\{x \in \mathcal{X} | w(x) > 1\}$ is exactly the opposite, meaning that in this case the distribution q samples such a x too infrequently resulting in a lift of importance for matching the distribution p. Further, the similarity score induced by the importance function (or Radon-Nikodym derivative) w is exact. That means that the similarity score matches the exact correction ratio for the probability of x under q w.r.t. p. This result can be very useful when dealing with data samples that have been drawn from these two distributions p and q. Given the two distributions p and q one can ask how likely a sample x from distribution q is under p. This can become especially useful when having a dataset sample $\{x_1, \ldots, x_N\}$ from q for which one would like to know how likely the observed data is sampled from the other distribution p. As a result it becomes possible to determine which samples from the q dataset might be very similar to another dataset sample from p and which are not. Therefore, the potential of the application of importance sampling and in particular of the importance function w goes beyond the mere (although important) improvement of the calculation of the sample mean. The idea and application of the weight function

[2] A measure p is called absolutely continuous w.r.t. another measure q if for any element $A \in \sigma(\Omega)$ (the σ-algebra [46]) $p(A) = 0 \Rightarrow q(A) = 0$. The notation for this is $p \ll q$.

for measuring the similarity between some dataset samples is paramount for this thesis and will be the basis for the new findings. The next chapters will introduce concepts that depend on this idea in detail and show the benefits when applying these concepts to some new kind of problems.

2.2 Divergence Measures

This section is about a class of mathematical functions that are commonly referred to as divergence measures. The purpose of these functions is to measure the similarity or dissimilarity of probability distributions. This is achieved by wrapping the importance function introduced in the previous section and returning a value that can be interpreted as a similarity score. The following subsection mainly follows the presentations given in [23].

2.2.1 Divergence Measure

Divergence measures are widely used in a lot of areas of statistics, machine learning, inference, optimization and others [23]. The definition requires two probability density functions p, q from the space of functions $\mathcal{P} := \{f | f : \mathcal{X} \subset \mathbb{R}^D \rightarrow \mathbb{R}^+, \int_{\mathcal{X}} f(x)dx = 1\}$. Then the divergence measure is defined by $D(\cdot||\cdot) : \mathcal{P} \times \mathcal{P} \rightarrow \mathbb{R}^+$ such that $D(.||.) : (p, q) \mapsto \mathbb{R}^+$ assigns two given probability density functions p and q a positive number. In the case of a discrete probability distribution $p = (p_1, \ldots, p_n) \in [0, 1]^N$, where the probabilities are given as a N-dimensional vector, the definition is given as $D(\cdot||\cdot) : [0, 1]^N \times [0, 1]^N \rightarrow \mathbb{R}^+$. A divergence measure is often applied as a distance measure between two distributions p, q although it does not satisfy all the requirements of a metric. Since it does not necessarily meet the symmetry condition, i.e. $D(p||q) = D(q||p)$ does in general not hold, it can not be considered a metric. Further, it also does not need to satisfy the triangular equality, i.e. $D(q||p) \nleq D(q||r) + D(r||p)$ for some other distribution r. However, they are reflexive, i.e. $D(p||q) = 0$ if and only if $p = q$, and positive $D(p||q) \geq 0$. Therefore a divergence measure provides a quasi-distance or directed difference between two probability distributions or densities p and q.

It should be noted that in most machine learning literature the terms distance and divergence are used interchangeably although this is strictly formally spoken not correct. For this work, two main classes of divergences are of importance. The first one is the class of Csiszár f-divergences and the second one is the class of Bregman divergences. Csiszár f-divergences [25] describe a class of (permutation) invariant and non-decreasing local projections [23]. The class of Csiszár f-divergences covers properties of popular divergences like Kullback-Leibler divergence [52] or Hellinger distance [42]. The second class are the so called Bregman divergences [15]. This class emerges from strictly convex functions and gives a generalization of the squared euclidian distance. Both classes are briefly discussed in the following two sections.

2.2.1.1 Csiszár f divergences

The *Csiszár f-divergences* describe a class of functions or functionals that can be used to quasi-measure the difference between two given probability functions p and q. They have been independently defined by [25], [58] and [4]. For two given probability densities p and q with support $\mathcal{X} \subset \mathbb{R}^D$ and p absolutely continuous w.r.t q (i.e. $p \ll q$) the function is defined by:

$$D_f\left(p||q\right) = \int_{\mathcal{X}} f\left(\frac{p(x)}{q(x)}\right) q(x) dx. \tag{2.5}$$

Here, $f : [0, \infty) \to \mathbb{R}^+$ is a convex function with $f(1) = 0$ and p, q are absolutely continuous with respect to the Lebesgue integral. The quotient $\frac{dp}{dq}$ denotes the Radon–Nikodym derivative [62]. In a discrete setting the divergence becomes:

$$D_f\left(p||q\right) = \sum_{i=1}^{N} d\left(p_i||q_i\right) = \sum_{i=1}^{N} f\left(\frac{p_i}{q_i}\right) q_i \tag{2.6}$$

where the distributions are given as $p = \{p_1, \dots, p_N\}$ and $q = \{q_1, \dots, q_N\}$. The quotient (which is the discrete counter part of the Radon-Nikodym derivative) is also referred to as *odds ratio* or *likelihood ratio*.

f-divergences have several properties which are briefly stated here. The first property is the separability property which means that in the case of discrete distributions $D(p||q)$ can be written as a sum $D(p||q) = \sum_{i=1}^{N} D(p_i, q_i)$. This holds true for any f-divergence. The f-divergence is invariant which means that in the case of permutations of discrete distributions p_i and q_i the divergence measure does not change. In the continuous setting one has to apply a diffeomorphism $h : \mathcal{X} \mapsto \mathcal{X}'$. Given $\mathcal{J}_h(x)$ the determinant of the Jacobian of h one gets for a density $p(y) = p(h(x)) = p(x)|\mathcal{J}_h(x)|^{-1}$. Therefore this transformation yields:

$$\begin{aligned} D_f\left(p||q\right) &= \int_{\mathcal{X}'} q(y) f\left(\frac{p(y)}{q(y)}\right) dy = \int_{\mathcal{X}} q(x)\left|\mathcal{J}_h(x)\right|^{-1} f\left(\frac{p(x)\left|\mathcal{J}_h(x)\right|^{-1}}{q(x)\left|\mathcal{J}_h(x)\right|^{-1}}\right) \left|\mathcal{J}_h(x)\right| dx \\ &= \int_{\mathcal{X}} q(x) f\left(\frac{p(x)}{q(x)}\right) dx = D_f(p||q). \end{aligned}$$

f-divergences are also non-negative, since due to Jensen's inequality and the fact that f is convex function it holds that:

$$D_f(p||q) = \int f\left(\frac{p(x)}{q(x)}\right) q(x)dx \geq f\left(\int \frac{p(x)}{q(x)} q(x)dx\right) = f(1) = 0.$$

Further f-divergences satisfy the information monotonicity [22] criteria. Figuratively spoken that means that combinations of events on which the divergence is calculated is upper bounded by the calculation on the most basic sets. Mathematically, this can be described by assuming several events $A = \{A_1, \ldots, A_n\} \subset \mathcal{X}$ and $B = \{B_1, \ldots, B_n\} \subset \mathcal{X}$ with each event A_i and B_i being an atomic element [46] having a probability p_i and q_i each and $\sum_{i=1}^n q_i = \sum_{i=1}^n p_i = 1$. By considering the σ-algebras [46] $\sigma(A)$ and $\sigma(B)$ with cardinality $|\tilde{A}| = |\tilde{B}|$ each new element $\tilde{A} \in \sigma(A)$ and $\tilde{B} \in \sigma(B)$ is a combination (like union, intersection etc.) of those basic or atomic elements. From the properties of a σ-algebra each new set arising from $\sigma(A)$ and $\sigma(B)$ has corresponding distributions $\tilde{p}$ and $\tilde{q}$. The information monotonicity states that the combination of atomic events to one event implies that the divergence measure does not increase. That is: $D(p||q) \geq D(\tilde{p}||\tilde{q})$ for each distribution p and q defined on the atomic set and any other distributions $\tilde{p}, \tilde{q}$ defined on elements from the σ-algebras.

Depending on the convex function f one gets different f-divergence measures. An extensive list of different measures is given in [23]. Important examples are:

- *Total Variation* distance where $f(u) = |u - 1|$. Plugging this f into the definition of a f-divergence one gets (the non-differentiable):

$$D(p||q) = \sum_{i=1}^n |p_i - q_i|.$$

- Squared Hellinger distance [42] which is given by $f(u) = (\sqrt{u} - 1)^2$ for which the divergence becomes:

$$D(p||q) = \sum_{i=1}^n (\sqrt{p_i} - \sqrt{q_i})^2.$$

- Pearson and Neyman Chi-square divergence defined by $f(u) = \frac{1}{2}(u-1)^2$ and $f(u) = \frac{1}{2}{(u-1)^2}/{u}$ yield:

$$D(p||q) = \frac{1}{2}\sum_{i=1}^n \frac{(p_i - q_i)^2}{p_i} \quad \text{and} \quad D(p||q) = \frac{1}{2}\sum_{i=1}^n \frac{(p_i - q_i)^2}{q_i}.$$

- The Kullback-Leibler divergence [52] or KL-divergence given by $f(u) = u-1-\log u$ yields:

$$D(p||q) = \sum_{i=1}^{n} q_i \log\left(\frac{p_i}{q_i}\right).$$

The KL-divergence and the Hellinger distance are special cases of the so-called α divergence [23].

2.2.1.2 Bregman Divergences

A second important class are the so-called Bregman divergences. The name and definition of this class of divergences goes back to [15]. Bregman divergences appear naturally in many fields of applications and the justification of its application depends on the problem at hand. For instance, Bregman divergences are used in the field of regret (or opportunity loss) [3] minimization [79]. [93] state a new algorithm for maximum margin estimation of structured output models like Markov Random Fields [13] which are undirected graphical models having a Markov property. They apply Bregman's method [37] which is an iterative algorithm for solving convex optimization problems by applying divergence measures. [12] develop an abstraction framework for the k means clustering algorithms. Instead of only applying the euclidian distance they consider different divergences of the class of Bregman divergences which leads to a generalization to a large class of clustering loss functions. Another application is the definition of conjugate priors with the help of Bregman divergences [2]. [2] give new insights on why it is reasonable to apply conjugate priors in Bayesian approaches by applying Bregman divergences. This way they can show that the geometric properties of different Bregman divergences give a better intuition on conjugate priors and additionally state a method for deriving better hyperparameters. A further notable application of Bregman divergences is given by [24]. They show that boosting in the view of a Bregman divergence leads to a new convergence proof of the adaboost algorithm [31] and additionally allows to make a connection to Logistic Regression.

Bregman divergences can be understood as a generalization of the squared euclidean distance between two points to a class of distances that share certain similarities. To understand this, consider the following decomposition for the distance between two points:

$$||p-q||^2 = \langle p-q, p-q\rangle = ||p||^2 - ||q||^2 - \langle 2q, p-q\rangle \tag{2.7}$$

[3] The regret is the difference between the difference of the outcome $\hat{f}(x)$ and the actual real label y, namely $||\hat{f}(x) - y||$ and the difference between any other outcome $\neg\hat{f}(x)$ (read like *not* $\hat{f}(x)$) and y.

The geometrical interpretation of this expression is that the second part of the expression, namely $||q||^2 - \langle 2q, p-q \rangle$ denotes a tangent at q evalutated at the point p. Therefore (2.7) is the difference between the convex function $f(q) := ||p||^2$ and the corresponding tangent for q evaluated at p. That is:

$$D(p,q) = f(p) - f(q) + \langle \nabla f(q), p-q \rangle. \tag{2.8}$$

The function $D(\cdot, \cdot)$ is always positive which geometrically implies that any tangent of f lies always below the function f. Therefore this expression (2.7) can also be considered a measure of convexity. However, the function f does not necessarily have to be the squared euclidian distance but can be any convex function.

This directly leads to the definition of the class of Bregman divergences: Given a continuously-differentiable, real-valued and strictly convex function $f : \mathcal{X} \to \mathbb{R}$ where $\mathcal{X} \subset \mathbb{R}^D$ is convex and closed. Then, for two given points $p, q \in \mathcal{X}$ the Bregman divergence is defined by:

$$D_f(p||q) := f(p) - f(q) - (p-q)^t \nabla f(q)$$

where $\nabla f(q)$ denotes the gradient of f at q. Thus, Bregman divergences are parameterized by the convex function f.

Since Bregman divergences can be understood as some kind of distance measure between a convex function f and it's tangent a lot of results of convex analysis can be applied to this mathematical structure. Like the Csiszár f-divergences Bregman divergences are in general not symmetric and also do not satisfy the requirements of the triangular inequality. In contrast to the Csiszár f-divergences the Bregman divergences do not conform to the information monotonicity criteria. Other properties of this class of functions are the non-negativity which is a consequence of the convexity of f. Further, due to the strict convexity of the function f the Bregman divergence can be expressed in dual terms. One can apply the *Legendre* transformation to obtain a dual representation. By defining $q^* := \nabla f(q)$ one gets a bijective mapping between the slope of the function f and it's function value. And the *Legendre* transformation is given by:

$$f^*(q^*) = \max_q \left\{ q q^* - f(q) \right\}. \tag{2.9}$$

Geometrically speaking, the function inside the $\max\{\cdot\}$ expression denotes a tangent of f which is always smaller or equal to the original function f due to the convexity of f. From this fact one can conclude that for a certain value q_0 and for $q^*(q_0) = \frac{d}{dq} f(q_0)$ the following holds:

$$f^*(q^*) = \max_q \{qq^* - f(q)\} = q_0 q^* - f(q_0).$$

Therefore, the *Legendre* transform satisfies $f(q) + f^*(q^*) - qq^* = 0$. Taking these considerations, the dual expression of an arbitrary Bregman divergence is given by:

$$D(p, q) = f(q) + f^*(p^*) - qp^*.$$

A special case of a Bregman divergence is the Kullback-Leibler divergence (KL-divergence). The KL-divergence is the only divergence that satisfies the requirements for both the f-divergence and the Bregman divergence. The KL-divergence in the Bregman setting is obtained by considering $f(p) = p\log(p)$. Other examples for frequently used Bregman divergences are:

- *Squared Euclidian* distance by setting $f = ||\cdot||^2$
- *Itakura-Saito* distance by $f := -\log(p)$ yields:

  $$D_{\mathrm{IS}}(p, q) = \sum_{i=1}^{N} \left(\log\left(\frac{q_i}{p_i}\right) + \frac{p_i}{q_i} - 1 \right)$$

  This special divergence is often used in the spectral analysis of speech signals.

- *Inverse* divergence, when $f := {}^1\!/\!_p$:

  $$D(p, q) = \sum_{i=1}^{N} \left(\frac{p_i}{q_i^2} + \frac{1}{p_i} - \frac{2}{q_i} \right)$$

- *Exponential* divergence for $f := e^p$:

  $$D(p, q) = \sum_{i=1}^{N} \left(e^{p_i} - (p_i - q_i + 1)\, e^{q_i} \right)$$

Further examples for Bregman divergences can be found in [23].

2.2.2 Summary

Divergence measures are a broad toolkit for measuring the similarity of two given distributions. Most of the divergences rely at their core on the Radon-Nikodym derivative and can therefore be considered to be closely connected to importance sampling. Thus, the concept of importance sampling fits canonically and naturally into this whole concept. Further, divergence measures may serve as a framework for applying the importance function $w(x) = \frac{p(x)}{q(x)}$ in the process of estimating or determining similarity. Therefore, the combination of both importance sampling and divergence measures provide a foundation for stating new kind of algorithms which enable to solve new kind of problems. The settings for such kind or problems will be introduced in the coming chapters.

Chapter 3

Fourier Series Approximation and Hyperbolic Cross

The following chapter will introduce the *Fourier Transform* and an approximation scheme for reducing the computational costs of the numerical calculations. For completeness, the first part will introduce the well known Fourier transform and the Fourier series and its basic properties. Of special interest will be the situation of a truncated Fourier series and its smoothing property for approximated functions which is due to omitting higher frequencies. Subsequently, the Fourier Transform will be considered for higher dimensional settings. In order to handle the arising curse of dimensionality an approximation scheme called *Hyperbolic Cross* will be introduced which reduces the computational costs significanty. Therefore, the main purpose of this chapter is the investigation of the smoothing property of a truncated Fourier series and the introduction of a special approximation scheme for accelerating the computations.

3.1 Fourier Transform

The first section of this chapter will give an introduction to the Fourier transform which is a field of study in the area of Fourier analysis. It generally investigates the representation of a function by trigonometric functions. Fourier analysis can be considered a branch of harmonic analysis, a field in mathematics that investigates the representation of functions by linear combinations of wave like functions. Within this field the Fourier transform itself can be again categorized into several types. The most general one is the continuous Fourier transform that is given by:

$$\mathcal{F}(f)(\xi) := \int_{-\infty}^{\infty} f(t) e^{-i2\pi\xi t} dt$$

for a Lebesgue integrable function $f : \mathbb{R} \to \mathbb{C}$. As the name suggests, both t and ξ are required to be continuous. In many applications these variables t and ξ are commonly referred to as time t and frequency ξ, respectively. Since the transformed function, $\mathcal{F}(f)(\xi)$, only depends on the frequency ξ it is also often called the frequency distribution. An interesting property of the Fourier transform is that under certain circumstances the Fourier transform can be reverted: If the function f and it's corresponding Fourier transform are both absolutely integrable (w.r.t. Lebesgue) then the Fourier transform can be fully reversed by:

$$f(t) = \int_{-\infty}^{\infty} \mathcal{F}(f) e^{i2\pi\xi t} d\xi.$$

Here, f is again the original function. Whenever $f \in \mathcal{L}^2(\mathbb{R})$ the Fourier transform is an isomorphism such that every function f has a characteristic frequency distribution that uniquely identifies the function. Therefore, having two frequency spectrums that are equal implies that both underlying functions f_1 and f_2 are the same.

Under certain conditions the Fourier transform may be represented by an infinite sum of complex exponentials. Then the Fourier transform becomes:

$$f(x) = \sum_{k=-\infty}^{\infty} c_k e^{i2\pi x \frac{k}{T}} \tag{3.1}$$

and the coefficients c_k are given by:

$$c_k = \frac{1}{T} \int_{c}^{c+T} f(x) e^{-i2\pi x \frac{k}{T}} dx. \tag{3.2}$$

Here, $c \in \mathbb{R}$ and $T \in \mathbb{R}^+$. This representation of a function is referred to as Fourier series. It should be noted that expression (3.1) already is the inverse of the Fourier transform, which implies that one actually deals with the inverse transform rather than the (forward) transform when talking about the Fourier series. The *Dirichlet* conditions give criterias under which conditions the Fourier series exactly approximates a function:

Dirichlet Conditions [19] Let $f(x)$ be a real or complex valued function. The Fourier series of f equals exactly the original function f if the following conditions are all satisfied on some finite interval $[c, c+T]$:

1. f is bounded on $[c, c+T] \subset \mathbb{R}$ with $c \in \mathbb{R}$ and $T \in \mathbb{R}^+$
2. f has a finite number of minimas and maximas on the interval
3. f has a finite number of discontinuities and
4. f is periodic on the interval $[c, c+T]$, i.e. with period T

further, for a given point $x \in [c, c+T]$, $\epsilon > 0$ and denoting $f(x^+) = \lim_{\epsilon \to 0} f(x+\epsilon)$ and $f(x^-) = \lim_{\epsilon \to \epsilon} f(x-\epsilon)$ respectively, then

$$f(x^+) = f(x^-) = f(x) = \sum_{k=-\infty}^{\infty} c_k e^{i2\pi x \frac{k}{T}}$$

where the coefficients c_k are defined as in (3.2).

The Dirichlet conditions are sufficient conditions that guarantee that the Fourier series is equal to the function f. However, not every function meets the Dirichlet conditions. This is often especially the case for practical problems. Therefore, it is reasonable to ask what happens to the approximation quality if at least one condition is violated. Of special interest for this question is in particular the periodicity condition. If the periodicity condition is violated, one still can try to approximate the function with a Fourier series though. Then, the function f will be exactly approximated in the interval $[c, c+T]$ but outside the deviation of the original function can be arbitrarily large. As a consequence, figuratively spoken, one can arrive at the continuous Fourier transform by extending the bounds of the interval $[c, c+T]$ to $[-\infty, \infty]$, i.e. when the period reaches infinity. Due to that property, it also makes sense applying the Fourier series approximation to functions which are not periodic but for which only a certain subset of the domain of definition is of interest. Situations in which the periodicity condition does not hold often occur in practical settings and one then pretends that a given function is periodic outside of the sub domain of interest and therefore can continue to deal with an exact approximation within. Fourier methods are only occasionally used in the machine learning context. For instance, [73] apply Fourier to approximate the calculation of the the kernel function. This approach was extended by [107] by stating another method for approximating the kernels. The application of the Fourier series approximation in the thesis requires some further considerations and results.

A very important set of functions that can be represented as a Fourier series are functions from a space known as *Isotropic Sobolev spaces*. The reason is that these functions provide a certain type of smoothness. A function is called smooth if all of it's partial derivatives of any order exist and are continuous. The Isotropic Sobolev space is defined by:

$$\mathcal{H}^s(\mathbb{R}) := \left\{ f(x) \in \mathcal{L}^2 : \sum_{0 \le v \le s} ||D^v f||^2_{\mathcal{L}^2} < \infty \right\}.$$

Here, D^v denote the derivatives up to degree s. So far this definition is not explicitly related to functions that are represented by the Fourier series. However, it is possible to state an equivalent definition in the trigonometric setting by applying the continuous mapping $\left((1+|\cdot|^2)^{-\frac{s}{2}}\hat{u}(\cdot)\right)^v$. Then one gets an equivalent definition of the Soblev space by [48]:

$$\mathcal{H}^s = \left\{ f(x) = \sum_{k\in\mathbb{Z}} c_k e^{-i2\pi kx} : \sum_{k\in\mathbb{Z}} (1+|k|)^{2s}|c_k|^2 < \infty \right\}.$$

The smoothness properties of functions from $\mathcal{H}^s(\mathbb{R})$ have direct implications for the decay of the Fourier coefficients c_k [1, 8, 97]. Therefore, the latter set is characterized by the decay of the c_k coefficients, i.e. each function f for which the coefficients of it's Fourier series decay like $|c_k| \leq C(1+|k|)^{-s}$ belongs to $\mathcal{H}^s$. This decay of the coefficients will become especially important in the next sections for the construction of the Hyperbolic cross approximation of the Fourier series in higher dimensional spaces.

Another problem that occurs in practical applications is the actual calculation of the Fourier series itself. Since the calculation of the infinite sum is always infeasible it is necessary to approximate the Fourier series. This is commonly achieved by simply truncating the sum of the Fourier series at some integer $K \in \mathbb{N}$:

$$f(x) = \sum_{k=-K}^{K} c_k e^{i2\pi x\frac{k}{T}}.$$

The consequence of this approach is the lack of contribution of higher frequencies (encoded by large values of k) in the approximation of the original function f. By truncating the Fourier series one implicitly restricts the space of functions that are considered. Thus, this approach can be understood as some kind of regularization, a.k.a. regularization by discretization [48, 60]. The degree of truncation, however, has a major impact on the approximation quality of the Fourier series of the function f. For a given function $f \in C^{\nu,\beta}(2\pi)$ that is 2π-periodic, has continuous derivatives up to order $\nu \in \mathbb{N}$, and is Hölder continuous with Hölder exponent $\beta \in (0,1]$ it holds that $||f - \mathcal{F}_K(f)|| \leq \rho_\nu \frac{\log K}{K^{\nu+\beta}}$, where $\mathcal{F}_K(f)$ denotes the truncated Fourier series up to degree K, and ρ_ν is a constant that linearly depends on the Hölder constant M_ν of $f^{(\nu)}$ [9]. In a common approximation setting, in which one is interested in approximating f very accurately, one normally would like to truncate the Fourier series as less as possible in order to get a very precise approximation of high quality. This becomes particularly important when the function f is very volatile or finely structured at some points (see figure 3.1 for an illustrative example). Those volatile or finely structured parts are mostly captured at points of high frequency, i.e. large values for k. On the other hand the lower values (i.e. low values of k) of the frequency spectrum approximate the dominating (global) structure of the function f. Figure 3.1 demonstrates the effect of the truncation of the Fourier series and the consequences for the quality of approximation. The plot shows how fine

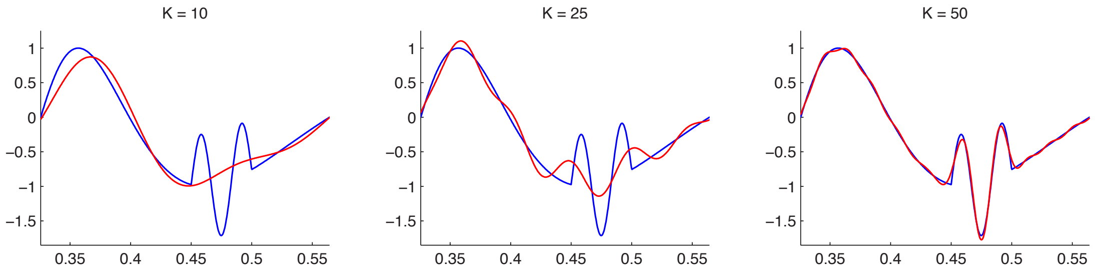


Figure 3.1: Illustrative example of the influence of the truncation of the Fourier series (red) on the quality of the function approximation. Lower values of K mainly capture the general structure of a function f (blue) while high values of K approximate finer structures. The result of such a truncation can therefore be interpreted as some kind of smoothing. In certain situations this behaviour can be useful.

structures are better approximated when K grows. Although not being shown in the plot, at $K = 100$ the approximation is very accurate. On the other hand, reducing the integer K to lower values means that the approximation only captures lower frequencies and thus represents the major global structure of the function f. This can be useful if finer grained areas of the function are not of interest or if these fine areas are the result of some (random) distortions. Then one could say that the transform is some kind of filtering that filters the perturbations that are only reflected in these high frequencies from the Fourier series. This can become, for example, very useful in situations when a function has been regressed from data. The resulting fitted function might be very varying due to unknown perturbations the data has been exposed to. An appropriate truncation then could soften the approximation and therefore yields a better fit. Section 7.2.1 will carry out this idea and give examples when such a truncation makes sense.

3.1.1 Higher Dimensional Fourier Transform

So far the Fourier transform was only considered for the one dimensional space. The extension to higher dimensions can be achieved by a tensor product construction:

$$\mathcal{F}(f)(x) := \int_{\mathbb{R}} \dots \int_{\mathbb{R}} e^{-2\pi x^t \xi} f(x) dx_D \dots dx_1$$

where $x, \xi \in \mathbb{R}^D$ and $x^t \xi$ denotes the dot product. The Fourier series is then given by:

$$f(x) = \sum_{k_1=-\infty}^{\infty} \dots \sum_{k_D=-\infty}^{\infty} c_{k_1,\dots,k_D} e^{i2\pi \sum_{d=1}^{D} \frac{k_d}{T_d} x_d} \tag{3.3}$$

$$c_k = \frac{1}{\prod_{d=1}^{D} T_d} \int_{t_1}^{t_1+T_1} \dots \int_{t_D}^{t_D+T_D} f(x) e^{-i2\pi \sum_{d=1}^{D} \frac{k_d}{T_d} x_d} dx. \tag{3.4}$$

As in the one dimensional case a numerical approximation of the Fourier series requires a truncation at some integer $K \in \mathbb{N}$. Although one could consider different Ks for each dimension and for the sake of simplicity it is assumed that K is the same for each dimension. As for the one dimensional case the aspects about the approximation of a function of higher dimensions generalize as well. However, although being a simple generalization to the multidimensional case one quickly runs into numerical problems when trying to practically calculate the Fourier series for higher dimensions. The problem that arises is the exponential growth of the computational costs one has to face. Considering that the one dimensional case requires just the calculation of $2(K+1)$ coefficients c_k the calculation of approximations of dimensions of the order of $D \in \mathbb{N}$ require $(2(K+1))^D$ which grows too quickly for high dimensional functions. Thus, the calculation of the Fourier series is prone to the curse of dimensionality leading to a situation in which the computation of a sufficiently good approximation becomes infeasible. As a consequence, it is virtually impossible to calculate the Fourier series in an adequate quality for the full set of coefficients. The solution to this difficulty has to be a compromise or trade off between the number of coefficients c_k taken into account and the accuracy of the approximation such that the result is of acceptable quality while providing tolerable computational costs. An approach that pursues such a compromise is the so called Hyperbolic cross which will be introduced in the next section.

3.2 Hyperbolic Cross

The previous section highlighted the problems that arise when trying to calculate the Fourier transform in higher dimensional spaces. The question is whether it is possible to give an alternative to the full calculation that reduces the computational efforts while achieving good approximation properties. In fact, an alternative has been very well investigated in the past [10, 87]. The idea is to approximate the Fourier series by employing only a very selective subset of the full coefficients c_k. Preferably the coefficients that contribute the most to the approximation of the function should be taken into account while simultaneously dropping those that do not have a huge impact. Under certain assumptions on the function, an approach for achieving this is the so called Hyperbolic cross [10, 87] - a method that basically is said to describe a special selection scheme for the coefficients c_k. As a result, the Hyperbolic cross approximation significantly reduces the computational costs while simultaneously keeping a high approximation accuracy. This section will give an introduction to the Hyperbolic cross, how it can be constructed, and what the benefits are. Fortunately, the application of the Hyperbolic cross requires just a certain smoothness of a function f. A function is called smooth if all of its partial derivatives of any order exist and are continuous:

$$D^{\upsilon} f(x) := \frac{\partial^{|\upsilon|}}{\partial^{\upsilon_1} x_1 \ldots \partial^{\upsilon_D} x_D} f(x). \tag{3.5}$$

Here, $\upsilon = (\upsilon_1, \ldots, \upsilon_D) \in \mathbb{N}^D$ is a multi-index of length D and $|\upsilon| = \upsilon_1 + \ldots + \upsilon_D$ is called the order. As stated in the previous section, this smoothness of functions applies to function from the Sobolev space $\mathcal{H}^s$ and has implications for the decay of the Fourier coefficients c_k. The application of the Hyperbolic cross requires some further decay properties that are ensured by considering a special combination of several one dimensional Sobolev spaces in form of tensor products [94]. The relationship between the tensor product space and the decay of the Fourier coefficients - as for the one dimensional case - has also been very well investigated [16, 28, 63] and extends to the multidimensional setting. The extension, however, is not straight forward and requires a modification of the existing definition. In particular, each dimension can be combined differently with another dimension by providing different types of combined decay criterias [38]. The Hyperbolic cross considers additive and multiplicative combinations for the characterization of the coefficient decays.

In this particular case the set of functions taken into account is defined by the following set:

$$\mathcal{H}^{t,l}_{mix} := \left\{ f(x) = \sum_{k \in \mathbb{Z}^D} c_k e^{-ikx} : ||f||_{\mathcal{H}^{t,l}_{mix}} = \left(\sum_{k \in \mathbb{Z}^D} \prod_{i=1}^{D} (1 + |k_i|)^{2t} (1 + |k|_\infty)^{2l} |c_k|^2 \right)^{\frac{1}{2}} < \infty \right\}$$

for $-\infty < t, l < \infty$ and $|k|_\infty = \max_{1 \leq i \leq D}\{k_i\}$. This special construction is a two parametric Sobolev space that contains the isotropic as well as the spaces with dominated mixed smoothness. By appropriately setting the parameters t and l one gets:

$$\mathcal{H}^s = \mathcal{H}^{0,s}_{mix} = \left\{ f(x) = \sum_{k \in \mathbb{Z}^D} c_k e^{-ikx} : ||f||_{\mathcal{H}^s} = \left(\sum_{k \in \mathbb{Z}^D} (1 + |k|_\infty)^{2l} |c_k|^2 \right)^{\frac{1}{2}} < \infty \right\}$$

the isotropic Sobolev space and correspondingly the so-called space of dominated mixed smoothness:

$$\mathcal{H}^{t}_{mix} = \mathcal{H}^{t,0}_{mix} = \left\{ f(x) = \sum_{k \in \mathbb{Z}^D} c_k e^{-ikx} : ||f||_{\mathcal{H}^{t,0}_{mix}} = \left(\sum_{k \in \mathbb{Z}^D} \prod_{i=1}^{D} (1 + |k_i|)^{2t} |c_k|^2 \right)^{\frac{1}{2}} < \infty \right\}.$$

The name dominated mixed smoothness refers from the fact that the norm is equivanlent to:

$$||f||^2_{\mathcal{H}^t_{mix}} \equiv \sum_{0 \le \upsilon \le t} ||f^{(\upsilon)}||^2_{\mathcal{L}^2}$$

where $f^{(\upsilon)}, \upsilon \in \mathbb{N}^D$ a multi-index, denotes the general mixed derivative. The main aspect about these spaces is that they are all characterized by the decay of the Fourier coefficients. In particular the Fourier coefficients of a function from $\mathcal{H}^{t,l}_{mix}$ are bounded by $|c_k| \le C \left(\prod_{i=1}^D (1+|k_i|)^t (1+|k|_\infty)^l \right)^{-1}$ and $k \in \mathbb{Z}^D$. This kind of decay provides a guideline for the construction of the Hyperbolic cross. For the construction one needs to define two sets. Let $D \in \mathbb{N}$ the dimension and T a parameter with $T \in (-\infty, 1]$. The first set is the set of indeces or frequencies:

$$FI^T_d := \left\{ k \in \mathbb{Z}^D : \prod_{i=1}^D (1+|k_i|) \cdot (1+|k|_\infty)^{-T} \le (1+d)^{1-T} \right\} \tag{3.6}$$

and the second set defines the actual function approximations:

$$FV^T_d := \left\{ f(x) = \sum_{k \in FI^T_d} c_k e^{-ikx} \right\}.$$

From the definition, one can see that the sets only depend on the parameters T and d. The first parameter T describes the number of coefficients taken into account. Here the relationship is reversed meaning that a smaller (negative) value implies a larger set while a bigger reduces the number of elements. The second parameter d controls the number of frequencies taken into account in each direction. This value normally matches the truncation value of the truncated Fourier series. When $T \to -\infty$ one can give a natural extension as $FI^{-\infty}_d := \{k \in \mathbb{Z}^D : |k|_\infty \le d\}$ and analogously the set of approximations by $FV^{-\infty}_d$. This extension exactly matches the normal case, i.e. when the full (cubical) space of $[-d,d]^D$ is considered which corresponds to the complete set of coefficients c_k of the truncated Fourier series. By keeping the parameter d fixed the sets become nested with respect to the value of T, i.e.:

$$FV^{T_1}_d \subset FV^{T_2}_d, \text{ for } 1 \ge T_1 > T_2 > -\infty.$$

The standard Hyperbolic cross now arises by fixing the parameter $T = 0$ and arbitrary parameter d [87]. Thus the Hyperbolic cross is just a special case of $FV^T_d = FV^0_d$. An example plot of the Hyperbolic cross and other realizations of the index sets FI^T_d are given in figure 3.2.

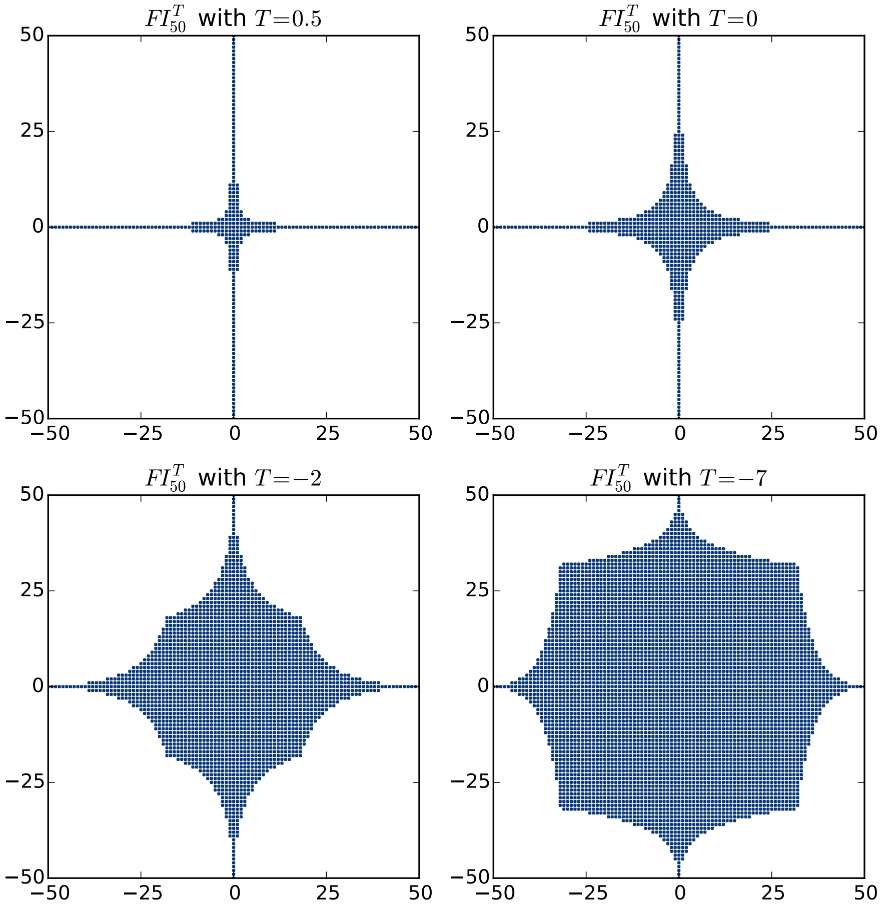


Figure 3.2: Different index sets FI_{50}^{T} (defined in 3.6) for different parameters $T \in \{0.5, 0, -2, -7\}$ and $d = 50$. The special case $T = 0$ is also referred to as Hyperbolic cross.

Since the Hyperbolic cross is an approximation of the Fourier series, it is desirable to state analytical bounds on the approximation quality. The following theorem [50] states such bounds for different parameter combinations d and T and for different functions:

Let $s < l + t, t \geq 0, u \in \mathcal{H}^{t,l}_{mix}, u(x) = \sum_k c_k e^{-ikx}$ and $u_d^T = \sum_{k \in FI_d^T} c_k e^{-ikx} \in FV_d^T$. Then the following holds:

$$||u - u_d^T||_{\mathcal{H}^s} \leq \begin{cases} (1+d)^{s-l-t+(Tt-s+l)\frac{D-1}{D-T}} ||u||_{\mathcal{H}^{t,l}_{mix}} & \text{if } T \geq \frac{s-l}{t} \\ (1+d)^{s-l-t} ||u||_{\mathcal{H}^{t,l}_{mix}} & \text{if } T < \frac{s-l}{t} \end{cases} \tag{3.7}$$

This theorem guarantees certain error bounds for functions from the $\mathcal{H}^{t,l}_{mix}$ that are approximated by the Hyperbolic cross. The particular error bounds for the Hyperbolic cross are given for $T = 0$ as [50]:

$$||u - u_d^0||_{\mathcal{H}^s} \leq \begin{cases} (1+d)^{s-l-t+(s+l)\frac{D-1}{D}} ||u||_{\mathcal{H}^{t,l}_{mix}} & \text{if } 0 \geq \frac{s-l}{t} \\ (1+d)^{s-l-t} ||u||_{\mathcal{H}^{t,l}_{mix}} & \text{if } 0 < \frac{s-l}{t} \end{cases}$$

Thus, the quality of the approximation will depend on the choice of parameters or functions considered that are to be approximated. On the other side, the computational costs decrease significantly. The effort that is necessary is given by [50]:

$$\begin{aligned}
&\mathcal{O}\left(1+d\right) && \text{for } 0 < T \leq 1\\
&\mathcal{O}\left((1+d)\log(1-d)^{D-1}\right) && \text{for } T = 0\\
&\mathcal{O}\left((1+d)^{T-1/\frac{T}{D-1}}\right) && \text{for } T < 0\\
&\mathcal{O}\left((1+d)^{D}\right) && \text{for } T = -\infty.
\end{aligned}$$

Therefore, the computational costs for the calculation of the Hyperbolic cross are reduced to $\mathcal{O}\left((1+d)\log(1-d)^{D-1}\right)$ in constrast to the full set $\mathcal{O}\left((1+d)^{D}\right)$. This makes the computation of an approximated Fourier series in higher dimensional space feasible by simultaneously keeping a good approximation of the original problem. In later chapters, these results will be applied in order to calculate the Fourier series approximations.

3.3 Fourier Series, Hyperbolic Cross and Monte Carlo Method

This section will discuss the approximation properties of the combination of the Fourier series approximation by the Hyperbolic cross and the Monte Carlo method introduced in chapter 2. The analysis presented here is of importance for chapter 7 which will apply this combination in order to state new and improved data analysis methods.

For this analysis, it is assumed that an arbitrary distribution $p : \mathbb{R}^D \to \mathbb{R}^+$ is given (where D denotes the dimension) with a compact support $[c, c+T]^D \subset \mathbb{R}^D$ that satisfies the requirements for the application of the Fourier series approximation. To simplify the presentation the case $D = 1$ is considered. The Fourier series is then given by:

$$p(x) = \sum_{k=-\infty}^{\infty} c_k e^{i2\pi x \frac{k}{T}} \quad \text{and} \quad c_k = \frac{1}{T}\int_c^{c+T} p(x) e^{-i2\pi x \frac{k}{T}} dx.$$

Of special interest here is in particular the expression for the coefficients c_k since they denote in fact the analytic calculations of the expectation of the random variable $e^{-i2\pi x \frac{k}{T}}$ w.r.t. the distribution p, i.e. $\mathbb{E}_p\left[e^{-i2\pi x \frac{k}{T}}\right]$. Therefore, the application of the Monte-Carlo method discussed in the previous chapter 2 can be applied. T do so, one needs a dataset sample $\{x_1, \ldots, x_L\} \subset \mathbb{R}^{D\times L}$ (here $D = 1$) which has been sampled according to the distribution p. Then the expectation can be approximated by:

$$c_k = \frac{1}{T}\int_c^{c+T} p(x) e^{-i2\pi x \frac{k}{T}} dx \approx \frac{1}{TL}\sum_{l=1}^{L} e^{-i2\pi x_l \frac{k}{T}}.$$

For a complete understanding of the approximation properties of this approach, one has to consider the approximation quality of the (truncated) Fourier series or, when applied, of the Hyperbolic cross and those of the empirical mean. Since the empirical mean highly depends on the number of available (random) data samples it is only possible to state error bounds for the empirical approximation of the integral for the computation of the Fourier coefficients in a probabilistic manner. An appropriate way to do so is the derivation of confidence intervals. The following observations are based on statistics and the theory of Monte Carlo methods, further details can for example be found in [78] or [30].

The setting is now extended to the D-dimensional case for a more general and complete analysis. Then:

$$\begin{aligned} c_{k_1,\dots,k_D} &= \frac{1}{\prod_{d=1}^{D} T_d} \int_{t_1}^{t_1+T_1} \cdots \int_{t_D}^{t_D+T_D} e^{-i2\pi \sum_{d=1}^{D} \frac{k_d}{T_d} x_d} dp(x) \\ &\approx \frac{1}{L \prod_{d=1}^{D} T_d} \sum_{l=1}^{L} e^{-i2\pi \sum_{d=1}^{D} \frac{k_d}{T_d} x_{dl}} =: \hat{c}^{(L)}_{k_1,\dots,k_D}. \end{aligned} \tag{3.8}$$

Since the function $e^{-i2\pi \sum_{d=1}^{D} \frac{k_d}{T_d} x_{dl}}$ is integrable, and the empirical mean is a sum of independent and identically distributed samples, expression (3.8) converges to the expectation $c_{k_1,\dots,k_D}$. Therefore, as a result of the central limit theorem the sum is distributed normally, i.e.

$$\hat{c}^{(L,0)}_{k_1,\dots,k_D} := \left(\hat{c}^{(L)}_{k_1,\dots,k_D} - c_{k_1,\dots,k_D} \right) \sim \mathcal{N}\left(0, \frac{\left(\eta^{(k_1,\dots,k_D)}\right)^2}{L} \right),$$

here, $\left(\eta^{(k_1,\dots,k_D)}\right)^2$ denotes the variance for the combination $(k_1,\dots,k_D)$. Note that for simplicity, the real exact mean $c_{k_1,\dots,k_D}$ has been subtracted in order to center the distribution at mean 0. Although the analytic variance $\left(\eta^{(k_1,\dots,k_D)}\right)^2$ is, in general, unknown it is possible to use the empirical variance instead for practical calculations. The unbiased sample variance is given by:

$$\left(\eta^{(k_1,\dots,k_D)}\right)^2 \approx \left(\hat{\eta}_L^{(k_1,\dots,k_D)}\right)^2 = \frac{1}{(L-1)\prod_{d=1}^{D} T_d^2} \sum_{l=1}^{L} \left(e^{-i2\pi \sum_{d=1}^{D} \frac{k_d}{T_d} x_{dl}} - c_{k_1,\dots,k_D} \right)^2.$$

Hence, the $(1-\gamma)\%$ confidence interval for expression (3.8) can be stated as:

$$\left[\hat{c}^{(L,0)}_{k_1,\dots,k_D} - z_{\left(1-\frac{\gamma}{2}\right)} \frac{\hat{\eta}_L^{(k_1,\dots,k_D)}}{\sqrt{L}} \;,\; \hat{c}^{(L,0)}_{k_1,\dots,k_D} + z_{\left(1-\frac{\gamma}{2}\right)} \frac{\hat{\eta}_L^{(k_1,\dots,k_D)}}{\sqrt{L}} \right]$$

where $z_{\left(1-\frac{\gamma}{2}\right)}$ denotes the $1-\frac{\gamma}{2}$ quantile of the standard normal distribution.

Considering the complete Fourier series, one can give a point-wise confidence interval, for each x. To do so, the definition of the covariance for two distinct $\hat{c}^{(L,0)}_{k_1,\ldots,k_D}$ is required. Given two distinct combinations $(k_1,\ldots,k_D),(\tilde{k}_1,\ldots,\tilde{k}_D) \in [-K,K]^D \subset \mathbb{Z}^D$, where at least one $k_d \neq \tilde{k}_d, d \in \{1,\ldots,D\}$, the covariance is given by $\operatorname{Cov}\left(\hat{c}^{(L,0)}_{k_1,\ldots,k_D}, \hat{c}^{(L,0)}_{\tilde{k}_1,\ldots,\tilde{k}_D}\right)$. Then the completed Fourier series approximation of function $p(x)$ at an arbitrary but fixed point x can be understood as a sum of the random variables:

$$\operatorname{Var}(p(x)) \approx \operatorname{Var}\left(\sum_{k_1=-K}^{K} \cdots \sum_{k_D=-K}^{K} \hat{c}^{(L,0)}_{k_1,\ldots,k_D} e^{i2\pi \sum_{d=1}^{D} \frac{k_d}{T_d} x_d}\right)$$

Since x is arbitrary but fixed each expression $e^{i2\pi \sum_{d=1}^{D} \frac{k_d}{T_d} x_d}$ can be considered constant, i.e. $\varsigma^{(k_1,\ldots,k_D)} := e^{i2\pi \sum_{d=1}^{D} \frac{k_d}{T_d} x_d}$. Being a sum of random variables multiplied by coefficients the following calculation holds:

$$\begin{aligned}
\operatorname{Var}(p(x)) \approx \widehat{\operatorname{Var}}(p(x)) &:= \operatorname{Var}\left(\sum_{k_1=-K}^{K} \cdots \sum_{k_D=-K}^{K} \hat{c}^{(L,0)}_{k_1,\ldots,k_D} \varsigma^{(k_1,\ldots,k_D)}\right) \\
&= \sum_{k_1=-K}^{K} \cdots \sum_{k_D=-K}^{K} \left(\varsigma^{(k_1,\ldots,k_D)}\right)^2 \operatorname{Var}\left(\hat{c}^{(L,0)}_{k_1,\ldots,k_D}\right) \\
&+ \sum_{k_1,\tilde{k}_1=-K}^{K} \cdots \sum_{k_D,\tilde{k}_D=-K}^{K} \mathbb{1}_{(k_1,\ldots,k_D)\neq(\tilde{k}_1,\ldots,\tilde{k}_D)} \varsigma^{(k_1,\ldots,k_D)} \varsigma^{(\tilde{k}_1,\ldots,\tilde{k}_D)} \operatorname{Cov}\left(\hat{c}^{(L,0)}_{k_1,\ldots,k_D}, \hat{c}^{(L,0)}_{\tilde{k}_1,\ldots,\tilde{k}_D}\right)
\end{aligned}$$

Here, $\mathbb{1}_{(k_1,\ldots,k_D)\neq(\tilde{k}_1,\ldots,\tilde{k}_D)}$ denotes the indicator function that is 1 if at least one $k_d \neq \tilde{k}_d, d \in \{1,\ldots,D\}$. Due to a special version of the central limit theorem (Lyapunov CTL) this sum converges to a normal distribution and the bounds for the $(1-\gamma)\%$ confidence intervals for this expression are given by:

$$p(x) \pm z_{\left(1-\frac{\gamma}{2}\right)} \sqrt{\widehat{\operatorname{Var}}(p(x))}.$$

The latter expression holds for a truncated version of the Fourier series. Additionally, the coefficient indeces can be replaced by those of the Hyperbolic cross. Therefore, one can achieve a probabilistic estimate of the point-wise difference between the Hyperbolic cross approximation using exact Fourier coefficients and the one employing approximated Fourier coefficients based on empirical means, where the estimate involves a variance depending on the given point x, which goes to zero for larger numbers of data. Thus, the error of the latter expression primarily depends on the amount of given data.

Chapter 4

Dataset Shifts in Machine Learning

This chapter is an introduction to *Dataset Shifts in Machine Learning*. The first part states the well known standard machine learning setting by introducing the mathematical framework and terms like model, classification, regression, discriminative and generative models. The second part is about new concepts that can occur in machine learning settings that have become known as *Dataset Shifts*. Important types of dataset shifts are explained and mathematically defined. Of special interest for this thesis are in particular the covariate and source component shift. These types of shifts will be investigated in later chapters in more detail. The dataset shift part follows mainly after the book [71]. The main goal of this chapter is to give the reader an overview about what dataset shifts are.

4.1 Traditional Machine Learning setting

When talking about traditional machine learning most people refer to supervised machine learning. It is a term that describes a situation in which one tries to learn an unknown functional relationship between some source domain $\mathcal{X} \subset \mathbb{R}^D$ and a target domain $\mathcal{Y} \subset \mathbb{R}$. Formally spoken the goal is to learn a function $f : \mathcal{X} \to \mathcal{Y}$. These functional relationships are typically referred to as regression when $\mathcal{Y} \subset \mathbb{R}$ or classification when $\mathcal{Y} \in \{y_1, \ldots, y_n\}$ with each $y_i \in \mathbb{R}$ an arbitrary discrete value. If $\mathcal{Y} \in \{0, 1\}$ or $\mathcal{Y} \in \{-1, 1\}$ the classification task is also referred to as binary classification. For the sake of simplicity this work will assume $\mathcal{Y} \subset \mathbb{R}$ since this also includes the special case of classification. An explicit remark will be given for situations in which it is appropriate to differentiate between both types of target domains. The mathematical treatment of such problems requires several assumptions in order to state a practical analytical framework. Firstly, it is assumed that one has given some data $X \subset \mathbb{R}^{D \times N}$, the so called covariates and the corresponding labels $Y \subset \mathbb{R}^N$. Equivalent terms for the covariates are explanatory variable, independent variable or predictors. Other common terms for labels are dependent variable or regressand. The covariates are assumed to be a subsample from the domain of the unknown function f and the labels correspond to the output of f, i.e. $f(x) = y$. This subsampling however follows certain rules which by assumption can be modeled by an unknown probability distribution p meaning that the subsamples X, Y are considered to be the result of a sampling that is distributed according to p, i.e. $\{X, Y\} \sim p(x, y)$.

The main goal of machine learning applications is the approximation or inference of the functional relationship f between $\mathcal{X}$ and $\mathcal{Y}$ by applying the datasets X, Y which was sampled from p. Since the dependencies of y on x by f are reflected in the distribution of the sampled data one can approximate the relationship by approximating the distribution p. However, the original p is unknown and therefore one has to make assumptions about it by assuming that p can be stated by a model $\hat{p}$ that depends on some set of parameters denoted by $\Theta \subset \mathbb{R}^D$. The idea is that given the data X, Y and the model $\hat{p}$ one tries to find an appropriate set of admissible parameters $\theta \in \Theta$ such that the model gives a best explanation of the data, i.e. the model needs to be fitted to the data by adjusting the parameters. However, since the model $\hat{p}$ itself is an assumption the fit can be arbitrarily bad. Therefore a variety of models exist in order to face different situations. The quality or goodness of fit is generally determined by applying a so called error or loss function $\ell(\cdot)$. While the chosen model is an assumption about the structure of the problem at hand the error or loss function ℓ evaluates which parameters $\theta \in \Theta$ fit the model best to the given data. The determination of such parameters with respect to the loss function is commonly achieved by solving an optimization problem. Since the parameter fitting requires the given sample data X, Y and the model $\hat{p}$ the problem can be formulated as:

$$\theta_{\text{opt}} = \text{argmax}_{\theta \in \Theta} \ell \left(\hat{p}(x, y | \theta), X, Y \right)$$

where ℓ denotes a loss function. The resulting parameters θ_{opt} fully specify the model $\hat{p}$. Although there exist a large number of models for solving machine learning problems all these models can be categorized into just two general types of models.

4.2 Generative Models and Discriminative Models

Modern machine learning approaches assume that the given or observed data was generated by or sampled from a probability distribution $p(x, y)$ and one has to use models for approximating the data. These models can be categorized into two complementary types of general models which are derived by applying the multiplication rule of probability. The first one is the so called generative model which is given by:

$$p(x, y) \approx \hat{p}(x, y) = \hat{p}(x|y)\hat{p}(y). \tag{4.1}$$

This approach fully models the joint distribution over the covariates x and dependent variables y. Once the model has been completely learned, it can be used to, as the name suggests, generate new samples, as it tries to explain all relationships and dependencies of the variables. The second class of models are the so called discriminative models or conditional models which are given by:

$$p(x, y) \approx \hat{p}(x, y) = \hat{p}(y|x)\hat{p}(x) \tag{4.2}$$

In practice, however, as a model one only tries to fit the conditional $\hat{p}(y|x)$ and the distribution of the covariates $p(x)$ is omitted. This is because a model for the covariates is of no interest - since they are already given - but only the conditional dependency is. As a consequence, the model can be much simpler than a generative model because potential complex dependencies between covariates are ignored. Therefore, the model is fully stated by modelling the conditional $\hat{p}(y|x)$. Important discriminative methods are for instance Logistic Regression [43] or Support Vector Machines [84]. The most significant disadvantage of discriminative models is that it can not be used to generate data.

Discriminative models are often used in supervised settings where one does not need or is not interested in the distribution of the covariates since one is only interested in the relationship of covariates x and dependent variable y but not in the dependencies in between the covariates. Ignoring such dependencies or correlations between the covariates yields a simpler model and reduces the potential risk of model miss specification because the number of assumptions about the structure of the problem is reduced and hence the sources of errors as well. In contrast, unsupervised methods often require the complete dependencies of all variables because unsupervised approaches commonly have to take strong assumptions about the structure of the data into account in order to get an appropriate fit. With that, a higher risk of model misspecification comes along and therefore generative models are very appropriate in cases where the properties about the data generating process are very well understood. Examples for generative models include Naïve Bayes (very high dimensional sparse data) [61], Hidden Markov Models (sequential data) [72] or Gaussian Mixture models (hierarchical clustering) [104].

4.3 Model Prediction

In the context of machine learning the purpose of models, as they have been introduced in section 4.2, is making predictions. In order to make predictions one has to fit a model $\hat{p}$ to a given set of so called training data X, Y which is assumed to be generated by some unknown distribution p. The fitting process requires the selection of a parameter θ from a parameter set Θ that fully specifies the model. The resulting model does no longer require any data since the information contained within the data is represented by the inferred parameters. After having learned an appropriate parameter one can use the obtained model to make predictions for new unseen data. I.e. given a new data point $x^* \in \mathcal{X}$ one is interested in the prediction of the corresponding label $y^* \in \mathcal{Y}$. In the case of a generative model one would calculate:

$$y^* = \text{argmax}_y \hat{p}(x^*, y) = \text{argmax}_y \hat{p}(x^*|y)\hat{p}(y) \quad \text{and} \quad y^* = \text{argmax}_y \hat{p}(y|x^*) \tag{4.3}$$

for the discriminative case. In order to get the best possible predictions given the model, this approach assumes that both the training data X, Y and the new data x^* have been drawn from the same unknown distribution, i.e. p. If this assumption holds true the training data and the new data point come from the same structure and therefore the knowledge from the training data can be applied straight forwardly to the new unknown data. However, this is an ideal situation which might not always hold true in most settings. The next section gives an overview of challenges that might violate this standard assumption.

4.4 Dataset shifts in Machine Learning

In most applications the assumption of equally distributed training data, and data that is to be predicted makes sense and yields good prediction quality. However, there are also a lot of situations in which this assumption does not hold true. The online advertising market is a good illustrative example for explaining different types of dataset shifts.

As of 2015, online advertsing is currently facing a dramatic change. As technology is advancing new types of systems enable new types of businesses also in online advertising. Online advertising becomes more and more data driven which means the evaluation and prediction of data preferably in real time. Basically the business is made out of two types of parties. The first one are the so-called publishers. The term publisher stands for entities like persons, companies or organizations that operate web pages or mobile apps that beside the actual content (like news, blogs, etc.) also contain blank spaces reserved for placing online advertisements (ads) in form of display ads (for example banners). One such reserved blank space for one ad is commonly called a placement. The second party are the so called advertisers. Advertisers generally represent companies that want to promote their products on the internet in form of banners, videos, texts, etc. They are the ones that are responsible for the actual content of the display ads. In the (financial) interest of each others both parties have to be brought together. In general, however, advertisers do not approach publishers directly. Instead they are represented by agencies called DSPs (Demand Side Platforms) that provide additional services and a technical infrastructure to manage the display ads for them. On the other side, publishers do not directly provide access to their placements but commercialize them through agencies which are called SSPs (which stands for Service Side Platforms). SSPs are companies that provide a new type of technology called Real Time Bidding or RTB. RTB describes an auction market (also called RTB exchange) for buying impressions. An impression is a complete page load or site visit of a web page or mobile app of a publisher by some internet user. In the process of one RTB auction the first thing to happen is that an internet user visits the site of a publisher. The web server of the publisher then contacts the RTB server of the SSP and tells it that a new user just started to load the web page. Then an auction is opened for this particular user for the particular web page and placement. Now, usually, several DSPs start bidding on this impression and the highest bid will win the auction. After the auction has been finished the winning DSP is allowed to place it's ad on the web page of the publisher. This whole process happens within 100 milliseconds (therefore the term real time in RTB) while the user is loading the webpage. The user normally does not take any notice of this background process.

Since the requirement of 100 milliseconds is very short, it is no longer possible to perform an auction process manually. Instead, the whole process is completely automated and programs called bidders automatically perform the bidding for a placement on a RTB exchange. However since every user can be treated individually it is possible to pursue a bidding strategy where one is willing to bid higher on users that can be considered as valuable (or click friendly) and lower bids for users being less worthy. Therefore, a prediction of click probability, also called click through rate (ctr), is done by evaluating data about the user. By being able to track the user behaviour it becomes possible to gather huge amounts of different data about each internet user.

This data then is normally given by $X \subset \mathbb{R}^{D \times N}$ the measurements of user features, and $Y \subset \mathbb{R}^N$ the historically observed click probability (ctr). However, usually this data X, Y is highly non-stationary which means it changes over time and thus the predictive power of a certain feature or attribute changes as well. Mathematically, this situation can be described by actually having two different datasets within the complete dataset X, Y. The first one is of the current time period or today and will be denoted by $(X^P, Y^P) \subset \mathbb{R}^{D+1\times M}, M \in \mathbb{N}$, the primal or P data also called target or testing data. This data is the data for which one would like to perform predictions and is therefore of primary interest. It is distributed according to the primal distribution $p^P(x, y)$. The second dataset also referred to as secondary or S data $(X^S, Y^S) \subset \mathbb{R}^{D+1\times N}, N \in \mathbb{N}$, a.k.a. source/training data, is distributed according to a secondary distribution $p^S(x, y)$ which is for instance from last month. This secondary dataset can be considered being a helper dataset since it is not of primary concern. To understand now why this is a problem, one should consider a model for ctr prediction that has been inferred on the S data from last months data. This model will most likely show good prediction performance on some control or evaluation data from the same time period but in todays live operations it dramatically fails and even makes false predictions resulting in inappropriate ctr predictions and a loss of money for the DSP and hence the advertiser. The reason for that lies in a drift or transformation of the distributions from which the two datasets have been sampled. Mathematically expressed, this means that:

$$p^S(x, y) \neq p^P(x, y). \tag{4.4}$$

In general, these two distributions are different since the data generating process is non-stationary. As a consequence, the obtained data for both time periods is differently distributed which causes a so called *dataset shift*. That means that the assumption of a standard machine learning setting is violated and thus the previously learned model is no longer applicable to the new unseen data. Even worse, it can become arbitrarily poor which in a real world setting can lead to dramatic losses. Therefore it makes sense to investigate these frequently occurring situations separately and consider them as *dataset shifts in machine learning*. Since expression (4.4) gives a rather general definition of such a context it is required to further characterize different types of dataset shifts. Briefly, typical types which are considered in the scientific literature [71] in the fields of machine learning are:

- Covariate shift - The functional relationship $p(y|x)$ remains but $p(x)$ changes.
- Prior Probability Shift - $p(y|x)$ changes and $p(x)$ remains.
- Sample Selection Bias - Application of a biased selection process to $\{\mathcal{X}, \mathcal{Y}\}$
- Imbalanced data - The dataset is dominated by one realization of y
- Source component shift - distributions p^P and p^S can differ arbitrarily

The following sub-sections will give an illustrative introduction to each of those dataset shifts. Since all kinds of explained dataset shifts can occur in the data provided by RTB exchanges the following examples will be motivated by typical situations in online advertising.

4.4.1 Covariate Shift

Covariate shift [71] is considered to be one of the simpler types of dataset shifts. Generally the model is inferred on the training/secondary or source data X^S, Y^S and a prediction is performed on the testing/primal or target data X^P, Y^P. As the name *covariate shift* suggests, the distributions of the covariates x differ. That means:

$$p(y|x)p^S(x) = p^S(x,y) \neq p^P(x,y) = p(y|x)p^P(x). \tag{4.5}$$

Here, the conditional $p(y|x)$ does not change but the covariate distribution $p^P(x)$ is not the same as $p^S(x)$. Informally speaking, while the functional relationship $p(y|x)$ stays the same the actual locations of the sampled data given by $p(x)$ differ between the two datasets. For instance consider a web page that at some time gets a lot of traffic from users that are interested in fashion. Therefore, display ads that are somehow related to fashion perform very likely better than ads about different topics. Consequently, if one would learn a model on the data given by the fashion users one would yield a very good model for the ctr on this particular group of users, i.e. $p(\text{click}|\text{user data}, \text{placement})$ is learned well. However, due to some changes on the web page (consider another news story) the topic changes and other different kind of users are now attracted to the new content. This new user group might be interested in technology and therefore the click probability for ads about tech products will be much higher. Although, the model learned on the fashion users still exhibits good prediction performance for user that are interested in fashion this model will perform not so well for the now much more frequently seen technology interested users. Mathematically loosely spoken the functional relationship $p(\text{click}|\text{user})$ remains the same but the sampling location $p(\text{user})$ has changed and the covariate distributions $p^S(\text{fashion interested user}) \neq p^P(\text{technology interested user})$ have changed. Figure 4.1 reflects this simple situation which can also be characterized by a change of sampling location of the data.

4.4.2 Prior Probability Shift

In contrast to the Covariate shift setting a Prior Probability shift [71] denotes a situation in which the distribution of the dependent variable y changes from secondary/source to

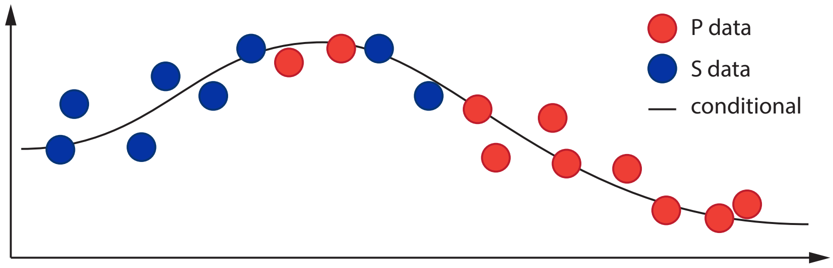


Figure 4.1: Illustrative example for a typical Covariate shift setting. While the functional relationship, i.e. the conditional $p(y|x)$, remains the same for all data points the "location" of the sampled data has changed between S and P data. The S data was obtained from other locations than the P data. As a consequence less information about the test data is contained within the training data.

primal/target dataset, i.e.:

$$p^S(x,y) = p^S(y|x)p(x) \neq p^P(y|x)p(x) = p^P(x,y).$$

Such a relationship is refered to as Prior Probability shift where the functional relationship $p(y|x)$ changes but the location of the sampled covariates $p(x)$ remains the same. A practical example from online advertising would be the application of so-called "retargeting". E-commerce websites like internet shops try to increase user engagement by providing DSPs with information about users visiting their shop. In combination with this shop behaviour the DSP tries to retarget the user by showing the users ads about products they looked at in the shop after they have visited the corresponding e-commerce website. This type of advertisement is therefore called user retargeting. From the collected DSP's data from the RTB exchanges it can be seen that the probability of clicking a retargeting ad and actually purchasing the product is the highest shortly after a user has visited the shop and then monotonically decreases over time. Therefore, every time a user is seen on a RTB exchange and does not click and buy the product the chances that the user is still interested in a purchase decreases. Hence, a purchase prediction model that has been learned on the purchase behaviour of recent users is not appropriate for users that have been already observed several times as it would be too optimistic. This is due to the shift in purchase probability which is the dependent variable in the model and implies a change in the functional relationship $p(y|x)$ over time although the user group $p(x)$ remains the same since the users and hence their characteristic properties remain the same. A graphical illustration of a Prior Probability shift is given in figure 4.2.

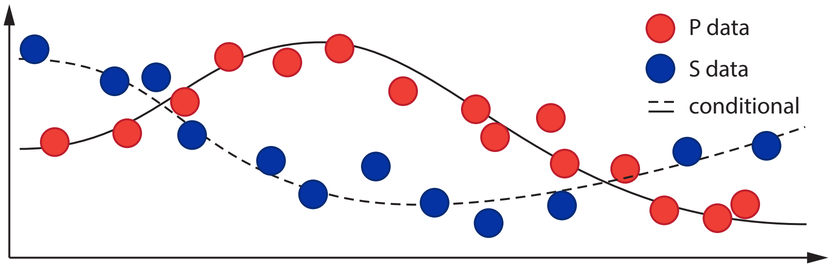


Figure 4.2: Figurative example of a Prior Probability shift setting. The functional relationship, i.e. the conditional $p(y|x)$, differ for the S and P datasets. However, in contrast to Covariate shift, the locations of the sampling process remain the same.

4.4.3 Sample Selection Bias

Sample Selection Bias [71] describes a dataset shift in which the sampled data was affected by a systematic bias in selecting samples from a distribution, i.e.:

$$\begin{aligned} & p^S(x,y) = p(x,y|c=1) = p(c=1|x,y)p(x,y) \\ \text{and} \quad & p^P(x,y) = p(x,y). \end{aligned}$$

That means that the target (or P) domain contains an unbiased data sample while for the source data a selection process has been applied before choosing a data point. Such a selection can be intentional. Consider a DSP that wants to learn a specific model for a subuser group interested in a particular topic in order to identify new similar users. In that case the DSP would learn a model on those users but would also apply the learned biased model to all new users in order to identify new users belonging to this interest group by investigating the click behaviour. However, although being an important dataset shift problem, Sample Selection Bias is not of any concern in this work.

4.4.4 Imbalanced Data

Imbalanced data [71] occurs primarily in classification settings where the dependent variable is discrete and one class is overrepresented. That means that the number of samples from one class strongly dominates the number of samples of all other classes. This type of imbalance has a huge impact on the prediction quality of a model since the model is very likely to consider just the examples that are ubiquitously available and further ignores the seldomly occurring but important other classes. Such a situation is typical in online advertisement since the number of clicks on ads is significantly lower than the number of ad impressions. As for Sample Selection Bias this type of dataset shift will not be investigated in this work.

4.4.5 Source Component Shift

Another important dataset shift is the Source Component shift [71] that can be considered as a mixture of covariate and prior probability shift. Source component shift normally results from a non-stationary stochastic process that is responsible for the observed/generated data and which changes (for instance) over time. As a consequence, two different dataset samples from such a process are the result of two different distributions that can differ in both the covariates and the dependent variables. The source component shift is characterized by:

$$p^S(x, y) \neq p^P(x, y) \text{ where } p^S(y|x) \neq p^P(y|x) \text{ and } p^S(x) \neq p^P(x).$$

Since the conditional and the covariate distribution are not necessarily the same, this situation can also be understood as a full shift in distribution. One can think about the non-stationarity of the generating process as a (smooth) transition from one distribution p^S to another p^P. However, this might not be the case. In reality, the two datasets X^S, Y^S and X^P, Y^P might also be the result of two independent underlying processes.

A practical example of a source component shift in the online advertising space is the combination of the retargeting shift (prior probability shift) and simultaneously a shift in the interests of some users (covariate shift). As in the previous prior probability shift section the dependent or conditional variable y, which is the ctr, decreases monotonically but now additionally the types of users also change. The given data X^P, Y^P might therefore be comprised of recent users that are more likely to click and that are fashion interested and older users X^S, Y^S which are less likely to click and that are maybe interested in shoes. Thus, a current model that has been learned on recent fashion interested users performs well in the prediction of the ctr, that is

$$\hat{p}(\text{ctr}|\text{recent-user}, \text{fashion-interested})\hat{p}(\text{recent-user}, \text{fashion-interested})$$

but this same model will probably show a poorer prediction performance on the shoes interested older users for which a model described by

$$\hat{p}(\text{ctr}|\text{older-user}, \text{shoes-interested})\hat{p}(\text{older-user}, \text{shoes-interested}) \tag{4.6}$$

will most likely perform better. In such a case one could characterize the situation as being the result of a data generating process that devolves from one type of an user to another. However, it must be mentioned that the source component shift can lead to situations that are virtually impossible to solve. The biggest problem in the source component shift setting is the dissimilarity of the two datasets or put differently the amount of variation of the underlying generating non-stationary process. A well known example for such a problem are stock markets. In the stock market the distributions are so highly non-stationary that in a practical setting it effectively becomes impossible

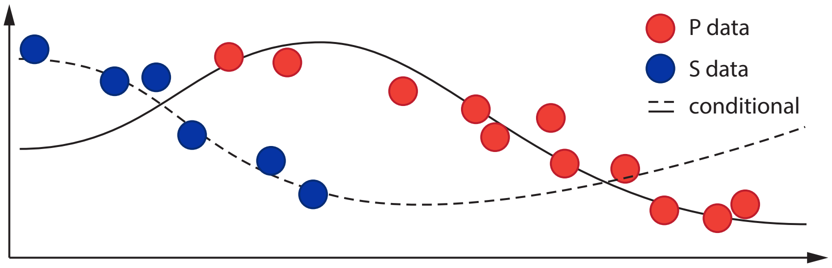


FIGURE 4.3: Toy example of a Source Component Shift setting. This type of shift can be understood as a mixture of Covariate and Prior Probability shift. It can be a particularly hard problem because in cases when both datasets differ too much a transfer of knowledge will become impossible. Thus, in order to tackle such a shift one must always assume that both datasets are at least partially similar.

to find any connection between different sources. Even worse, one neither knows how many source distributions (or hidden influences) are currently influencing the market nor their impact on the stock prices. Also, it is not clear what kind of assumptions about the price generating process should be considered as meaningful. Or put differently: nothing is known about the complex structure of the data generating process. Hence, source component shift can be so severe that a model learned on the training or S data is no longer applicable for the prediction of the testing or P data. Therefore, the source component shift can only be tackled appropriately if the S and P distributions are similar or the complex non-stationarity is so well understood that the transformation of the data generating process can be stated very accurately.

Unfortunately, source component shift is ubiquitous and all the other types of dataset shifts can be considered a special case of this type of shift. Since the definition of this type of shift is so broad one has to consider additional assumptions in order to specify a problem that can be solved. This normally is reflected in either assuming a mild non-stationarity or a known type of transformation within the process. For instance, in the fashion and shoes example, it is rational to assume that both user groups still have something in common because people interested in fashion are very likely to be also interested in beautiful shoes. If no assumption holds, the problem is infeasible as the stock market example shows.

4.5 Summary

The latter chapter gives a brief introduction to different types of dataset shifts in machine learning. The first part considers the traditional machine learning setting where the data is assumed to follow a certain stationary distribution. Subsequently the current abstract idea of what is considered to be a dataset shift in the domain of scientific research is introduced and the differences to the standard setting are pointed out. Several types of dataset shifts are explained while for this thesis the most important ones are covariate shift and source component shift (which also includes the prior probability shift implicitly). In addition to the standard setting the dataset shift setting assumes two (or more) datasets, a primary and secondary dataset(s), that have been sampled from a non-stationary distribution. Due to these distributional differences in the datasets, the prediction task of the new primal data by a model learned on the secondary data becomes harder than in the standard setting. From the example of online marketing it can be shown that these dataset shifts are not just theoretical considerations but are often occurring problems in real life. In fact, as for the standard machine learning setting, considering that, for instance, two datasets are from the same distribution is quite often (a necessary) simplification.

Chapter 5

Transfer Learning

The previous chapter introduced dataset shifts as some kind of difference in distribution between S and P data. This chapter is about *Transfer Learning* that builds on top of these concepts and describes a set of machine learning approaches for compensating these shifts. For point out the difference, dataset shifts describe the mathematical setting while the term transfer learning describes a set of algorithms for accounting for such shifts that can themselves be categorized into different classes of algorithms. Therefore, the first part of the chapter explains why the definition of a new class of algorithms makes sense and which classes of transfer learning algorithms exist. For each class, a set of known algorithms is presented. The second part mainly focuses on how importance sampling and divergence measures can be applied in order to compensate dataset shifts. Such an approach is known as *instance based* approach because it considers each data point or instance individually. Since this work is primarily concerned with instance based approaches this special type of technique is, therefore, discussed in detail. By reading this chapter the reader should become familiar on how to account for dataset shifts by applying different methods and techniques.

5.1 Motivation for transfer learning

Dataset shifts do occur frequently and have a serious impact on the prediction quality. Therefore, they should be considered when inferring a model for the prediction of new data. An intuitive way to tackle the problem could be a regular update of the models such that the model accounts for recent changes in the distributions. This could be done every time after the prediction quality has dropped below a predefined and subjectively chosen threshold. However, it could also be dealt with by taking the information of the shifted data, i.e. the S data, into account when learning the model on the P dataset. As an example, consider the behaviour of internet users concerning a certain online advertising ad where one has a lot of data and a new but related display ad is considered to replace an older one. It is reasonable to assume that the following new ad will attract similar user groups however the behaviour will not be the same as for the old one. Since in the beginning one only has very few data available for the new ad it could be beneficial to incorporate the knowledge of the older ad into the prediction for the new one. Transfer learning is a term that describes methods which try to *transfer the knowledge contained in one dataset A to another dataset B* in order to improve the prediction performance of a new model on the dataset B that is of interest. This is done by modifying existing models or developing new ones. In the past, numerous approaches have been stated for handling such situations that conform the transfer learning setting. Since transfer learning is also highly related to dataset shifts, there exist different methods for each type of shift pursuing different strategies. In fact, a lot of formulations and definitions go hand in hand with the ones for dataset shifts. Therefore, as in the case of dataset shifts, a good practice is to categorize different methods into the type of shift they deal with and type of strategy for treating such shift occurrences.

Before going into detail, the setting of transfer learning should be mathematically formulated. It can be generally described by several given sources or secondary datasets denoted by $\left\{X_i^S, Y_i^S\right\}_{i=1}^N$, with $X_i^S \in \mathbb{R}^{N_i \times D}, Y_i^S \in \mathbb{R}^{N_i \times 1}$, which have been sampled according to some source distributions $p_i^S(x,y)$, $i \in \{1, \dots, N\}$ and a target or primal set $\left\{X^P, Y^P\right\}$ which is a sample from $p^P(x,y)$. Further, the distributions of the datasets are assumed to be not equal that is $p_i^S(x,y) \neq p^P(x,y), \forall i$. The goal of transfer learning is to learn a good model on the primal data $\left\{X^P, Y^P\right\}$ by applying the part of knowledge contained within the sources that improves the prediction of the primary or target data. For simplicity this work will consider just one source dataset $\left\{X^S, Y^S\right\}$ as the general setting is a straight forward extension. Thus in the following, the given data is stated by the source data $\left\{X^S, Y^S\right\} = \left\{(x_1^S, y_1^S), \dots, (x_N^S, y_N^S)\right\}$ and the target data $\left\{X^P, Y^P\right\} = \left\{(x_1^P, y_1^P), \dots, (x_M^P, y_M^P)\right\}$. Additionally in the second part of this work a transfer learning setting is considered that assumes that the amount of given primal data is much smaller than the amount of source data, i.e. $\left|\left\{X^P, Y^P\right\}\right| \ll \left|\left\{X^S, Y^S\right\}\right|$ (where $|.|$ denotes the cardinality of a dataset). It will be shown that in such a situation it will make obvious sense to use additional data. Since the data in the target domain can be considered being sparse, it therefore can be assumed that a model solely learned on the target domain will not perform very well since the information within the target domain will be, in general, also sparse.

5.2 Transfer Learning Problem Classes

There basically exist three types of transfer learning settings. The first one is the Transductive Transfer Learning setting.

5.2.1 Transductive Transfer Learning

In the context of Transductive Transfer Learning [64] the functional relationship between covariates X and the dependent variables Y remains the same but the covariates themselves are shifted. This situation is commonly referred to as Covariate Shift (see section 4.4.1) and can be further categorized into two types. The first one occurs when the space of realm from which the covariate data has been drawn is different. This means that the actual dimensionality of the data can be different or the data itself (the features or attributes) are different. For instance, one dimension is comprised of discrete data while the other one comes from a continuous set. The other type of Transductive Transfer Learning (which is also considered extensively in this work) corresponds to the actual (canonical) Covariate Shift setting in which the covariate data comes from the same space and all dimensions describe the same data but the sampling location is different, i.e. $p^S(x) \neq p^P(x)$. The Sample Selection Bias (see section 4.4.3) scenario is very comparable to the covariate shift setting and algorithms for compensating this kind of shifts also fall in the class of Transductive Transfer Learning algorithms.

5.2.2 Inductive Transfer Learning

The Inductive Transfer Learning [64] setting assumes a source and a target dataset such that $\left|\{X^P, Y^P\}\right| \ll \left|\{X^S, Y^S\}\right|$. Both datasets are assumed to not having been sampled from the same distribution, i.e. $p^S(x, y) \neq p^P(x, y)$. Therefore, this setting also covers the Prior Probability shift, i.e. $p^S(y|x)p(x) \neq p^T(y|x)p(x)$, that is the distribution of the covariates remain the same but the dependent variable changes. In such a situation, one is only interested in the performance of the model prediction quality on the target data while the prediction performance on the source data is irrelevant. Therefore, the source data is exclusively used to improve the prediction of the target data and the evaluation of a trained model will only be performed on the target data. This kind of augmentation is also often referred to as knowledge borrowing [64].

Another specialized Inductive Transfer Learning setting is referred to as *self taught learning* [76] where the source data exhibits no labels but only covariates. In such situations, the source data cannot be applied directly to the target data. Instead, one tries to identify or intensify characteristic features of the target data with the help of the unlabeled data. This way the signal within the data may become clearer.

5.2.3 Unsupervised Transfer Learning

Unsupervised Transfer Learning [64] describes situations in which one has given source and target data that have been sampled from different distributions and additionally do not provide any label information. The datasets are only given by X^S and X^P. The information about the functional relationship is completely missing and cannot be reconstructed. However, it is commonly assumed that the topic (i.e. images, text, etc.) is similar but not necessarily the same [51]. Since this is deeply connected to clustering, unsupervised transfer learning can be understood as the attempt to improve clustering, dimensionality reduction or density estimation [27], [65] and [51]. For instance [27] show a way to learn high quality features in the target domain by applying unlabled auxiliary or source data to the learning approach. The results demonstrate that learning those features exclusively on the target data exhibits a poorer performance than the combination of both datasets. However, although being well defined, so far there is not much work on Unsupervised Transfer Learning available. The reason for that might be that unsupervised transfer learning is a particularly hard problem to solve.

5.2.4 Multi-Task Learning versus Transfer Learning

In contrast to all other types of transfer learning, the Multi-Task learning setting is a little bit different. However, due to its relatedness to transfer learning it is important to mention. Given a set or family of $N \in \mathbb{N}$ different datasets $\{X_n, Y_n\}_{n=1}^N$ Multi-Task learning is concerned with the problem of learning a method that performs well on all given datasets. As all the other transfer learning settings describe situations in which one is only and exclusively interested in improving the prediction performance on the target data, Multi-Task learning also tries to achieve good performance on the source *and* the target datasets simultaneously. Since in this scenario it does not make any further sense to speak about source and target datasets all datasets are denoted as tasks that are treated with equal priority. As a consequence, multi-task learning approaches can be considered as an isolated special topic in the area of machine learning. The topic is well investigated and a huge amount of scientific works have been published already. Important works are for instance [6], [112], [39], [40], [44] and [113]. There exist some connections to the classical transfer learning setting since some ideas could also be applied to transfer learning. However, the assumption of tasks instead of source and target data makes the problem fundamentally different and is a clear criteria for differentiating the two areas. Therefore, although being related to transfer learning, multi-task learning is not a particular topic of or of particular interest for this thesis and is not discussed further.

5.3 Types of Knowledge Transfer

Having seen different types of Transfer Learning settings, the next question is how to actually transfer the knowledge from one dataset to another. This knowledge transfer from one dataset to another can be achieved in different ways which also depend on the type of transfer learning setting under consideration. Therefore, a lot of methods have been stated. Since each method harnesses different possibilities, it is reasonable to categorize them. One category, however, namely the instance based transfer is of special interst for this work and will be discussed in much more detail than the other types.

5.3.1 Feature-Representation-Transfer

This first one is appropriately described by feature-representation-transfer. Such approaches try to identify those dimensions or features of the data that contribute to the dependent variable in the same way while simultaneously trying to avoid features that contribute negatively. Therefore, a common subspace of source and target data is learned such that the information contained in a subspace can be shared between each dataset. One then augments the information of the target data with the additional information from the source data which can be shared in between. As a consequence the model for the prediction of the target data can apply much more data in the learning process than before. The idea of these subspaces comes from the assumption that a common latent structure behind all dataset samples is assumed that is primarily responsible for the data generation process. However, generally the type of latent process is unknown which implies that the focus of feature-representation-transfer lies in trying to learn this shared structure.

An important work on this topic has been done by [6]. Their novel approach is to learn a low dimensional subspace that is shared across all datasets. [59] state a method called DICA (Domain Invariant Component Analysis) that is similar to kernel PCA [85]. The idea is to learn a projection onto a subspace such that features that contribute positively are identified. The data to predict is also projected onto this subspace which enables the augmentation of the target problem with data from the source datasets. The effectiveness of this method has been shown on diverse medical data that exhibits a covariate shift. Another important work treating such settings is given by [67] in which a non-parametric Baysian model is specified that infers the parameters (i.e. the weight vector) for a Logistic Regression model by inferring the parameters by sampling from non-parametric mixture of factor analyzers [36]. This model can be considered as generalized model for a number of other approaches such as [21], [105] and [74].

5.3.2 Hypothesis Transfer Learning

Another category of methods can be summarized by the term parameter-transfer or hypothesis transfer learning. Parameter transfer learning does not apply the source data directly to the model for the target data. Instead, a common model that is specified by a parameter θ^S is inferred on the source data. After that, the source data can be ignored and the new model is solely learned on the target data. However, the regularization for the parameter of new model will then incorporate the θ^S parameter which includes the knowledge from the source data. Mathematically represented, the following steps are taken. First, learn a model θ^S on the source data, i.e.:

$$\theta^S_{\text{opt}} = \text{argmax}_{\theta^S \in \Theta}\ \ell\left(p(x, y|\theta^S), X^S, Y^S\right).$$

After having estimated this parameter the new problem to solve becomes:

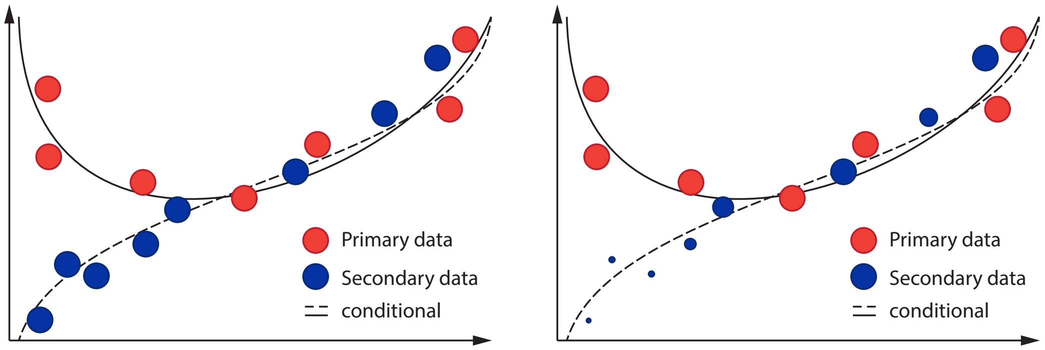
$$\theta^P_{\text{opt}} = \text{argmax}_{\theta^P \in \Theta}\left(\ \ell\left(p(x, y|\theta^P), X^P, Y^P\right) + \lambda \mathcal{R}\left(\theta^S, \theta^P\right)\right).$$

Here $\mathcal{R}(.)$ denotes a special regularization function and λ is the trade off-parameter that describes how much impact the regularization will have on the optimization. A common regularization term is the squared L^2-norm $||\theta^P - \theta^S||^2$ [53], [54]. This prior is equivalent to a Bayesian regression approach where the prior is a gaussian with mean θ^S and variance 1. A loose interpretation of such a regularization would be that the new parameter θ^P should be close (in terms of λ) to the model parameter for the source data. Since the source data is not explicitly required, such an approach can be beneficial in cases where the source data is too large [14] for consideration in computation of the target model. In a Bayesian setting, parameter transfer can be tackled by constructing an informative prior. For instance, [75] try to improve the prediction of text documents by applying auxiliary text documents. [75] achieved this by first learning models with non-informative priors on those auxiliary text documents. These text documents are specialized texts and therefore each source text can be considered as sampled from different distributions that are characteristic for each topic. Therefore the specialized texts contain robust data on the correlation of diverse sub-groups of words, like for instance technical terms. Due to sparseness of the data in the target domain, such a robust estimation will be infeasible when only performed on the target data. The correlations learned on the sources are represented in the parameters for the source models and are then incorporated into the correlation matrix for the target model in the form of an informative prior. The experiments done by [75] show that this approach improves the prediction quality of the target model significantly.

5.3.3 Instance Based Transfer

Instance based transfer learning describes an approach in which each data point or instance from the source and the target domain is considered individually. This is achieved by assigning each distinct data point a factor or weight that controls the degree of influence of that particular data point for the prediction of the target or primal data. The goal of every instance based method is to infer appropriate weight factors for each data point such that data points that contribute positively to the prediction of the primal data get a higher influence for the inference of the prediction model and those which do contribute negatively get less. Usually, a large weight indicates high and a small weight indicates low influence. Figuratively speaking, the idea of this approach is supported by the assumption that for some regions in the source and target datasets, the distributions do match locally or are at least very similar. In that case, it would make sense to consider source data from these regions that match the distribution of the target data in some degree since they exhibit a similar structure and hence reflect the same information. On the other hand, in regions where both datasets differ significantly, it would be desirable to penalize those data points such that their influence on the target model is reduced. The instance-based transfer approach accounts for such requirements.

FIGURE 5.1: The idea of instance based approaches as an illustrative example.



(A) Two functions with similar properties in some regions. Primary and secondary data samples are obtained from each function. For the prediction of the primal function one could apply the secondary data from regions of similarity in order to improve the prediction quality of the P data.

(B) Reweigted data from the example left. Larger secondary datapoints denote heavy weighted data with high importance for or influence on the final prediction while smaller weighted datapoints get less influence. Large points are similar, small ones dissimilar to the primal data.

Figures 5.1*a* and 5.1*b* give illustrative examples for the general idea of instance based approaches. Instance-based approaches are very intuitive and have been investigated in several works. A lot of instance based methods are guided by the idea of importance sampling. The following chapter will discuss the instance based approach in more detail since this work is mainly concerned with the introduction of two new instance based approaches.

5.4 Summary

Transfer Learning describes a problem setting in machine learning that is concerned with the transfer of knowledge or information contained in one dataset to another target dataset in order to improve the prediction on this target dataset. Transfer Learning is also highly related to dataset shifts and actually motivated by them. In the context of machine learning, three major Transfer Learning types have been established, the Transductive, Inductive and Unsupervised Transfer Learning settings. Each of these settings is different to the related Multi-Task learning setting and for each problem type, different methods have been developed that can be categorized into Feature Representation Transfer, Hypothesis Transfer, and Instance-based Transfer. The next chapter will extensively discuss the instance based transfer learning setting.

Chapter 6

Importance Sampling for Instance Based Transfer Learning

As explained in the previous two chapters, instance based transfer learning methods require the determination of individual weights for each datapoint. These weights indicate how important a datapoint is for the calculation of the final model for prediction. As has been shown in chapter 2, importance sampling is a method to state an accurate transformation function, i.e. the importance function, for transforming between two distributions. This transformation function can also be applied to a single individual point such that it expresses the importance of that point. Therefore, it seems straight forward to apply this method for the determination of individual weight for instance based transfer learning setting. The following chapter discusses this idea, will give an overview of existing methods and, further, will state modified learning methods for the application of those importance weights.

6.1 Importance Sampling for Distribution Matching

Practical settings that comprise shifted data arise from datasets that have been sampled from different distributions, p^P and p^S. As a consequence, a model learned on one dataset might not be well suited for the prediction of another dataset. As already pointed out instance-based methods provide one way to compensate for this shift by assigning individual weights to each data point. Importance sampling is one possibility to obtain such data point weights. The weights are calculated from the importance function and are then applied to an (often) modified model for the target data such that it can better fit the target dataset. In fact, as a result from section 2.1.3 this way one would obtain perfect weights, because these weights then come from the real analytical importance function which yields the exact actual ratio. This is possible in any analytical framework where the distributions of the sampled data are all known. Then, the estimation of the importance function is relatively easy because it can be exactly and straightforwardly calculated: $w(x) = {}^{p^P(x)}/_{p^S(x)}$. Thus, the optimal transformation would actually be the ratio between the two distributions, because then the transformed distribution $w(x)p^S(x)$ exactly matches the target or primal distribution $p^P(x)$. As a result, due to that *distribution match*, a sampling from the secondary distribution can be used as if it was a sample of the primal distribution, although is has been sampled differently. However, in general, these distributions p^S and p^P are completely unknown and the exact calculation becomes impossible. Instead, the only information about the distributions one can get is reflected within the given datasets for which it is assumed that they are sampled from these distributions. *Distribution Matching* [70] in the context of machine learning is a term that describes the attempt to obtain a transformation function $\hat{w}(x)$ for distribution $p^S(x)$ to arrive at the target distribution $p^P(x)$ based only on the given empirical datasets. It is basically a term that denotes methods that try to find an approximation of the real importance function. A very simple and straight forward approach in solving this distribution matching problem is the application of empirical density estimators. The field of *Density Estimation* [86] provides a variety of methods for the approximation of densities from given sample data. The idea is to apply two separate density estimations to each dataset to obtain density functions for p^P and p^S. Afterwards, these approximated densities are taken for the calculation of the importance function at each new datapoint of interest. However, although being very intuitive and simple, this approach has some drawbacks. For instance, it is not clear which type of density estimation method one should apply to the given datasets. Therefore, choosing an inappropriate method can lead to severe errors due to a false model assumptions for the density. Another problem is that the curse of dimensionality leads virtually always to the empty space phenomenon. As a result, only very scarce data is available for the density estimation in high dimensional spaces. This can very easily lead to highly incorrect model estimations of the density. Thus, approaches employing a two step procedure of estimating empirical densities and using those for the calculation of the importance weights are not preferable for the *Distribution Matching* problem setting.

Instead, one should skip the step of empirical density estimation and state a direct model for the importance function. [100] propose a method that approximates the solution of the integral equation $\int_{-\infty}^{x} dp^P(y) = \int_{-\infty}^{x} w(y)p^S(y)$ in a least squares setting. Another common approach for the direct approximation of the importance function of two given distributions p^S and p^P is a linear combination of Gaussian kernel functions as for instance given by [91]:

$$\frac{p^P(x)}{p^S(x)} = w(x) \approx \hat{w}(x) := \sum_{i=1}^{N} \alpha_i e^{-\frac{||x-\zeta_i||^2}{2\sigma^2}}. \tag{6.1}$$

Occasionally, the notation $\hat{w}^\alpha$ will be used in order to emphasize the dependence on the α coefficients. It is also possible to apply other approximation methods like linear or more general polynomial functions instead of the Gaussian functions [41]. However, Gaussian kernels are often applied in machine learning settings and provide a very flexible framework. For very small values of the bandwidth parameter σ one can construct very non-linear function shapes as well as linear shapes for large values of σ. Additionally, each α_i coefficient regulates the amount of influence of the corresponding kernel function for the whole approximation. While on the one hand the α coefficients and the σ bandwidth parameter can be regarded as parameters that determine the approximation the values for ζ_i's on the other hand have to be chosen carefully. The ζ_i's describe the centers or maximums of each of the N Gaussian functions. Therefore the ζ_i's will be referred to as center points for the approximation. The choice of center points is a crucial part for the accuracy of the approximation and it depends mostly on the problem at hand. In general, the center points should be set to those points for which one would like to measure the similarity. This can be the whole primary or secondary data, a subset of those datasets or even completely different data which might have been selected due to some kind of pre-knowledge about the current problem. Since the choice depends on the situation, the definition of the center points is discussed for each problem separately in the upcoming chapters. For a general discussion of the model, the specific structure of the center points is less important such that for the sake of simplicity it is now assumed that the center points have been chosen appropriately. Expression (6.1) gives a model which enables the calculation of individual weights simply by plugging in a location x of interest. Thus, one can determine the weights for each secondary data point from the set X^S very easily by considering an individual point $\hat{w}(x^S)$.

Remark: An alternative notation that explicitly contains the labels for expression (6.1) is:

$$\frac{p^P(x,y)}{p^S(x,y)} = w(x,y) \approx \hat{w}(x,y) := \sum_{i=1}^{N} \alpha_i e^{-\frac{||(x,y)-\zeta_i||^2}{2\sigma^2}}.$$

This expression also implies that the ζ_i's also include the label information implicitly. The latter notation will be used in cases for which it will be convenient to explicitly state or emphasize that the labels are also considered in the process of distribution matching.

6.2 Caveats of the approximation $\hat{w}$

The approximation $\hat{w}$ (6.1) of the importance function w depends on several entities that influence the quality of the approximation. These entities are the data samples X or (X, Y), respectively, the center points ζ_i and the parameter σ. This section will explain intuitively how the approximation depends on those. Since $\hat{w}$ applies a linear combination of Gauss kernels (a.k.a. radial basis functions (RBF) in the machine learning context) the reader should also refer to [77], [13] and [84] for more information about their properties.

The first entity are the given data points $X \subset \mathbb{R}^{M \times D}$ or when the labels are explicitly given $(X, Y) \subset \mathbb{R}^{M \times D+1}$. Normally, the data X are measurements that have been recorded at some time and should therefore be considered as just given. Thus the data can be considered as being arbitrary but fixed which implies that the fit of approximation cannot be improved by obtaining additional data. This can be especially a problem when only very few measurements are given because then, this limited amount of data prevents the ability of estimating a good fit of $\hat{w}$ due to the lack of information about the structure of the problem. The best situation for compensating this would be "all" possible data since then the actual distribution $p(x)$, which encodes the complete characteristics of the data, could be estimated. But not only is insufficient amount of data available, but also the properties of p naturally imply a different amount of sampled data at different regions. That means for instance that data is relatively dense in some regions of high sampling probability while being scarce at other regions of lower sampling probability. Thus, if one has given a limited sample of data points X the unknown distribution is most likely reflected very poorly or unclearly by this data X - at least in regions of low sampling probability. Therefore, the quality of fit will deviate depending on the amount and hence the quality of the data X.

The second crucial entity in the equation are the center points $\zeta \subset \mathbb{R}^{N \times D}$. These data points are assumed to be a sample from a second distribution $q(x)$. Essentially, the same issues that hold for data X also hold true for the ζ center points. I.e. the amount of given center points determines the quality of fit of the importance function. However, additionally, since the importance function $w(x)$ depends on both quantities p and q it is necessary to investigate the way both datasets X and ζ interact in terms of $\hat{w}$. Figure 6.1 provides an illustrative example of the problems that can occur. To better understand the interaction of both datasets, one should start by considering the worst case possible. A worst case scenario is when one cannot infer anything at all from some given data samples. The following example describes a situation when it is impossible to learn anything about $\hat{w}$. Considering two distributions p for X and q for ζ. Both distributions p and q are assumed to be normally distributed (figure 6.1) with an equal

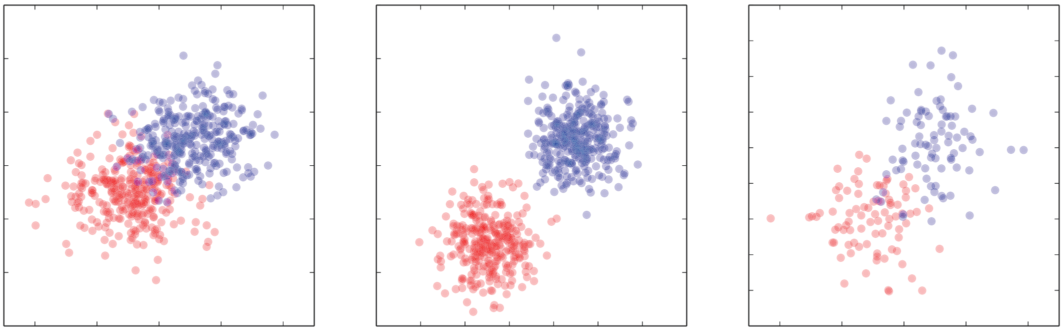


FIGURE 6.1: Left plot: Two sufficiently large dataset samples that are dense and overlapped. This provides an ideal situation for inferring an importance function approximation. Middle plot: No overlap implies that virtually nothing can be learned about the importance function of the datasets (worst situation). Right plot: Both datasets are overlapped, however, the overlap is sparse which can make inferences very difficult.

and fixed variance $\eta \in \mathbb{R}$ but two (different) means, μ_1 for p and μ_2 for q. Now, from both distributions, two data samples are taken denoted, by $X \sim p$ and $\zeta \sim q$. Given an equal variance for both distributions, the similarity/closeness or overlap of the data points X and ζ depends on the distance between μ_1 and μ_2. By having a large distance in between, this means that the actual overlap of both datasets might be too small or even worse the datasets might be disjoint (figure 6.1 middle plot). In the case of two disjoint data samples there is virtually nothing one can learn about. Such a situation can be considered as too sparse in order to learn a good representation for the importance function w. Therefore, it is assumed that both datasets must have at least partially a sufficiently large overlap in order to determine properties of w. On the other hand, the more data lying in the overlap, the better the representation $\hat{w}$ of w can be learnt (figure 6.1 left plot). Another problem can arise when an overlap is present but the data within this overlap is too sparse (figure 6.1 right plot). This situation makes the approximation basically possible but prone to perturbed data. The consequence is that methods of instance based approaches should also take robustness into account. Hence the actual second crucial issue is the degree and density of overlap between both datasets. Since this cannot be controlled, it must be assumed that at least some overlap exists.

The third entity that crucially influences the approximation quality depends on the bandwidth parameter σ. The possible values for σ depend directly on the number of samples of both datasets X and ζ and its corresponding degree of overlap. To better see this, it is now assumed that the two dataset samples X and ζ are considered as being fixed. In that case, the number or density of samples have great implications on the magnitude of the (controllable) parameter σ. These implications are visualized in figure 6.2 where the influence of the bandwidth parameter on the fit is shown. In order to make the figure better interpretable, each resulting function has been normalized to $max(\hat{w}(x)) = 1$. This is achieved by setting $\alpha_i = {}^{1}/_{max(\hat{w})}\ \forall i \in \{1, \ldots, M\}, x \in X$.

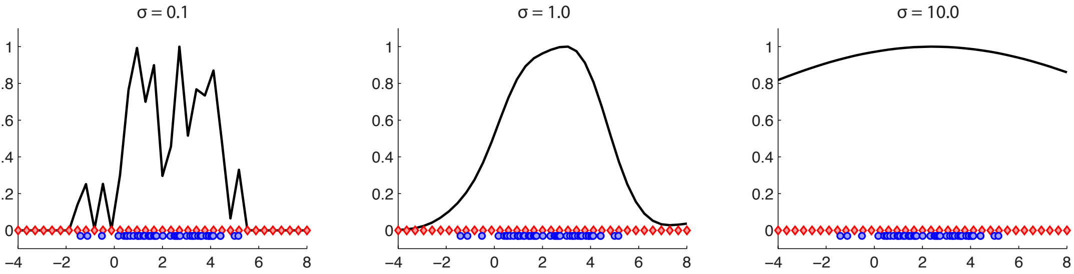


Figure 6.2: The term density of the data samples depends on the magnitude of the bandwidth parameter σ. If σ is chosen small the density of the data sample can be interpreted as being too small. On the other hand an increased value of σ leads to a generalization which might be too broad. In this particular example the value $\sigma = 1$ will yield a very good result. However, this is due to the fact that the number of data points in this particular example can already be considered as a very dense sample. In higher dimensional spaces this will most probably not continue to hold due to the empty space phenomenon. Additionally, it should be noted that the center points are a sample from a normal distribution which has a relatively simple structure and thus is relatively easy to approximate.

Further, the X data in this example is an equidistant "sample" set since it makes the approximation less dependent on the sparseness of X. Under these circumstances, if, for instance, the bandwidth parameter is chosen small relative to the amount of available data, the resulting approximation will expose a high variance. Thus, the approximation of the importance function in between sampled data points can be regarded as an overfit and will be very unreliable or, even worse, will output wrong results. On the other hand, increasing the value of σ will soften this volatility. However, increasing σ can lead to a too large bandwidth which might yield an approximated importance function that generalizes too strongly. Figuratively speaking, the function becomes to broad such that the weights are all similar and no longer individual enough. This effect becomes especially problematic for higher dimensional data where due to the curse of dimensionality, the sampled dataset will virtually always be very sparse. Also the degree of overlap has implication on the choice of σ. If for instance no overlap exists, the approximation $\hat{w}$ (6.1) has to be broadened in order to capture the interaction of both dataset samples on the edges. If σ is chosen too small the weights in the overlap are also small which leads to a situation where all additional data get 0 weights and hence can not be applied. However, this can partially be a desired property since it indicates that the additional data is too dissimilar to the primary data and will not be applied. Therefore, dependent on the number of data samples X and ζ the bandwidth σ has to be chosen smaller or greater which makes the approximation of function $\hat{w}$ (6.1) dependent of the density of data samples. The more data is given the smaller σ can be and the more characteristic properties of the importance function can be approximated without risking a too volatile function fit.

However, in some situations it is necessary to apply a small σ which is potentially prone to result in an overfit. This is especially true for higher dimensional problems when the given dataset is sparse. For example in a 10 dimensional space a sample of 1000 data points is very often already a very sparse dataset. Sparse data often leads to a too large choice for the bandwidth σ. The consequence is that this leads to a too broad approximation which in turn yields a too general weight function that assigns each additional secondary datapoint the same weight. This means that each secondary datapoint is equally important to take into account. However, if a dataset shift is present this cannot be the case because then the two datasets are at least partially different which implies different weights. Therefore, it can happen that one has to accept a certain amount of overfit and consequently a volatile approximation. Since this is a problem, the new algorithm presented in chapter 7 was designed with this aspect in mind and improves the approximation of $\hat{w}$ in such cases.

Finally, a fourth entity has also to be considered which cannot be controlled. So far it was assumed that the data is a "clean" sample from both distributions. This actually does not match the circumstances in practical settings. Normally the measurements have been exposed to some random influences such that the data sample is practically always perturbed or noisy. These perturbations are commonly considered to be the result of various unknown and overlaid hidden sources of influence. Due to that, they are commonly modeled as noise in form of a Gaussian distribution. However, since these perturbations have a direct influence on the quality of the dataset samples, it is even more difficult to estimate an appropriate $\hat{w}$ function. This holds particularly true for situations of sparse data since then the effect of an individual distorted data point has a higher impact relative to the number of all data points. As a consequence, in order to make the approximation smoother, there is a further tendency to estimate a larger σ than the one that is actually required.

From the above, one can see that although approximation $\hat{w}$ (6.1) is very simple, there are a lot of problems that might arise. Some of these, like the data itself, cannot be controlled while other caveats (especially in higher dimensional spaces) can be thought of when inferring the approximation $\hat{w}$ of the importance function. This reduces the instance based approach to the estimation of an appropriate importance function approximation $\hat{w}$ which captures the given structure well by simultaneously not being too volatile. To account for this, the thesis will concentrate on the third and last aspect of the discussed topics, i.e. the influence of the parameters of the importance function approximation and the perturbations of the given data. Throughout the rest of this work, the assumptions for the data are that they are sufficiently dense and provide at least some overlap. Such assumptions are required because normally one does not have a lot of influence on the structure of the given data itself. The next question to ask would be the inference of an appropriate $\hat{w}$ under these assumptions.

6.3 Divergence Measures for Distribution Matching

Until now no method for estimating the parameters of expression (6.1) is stated. Since this work will apply importance sampling for the estimation of appropriate weights a reasonable idea is the application of divergence measures (as discussed in chapter 2). The reason for its application is that divergence measures are motivated by probability functions, distributions and at its core by importance sampling and can therefore be considered a natural approach. Thus divergence measures are a good choice because they offer a mathematical framework for estimating appropriate parameters. Further, by providing these capabilities they consequently serve well the needs required in distribution matching such that one arrives at the distribution of interest. Assuming p^P is the target or primal distribution and p^S the source distribution, then mathematically presenting, one is looking for something like:

$$p^P(x) = w(x)p^S(x) \Leftrightarrow \frac{p^P(x)}{p^S(x)} = w(x) \approx \hat{w}(x).$$

Here, $\hat{w}(x)$ denotes the approximation or model of the importance function given in (6.1). Based on the model (6.1) for the reweighting function w it becomes possible to apply divergence measures to obtain an optimization problem in order to find appropriate α and σ parameters. The abstract formulation for this problem is given by:

$$\min_{w} D(p^P||wp^S) \approx \min_{\hat{w}} D(p^P||\hat{w}p^S).$$

This expression suggests that this approach is straight forward because it is formulated using a general definition of a divergence. However, employing any divergence measure is only possible in principal. In a practical settings, in which one lacks of the knowledge of the distributions, this will not always be possible because one will encounter different problems on an analytical level. The biggest problem one has to care for is that these two quantities, p^P and p^S, vanish somehow from the optimization problem. This is necessary because these two functions are unknown and therefore cannot be employed. Not every divergence measure can (in an analytical way) be transformed in such a way that the optimization no longer depends on the distributions p^P and p^S. Yet, some divergence measures are well suited for this task. An important example of a divergence for which this analytical transformation is possible is the Kullback-Leibler divergence:

$$\min_{\hat{w}} D_{KL}(p^P||\hat{w}p^S) = \min_{\hat{w}} \int_{-\infty}^{\infty} \log\left(\frac{p^P(x)}{\hat{w}(x)p^S(x)}\right) p^P(x)dx.$$

Using this divergence one can get rid of the explicit distributions by applying the following calculations:

$$\arg\min_{\hat{w}} \int_{-\infty}^{\infty} \log\left(\frac{p^P(x)}{\hat{w}(x)p^S(x)}\right) p^P(x)dx \tag{6.2}$$

$$= \arg\min_{\hat{w}} \left(\int_{-\infty}^{\infty} \log\left(p^P(x)\right) p^P(x)dx - \int_{-\infty}^{\infty} \log\left(\hat{w}(x)\right) p^P(x)dx - \int_{-\infty}^{\infty} \log\left(p^S(x)\right) p^P(x)dx\right) \tag{6.3}$$

$$= \arg\min_{\hat{w}} \int_{-\infty}^{\infty} \log\left(\hat{w}(x)\right) p^P(x)dx \approx \arg\min_{\hat{w}} \sum_{i=1}^{M} \log\left(\hat{w}(x_i^P)\right) \text{ with } x_i^P \sim p^P \tag{6.4}$$

Since p^P and p^S do not contribute to the optimization of the problem the first and third term in the second line can be regarded as constant and can therefore be ignored. In the last line the empirical mean has been applied on expression $\int_{-\infty}^{\infty} \log\left(\hat{w}(x)\right) p^P(x)dx$ which denotes an expectation with respect to p^P. Therefore, the data points $x_1, \ldots, x_M$ in the empirical estimate are samples from the distribution p^P. For large M this approximation becomes very accurate.

6.4 Instance Based Approaches for Compensating Covaritate Shift

The covariate shift problem setting has been introduced in section 4.4.1 where it was defined as $p^P(x, y) \neq p^S(x, y)$ caused by different marginal or covariate distributions $p^P(x) \neq p^P(x)$ but equal conditionals i.e.: $p^P(y|x) = p^S(y|x)$. As section 5.2.1 points out, the machine learning methods for compensating such a shift are often referred to as transductive transfer learning. This section will give an introduction on how instance based methods can be applied for compensating such shifts. Chapter 7 will present new findings based on similar methods which better compensate such a shift.

Covariate shift situations occur for instance if a non stationary process drifts over time such that the covariates change. In the experimental sections of the next chapter 7 an earth quake dataset is applied where the measurements are taken at different locations such that the actual (geographical) location may have an impact. To rectify the covariate shift problem, one can put more weight on secondary/training data points that lie close to the primal data $X^P \subset \mathbb{R}^{D \times M}$, assuming that these better represent the structure of the primal data. In supervised learning, such a situation where, besides the secondary data, additional samples are available for whose only the locations x are given is known as semi-supervised learning [20]. However, the difference is that semi-supervised learning does not assume a shift of any kind within the given data and is therefore more comparable to the classical machine learning setting.

For recent surveys on the state of the art on covariate shift as well as the more general dataset shift see [57, 71, 89]. In this survey, an important method discussed is the so called Kullback-Leibler Importance Estimation Procedure (a.k.a. KLIEP) [91]. The derivation of KLIEP is basically shown in (6.2) of the previous section. It benefits from the property of the KL divergence where the probabilities p^P and p^S vanish by approximating the KL divergence by taking empirical expectations. By further applying the approximation (6.1) of the importance function, as shown in (6.2), KLIEP determines importance weights that can then be applied to compensate the covariate shift.

Another application of a divergence measure was investigated in [47]. In their work they consider the euclidian distance as a divergence measure which is a divergence of the class of Bregman divergences. The idea is to consider the distance between the real unknown importance function w and it's approximation $\hat{w}$ (6.1):

$$\begin{aligned}
&\arg\min_{\hat{w}} \frac{1}{2}\int_{-\infty}^{\infty} (\hat{w}(x) - w(x))^2 \, p^P(x)dx \\
=&\arg\min_{\hat{w}} \frac{1}{2}\int_{-\infty}^{\infty} \hat{w}(x)^2 p^P(x)dx - \int_{-\infty}^{\infty} \hat{w}(x)w(x)p^P(x)dx + \frac{1}{2}\int_{-\infty}^{\infty} w(x)^2 p^P(x)dx \\
=&\arg\min_{\hat{w}} \frac{1}{2}\int_{-\infty}^{\infty} \hat{w}(x)^2 p^P(x)dx - \int_{-\infty}^{\infty} \hat{w}(x)w(x)p^P(x)dx \\
\approx&\arg\min_{\hat{w}} \frac{1}{2}\sum_{i=1}^{M} \left(\hat{w}(x_i)\left(\hat{w}(x_i) - 1\right)\right)
\end{aligned}$$

Similar to the KLIEP approach, this method applies the approximation of the integral by the empirical mean such that the probability functions vanish from the calculation. They call this method uLSIF (Unconstraint Least Squeares Importance Fitting). As in the case of KLIEP it is formulated for and applied to the covariate shift problem.

A further widely known method for instance based transfer is the so called Kernel Mean Matching algorithm (KMM) [45]. The idea of KMM is to find a match between data that has been mapped into some feature space by applying a kernel map and subsequently minimizing the distance in between these two mappings. The mathematical problem formulation can be given as:

$$\left\|\frac{1}{N}\sum_{j=1}^{N} \beta_j \phi(x_j^S) - \frac{1}{M}\sum_{i=1}^{M} \phi(x_i^P)\right\|^2 = \frac{1}{N^2}\beta^t \mathcal{K} \beta - \frac{2}{N^2}\kappa^t \beta + \text{const.}$$

Here, β^t denotes the transpose of the parameter of coefficient vector β, ϕ denotes the feature mapping, $\mathcal{K}$ the implied empirical kernel matrix, N, M the number of source and target data, and κ is the similarity measured by the kernel mapping function $\Bbbk(x, y)$, i.e. $\kappa_j = \frac{N}{M}\sum_{i=1}^{M} \Bbbk(x_i^P, x_j^S)$ for x_j^S coming from p^S and x_i^P coming from p^P. The final optimization problem is:

$$\min_{\beta} \frac{1}{2}\beta^t \mathcal{K}\beta - \kappa^t \beta \qquad \text{s.t.} \quad \beta_j \in [0, B] \text{ and } \left|\sum_{j=1}^{N} \beta_j - N\right| \leq N\epsilon.$$

The first constraint $\beta_j \in [0, B]$ bounds each individual value of β and the second constraint ensures that the sum of the β values sums up to 1 in expectation. A major drawback of KMM is that the authors do not provide a possibility for adjusting the parameters. For instance, when applying a Gaussian kernel function, there is no known method for adjusting the bandwidth parameter η. Therefore, one has to guess the parameter which clearly is a huge disadvantage, since the practical performance differs significantly on the chosen parameter.

6.5 Regression and Classification Methods for Covaritate Shift Compensation

The previous section introduced some methods for estimating individual weights for each datapoint. In particular, a weight for each S datapoint can be calculated such that each of these S datapoints gets assigned a certain amount of importance. These weights then determine how much importance or influence a datapoint gets for compensating the shift. High values denote important data points, whereas low values stand for less important data points. Classification and regression methods need to incorporate this information such that the prediction in regions of heavily weighted S data points 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 S data point. The derivation of these modified methods is presented in this section.

6.5.1 Weighted Support Vector Regression (WSVR)

For regression problems a modified version of a support vector regression (SVR) problem is given here by:

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

Here θ and b denote the model parameters and the newly introduced $\hat{w}(x_n)$ are the estimated importance weights. For each data point, 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 data points with large weights will tend to be lower than those multiplied by small weights, thus causing a lower tolerance to errors on important data points. In order to apply the kernel trick [84] it is necessary to dualize the optimization problem. The Lagrange function is given by:

$$\begin{aligned}
L(\theta, b, \xi, \xi^*, a, a^*, \beta, \beta^*) =& \frac{1}{2}||\theta||^2 + C\sum_{i=1}^{N} w_i(\xi_i + \xi_i^*) - \sum_{i=1}^{N}\beta_i\xi_i - \sum_{i=1}^{N}\beta_i^*\xi_i^* \\
& - \sum_{i=1}^{N} a_i\left(\epsilon + \xi_i - y_i + \theta^t x_i + b\right) - \sum_{i=1}^{N} a_i^*\left(\epsilon + \xi_i^* - \theta^t x_i - b + y_i\right)
\end{aligned}$$

For the elimination of the primal variable one needs to derive the Lagrange function w.r.t. each primal variable:

$$\begin{aligned}
\frac{\partial}{\partial\theta}L &= \theta - \sum_{i=1}^{N}(a_i + a_i^*)x_i = 0 \\
\frac{\partial}{\partial b}L &= \sum_{i=1}^{N}(a_i^* - a_i) = 0 \\
\frac{\partial}{\partial\xi_i^{(*)}}L &= w_i C - a_i^{(*)} - \beta_i^{(*)} = 0
\end{aligned}$$

Here, as for $\beta^{(*)}$, $a^{(*)}$ denotes a as well as a^*. Transforming the first equation to θ and the last to $\beta^{(*)}$ one can substitute θ and $\beta^{(*)}$ in the primal problem. Finally, the dual version of the problem becomes:

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

where dot product $x_i^t x_j$ has been replaced by $\mathcal{K}$ (the kernel trick). As for the weighted support vector machine, $\mathcal{K}$ denotes the resulting empirical kernel map. In the following Gaussian kernels will be applied.

6.5.2 Weighted Support Vector Machine (WSVM)

Analogously, a modified version of a weighted support vector machine for classification is given in [45] by the following optimization problem:

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

where θ describes the separation hyperplane and b its offset from 0. ξ are the slack variables. The dual formulation is also obtained in a similar way as the WSVR. The Lagrange function for the WSVM is:

$$L(\theta, b, \xi, a, \beta) = \frac{1}{2}\|\theta\|^2 + C\sum_{n=1}^{N}\hat{w}(x_n)\xi_n - \sum_{n=1}^{N} a_n\left(\xi_n - 1 - y_n\left(\theta^t\phi(x_n) - b\right)\right) - \sum_{n=1}^{N}\beta_n\xi_n$$

From that Lagrange function, similary as for the WSVR, the dual formulation of the optimization problem is given by:

$$\begin{aligned}
\max_{a} \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n,m=1}^{N} a_n a_m y_n y_m \mathcal{K} \\
\text{subject to: } \sum_{n=1}^{N} a_n = 0 \\
0 \leq a_n \leq \hat{w}(x_n)C \quad \forall n = 1 \ldots N
\end{aligned}$$

where a are the dual variables and $\mathcal{K}$ denotes the empirical kernel matrix.

6.6 Instance Based Approaches for Inductive Transfer Learning

The inductive transfer learning (ITL) problem has been presented in section 5.2.2. In contrast to the covariate shift setting, the ITL setting further assumes an additional shift also in the dependent variables y. Formally speaking, inductive transfer learning (ITL) refers to the situation with at least two datasets $(X^P, Y^P) \subset \mathbb{R}^{M \times D+1}$ and $(X^S, Y^S) \subset \mathbb{R}^{N \times D+1}$, which are sampled from the distributions $p^P(x, y)$ and $p^S(x, y)$, while in general $p^P(x, y) \neq p^S(x, y)$. The kind of underlying dataset shift is called source component shift and is explained in detail in section 4.4.5. Although the main assumption is $p^P \neq p^S$, it is also assumed that the distribution p^P and p^S are at least somewhat similar, and that for some connected sets of (x, y) one has $p^P(x, y) \approx p^S(x, y)$. Otherwise it is not possible to transfer anything from the S data to the P data. Additionally, it is assumed that the number of P data is much smaller than that of the S data, i.e. $|X^P| \ll |X^S|$, where $| \cdot |$ is the cardinality of a set. Therefore, a model learned solely on the P data will not provide a very good prediction quality due to the small number of data. As a consequence, the goal of ITL is to improve the prediction model and hence the quality of a model for the P data by employing data from the S data. In contrast to multi-task learning [6], inductive transfer learning is not concerned with the prediction quality on both the S and P data, but concentrates only on the prediction of the P data; the S data is exclusively used as data that helps to improve the prediction quality of the P data.

ITL situations often occur when the distribution drifts over time. An example for such a shift is given in the experimental section in Chapter 8, where a dataset is applied that describes the causes of delays of aircrafts over some years. Over the years, the delays and measurements vary which is due to a shift of the underlying data generating process. The cause for a changing generating process might be due to new airports that have been opened recently or new aircraft models that are more reliable. Therefore, the data can shift from year to year. Other examples are the classification of text data where one would like to transfer knowledge learned on texts about a certain topic to texts about a different topic [75].

As for the covariate shift, the ITL setting can also be dealt with instance based methods. Since, by assumption, it partially holds that $p^P(x, y) \approx p^S(x, y)$ the instance based methods should pick these similar datapoints from the S data and assign them a high importance. Ideally, the dissimilar points get a low importance and hence a low influence. TrAdaBoost [26] and an extension [3] are instance based methods that assign a weight to each data point such that some S data have an influence on the prediction quality for the P data. [66] states a similar boosting approach for regression. Other recent work based on instance transfer has been put forward by [92], [108] and [110] in which they use multiple input sources to improve the prediction quality of classifiers.

Other important existing works that implement instance based reweighting methods have been discussed in the previous section 6.4 and focus primarily on the covariate shift setting. For comparison reasons, it is interesting to apply these methods also to the inductive transfer learning setting. However, the major drawback of these methods in the ITL setting is that they do not take the information about the target labels from the P data into account. This can lead to situations where a S data point still gets a high weight assigned due to the similarity to the covariates of the P data although the label, which eventually is what one wants, is fundamentally different from the ones in the P data.

In chapter 8 two new instance based methods are presented and derived. The first one, which is called Direct ITL or DITL, can be considered being a supervised method and is a completely new approach. The second approach (explained in section 8.3.3) is inspired by KLIEP [91]. However, to compensate the shortcoming of the covariate shift assumption of KLIEP the second approach is modified such that it takes also the labels of the P data into account. The second approach will be refered to as Kullback-Leibler ITL or KLITL.

6.6.1 Weighted Kernel Ridge Regression for Compensating Source Component Shift

The application of the learned weights requires an adjusted model for the prediction. Therefore, a new weighted kernel ridge regression model is proposed now, which will be referred to as ITL-KRR in the following. The modified ridge regression model is given by:

$$J_W(\theta) = \frac{1}{2}\left(\sum_{i=1}^{M}(\theta^t\phi(x_i^P) - y_i^P)^2 + \sum_{j=1}^{N} w_j(\theta^t\phi(x_j^S) - y_j^S)^2\right) + \frac{\lambda}{2}\sum_{d=1}^{D}\theta_d^2 \tag{6.5}$$

where $\theta \in \mathbb{R}^D$ denotes the model parameter, λ the regularization parameter, ϕ the feature map that maps the input x into the feature space (see e.g. [13]), and $w_j := \hat{w}(x_j^S, y_j^S)$ denotes the weight for each supplementary data point from $\left(X^S, Y^S\right)$. This expression can be written in a more compact way. To get there, each term in the first sum will be multiplied by a neutral 1. Each 1 will become a weight factor for a corresponding data point from X^P. Secondly, the data points X^P and X^S, as well as the labels Y^P

and Y^S, will be concatenated to:

$$\begin{aligned}
X^{PS} &= \left(X^P|X^S\right) \in \mathbb{R}^{M+N\times D}\\
Y^{PS} &= \left(Y^P|Y^S\right) \in \mathbb{R}^{M+N\times 1}\\
\text{and } (\tilde{w}_1,\dots,&\tilde{w}_M,\tilde{w}_{M+1},\dots,\tilde{w}_{M+N}) = (\underbrace{1,\dots,1}_{M \text{ elements}},w_1,\dots,w_N).
\end{aligned}$$

Equipped with these new notations, the optimization problem can be rewritten as:

$$J_{\tilde{W}}(\theta) =\frac{1}{2}\left(\sum_{i=1}^{M+N}\tilde{w}_i(\theta^t\phi(x_i^{PS})-y_i^{PS})^2\right)+\frac{\lambda}{2}\sum_{d=1}^{D}\theta_d^2.$$

In order to be able to apply kernels to this method, it is necessary to dualize the latter expression. The derivative w.r.t. the primal parameter vector θ is:

$$\begin{aligned}
\nabla_\theta J_{\tilde{W}}(\theta) &= 0\\
\theta &= \sum_{i=1}^{M+N}\underbrace{\left(-\frac{1}{\lambda}\tilde{w}_i\left(\theta^t\phi(x_i^{PS})-y_i^{PS}\right)\right)}_{=:a_i}\phi(x_i^{PS})\\
\theta &= \Phi(X^{PS})a =: \Phi a
\end{aligned} \tag{6.6}$$

Substitution of θ by Φa yields:

$$\begin{aligned}
J_{\tilde{W}}\left(\Phi a\right) =&\frac{1}{2}\sum_{i=1}^{M+N}\tilde{w}_i\left((\Phi a)^t\phi(x_i^{PS})-y_i^{PS}\right)^2+\frac{\lambda}{2}\left(\Phi a\right)^2\\
=&\frac{1}{2}a^t\Phi^t\sum_{i=1}^{M+N}\phi(x_i^{PS})\tilde{w}_i\phi(x_i^{PS})^t\Phi a - a^t\Phi^t\sum_{i=1}^{M+N}\phi(x_i^{PS})\tilde{w}_i y_i^{PS}\\
&+\frac{1}{2}\sum_{i=1}^{M+N}\tilde{w}_i y_i^{PS2}+\frac{\lambda}{2}a^t\Phi^t\Phi a
\end{aligned}$$

The last expression can be written in matrix notation. Therefore, it is necessary to introduce the following new notations:

$$W := \begin{bmatrix} I_M & 0\\ 0 & \operatorname{diag}\left(w(x_1^S,y_1^S),\dots,w(x_N^S,y_N^S)\right)\end{bmatrix}$$

where I_M denotes the identity matrix of size $M\times M$. Additionally, the kernel trick [84] is considered, i.e.:

$$\mathcal{K} = \Phi(X^{PS})^t\Phi(X^{PS}) = \Phi^t\Phi.$$

As the dual optimization problem, one then gets:

$$\frac{1}{2}a^t\mathcal{K}W\mathcal{K}a - a^t\mathcal{K}WY^{PS} + \frac{1}{2}Y^{PS}WY^{PS} + \frac{\lambda}{2}a^t\mathcal{K}a.$$

Predictions for new data x^* can be obtained by considering the dual version of primal function for prediction:

$$\hat{f}(x^*) = \theta^t\phi(x^*) = (\Phi a)^t\,\phi(x^*) = a^t\Phi^t\phi(x^*) = a^t\Bbbk(x^*) \tag{6.7}$$

where $\hat{f}$ denotes the prediction function and the model parameter θ has to be substituted by the expression Φa (6.6). The kernel trick [84] has been applied on the expression $\Phi^t\phi(x^*)$. Thus, the expression $\Bbbk(x^*)$ is given by $(\Bbbk(x_1, x^*), \ldots, \Bbbk(x_L, x^*))^t$, i.e. the kernel map of the new datapoint x^* and the data X^{PS} on which the model is learned.

6.7 Summary

Instance based approaches for tackling the transfer learning problem setting are an adequate way for compensating diverse data set shifts. In fact, by pursuing the distribution matching paradigm one tries to approximate the importance function which is an exact transformation function. A further advantage of the approximation $\hat{w}$ given in (6.1) is that it is a direct way that avoids several pitfalls of the naive density estimation approach. Although the approximation (6.1) reduces the number of potential error sources, there still exist circumstances which have an impact on the approximation quality of (6.1). One critical property is the number and overlap of the given data: if not enough information or overlap is provided one cannot learn anything from the given data. However, if all requirements for a good approximation are satisfied, the application of divergence measures provide a range of different ways for estimating appropriate weight coefficients. By doing so, it is important that the canonical distributions p^P and p^S somehow vanish from the calculation since due to the lack of knowledge, they cannot be employed. This is mostly achieved by taking the empirical mean by employing the given data.

By following the described instance based approach, the next two chapters will introduce several new instance based approaches for compensating covariate shift and inductive transfer learning.

Chapter 7

Compensating Covariate Shift

The covariate shift setting (a.k.a. transductive transfer learning) has been discussed in sections 4.4.1 and 5.2.1. This chapter states a new approach for compensating such a shift and can therefore be considered a new method for the transfer learning problem setting. This new approach presented here will be, in particular, an instance based method that applies the importance sampling technique discussed in the previous chapters. But since the exact distributions of the datasets are unknown, it is necessary to approximate the weight function. Therefore, the approximation of the importance sampling function given in (6.1) is taken into account. The performance of the new method will be compared with KLIEP [91], Kernel Mean Matching [45] and uLSIF [98] since these methods are well known within the community and are often considered for comparing new approaches within the literature. Another reason is that all these methods are also instance based. They all assume some overlap of the samples from the two distributions p^S and p^P. As explained in section 6.2 if no overlap exist, i.e. where the secondary and primal distribution have nothing in common, no similarity between the data can be derived. In such a situation, without additional strong assumptions about the types of distributions involved, it will not be possible to derive reasonable information from the given data for the calculation of the weights.

The main idea of the proposed new approach for estimating the appropriate importance weights is based on the application of a Fourier approximation on different divergence measures. As a consequence, 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 can be applied to any distance measure, and a specific point set for empirically estimating the distance measure. The applied distance measures include the minimization of the total variation distance, the Kullback-Leibler divergence and the Euclidean distance. The primal and secondary data are then used during the estimation of the Fourier coefficients 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, some evidence will be given that under certain circumstances the application of the Fourier series will lead to a better weight estimation in comparison to other approaches. The curse of dimensionality for which the Fourier series approximation is prone to for high dimensional functions will be tackled with the Hyperbolic cross approach [10, 49, 88] which enables the application of a Fourier series approximation to high dimensional functions by simultaneously keeping an acceptable degree of accuracy.

7.1 New Fourier Based Approach

The new approach is now motivated and derived for the calculation of importance weights for the secondary data. Mathematically speaking, one would like to minimize the distance of the primal/test distribution p^P and the secondary or training distribution p^S which is reweighted by w

$$\min_w D(p^P(x)\|w(x)p^S(x)). \tag{7.1}$$

Expression (7.1) can be minimized using different distance measures. Typically, divergence measures from the classes of Csiszár or Bregman divergences are chosen, which are then empirically evaluated on some points $\{x_l\}_{l=1}^L$. Here one often uses secondary datapoints $X^S \subset \mathbb{R}^{D\times N}$ or primal data points $X^P \subset \mathbb{R}^{D\times M}$ for the evaluation points in the distance estimation.

The class of Csiszár divergences are now considered, which are defined as $D_h(p||q) = \sum_{l=1}^L q_l h(\frac{p_l}{q_l}), L \in \mathbb{N}$, where h is a real-valued convex function satisfying $h(1) = 0$ (for abbreviation define $p_l := p(x_l)$, $q_l := q(x_l)$). Different h yield different divergences. For the following exposition h is set to $h(u) = |u-1|$. Considering that $q_l > 0$ $\forall i$ the total variation distance becomes:

$$D_h(p||q) = \sum_{l=1}^L q_l \left|\frac{p_l}{q_l} - 1\right| = \sum_{l=1}^L |p_l - q_l|. \tag{7.2}$$

Substituting (7.2) into (7.1) yields:

$$\min_w D_h(p^P(x)\|w(x)p^S(x)) = \min_w \sum_{l=1}^L |p^P(x_l) - w(x_l)p^S(x_l)|. \tag{7.3}$$

It should be noted that in contrast to many other approaches, the methodology does not depend on a specific choice of the points $\{x_l\}_{l=1}^{L}$ and is able to use any point set in the distance estimation (7.3). Nevertheless, for the sake of comparison with other approaches, either the primal or secondary data points are used in (7.3) and for the experiments in section 7.4.

The Fourier based approach which will be described 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 $\sum_{l=1}^{L} |p_l^\gamma - q_l^\gamma|^{\frac{1}{\gamma}}$, could be used. In later sections, the approach will be combined with the Kullback-Leibler divergence and the Euclidean distance, respectively.

7.1.1 Choice of the Weight Function

The optimization problem (7.1) states the problem of finding an optimal weight function $w(x)$ which minimizes the distance of the two functions p^P and $w \cdot p^S$. The exact solution would be the quotient of the density functions, i.e. $w(x) = {}^{p^P(x)}/_{p^S(x)}$, which of course is not available and cannot be computed. Instead, as introduced in chapter 6.1, the approximation of w namely $\hat{w}$ (6.1) is applied. The experiments will use the primal data as the center points $\zeta = X^P$, as in [47, 91]. In that case, it is argued that using P data 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 secondary density $p^S(x)$ is small and the primal density $p^P(x)$ is large. Note that the ratio $w(x) = {}^{p^P(x)}/_{p^S(x)}$ 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 the linear combination of Gaussian kernels is considered in this work.

Inserting (6.1) into (7.3) then yields

$$\min_{w} D_h(p^P(x) \| w(x) p^S(x)) \approx \min_{\alpha \geq 0} \sum_{j=1}^{N} |p^P(x_j) - \hat{w}(x_j, \alpha) p^S(x_j)|. \tag{7.4}$$

However, this minimization problem still employs the probability densities directly. The next step will be to approximate this term by using a Fourier series approximation, which then removes the explicit densities.

7.1.2 Fourier Series Approximation

The new approach is based on the ideas introduced in chapter 3. It will employ the Fourier series approximation (as it was explained in section 3.1), with which the employed distance measure will be discretized by taking a more function centric view as opposed to the more common data centric view. Section 7.2.2 provides a discussion of the advantages of this new approach, and section 3.2 explains the case of more than one dimension.

A suitably smooth function f can be approximated by a Fourier series in a controlled fashion via a truncation with $|k| \leq K, k \in \mathbb{Z}$

$$f(x) \approx \sum_{k=-K}^{K} c_k e^{i\frac{2\pi k}{T}x}, \tag{7.5}$$

where K is chosen to achieve a given error, see section 3.2 for more details on the approximation properties.

Of interest now is the error function between the two densities:

$$f(x) := p^P(x) - \hat{w}(x,\alpha)p^S(x).$$

It is assumed that the given data (here in the one dimensional case, $D = 1$) is bounded to a certain region, i.e. $X^P \cup X^S \subset [t, t+T] \subset \mathbb{R}$, for suitable chosen t, T. Assuming periodicity of f on that interval implies that the same small error is made on the boundary, which is the 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 one can assume a continuous periodic extension of the Fourier series of f and avoid the Gibbs phenomenon, i.e. potential overshoots on the boundary, in practice.

The Fourier series approximation will now be applied to problem (7.4). From its definition, the densities are replaced by the empirical samples in the formula (3.2) for the coefficients c_k after splitting the integral into two:

$$c_k(\alpha) = \frac{1}{T}\int_t^{t+T} e^{-i\frac{2\pi k}{T}x} dp^P(x) - \frac{1}{T}\int_t^{t+T} \hat{w}(x,\alpha) e^{-i\frac{2\pi k}{T}x} dp^S(x) \tag{7.6}$$

$$\approx \hat{c}_k^{(M,N)}(\alpha) = \frac{1}{TM}\sum_{m=1}^{M} e^{-i\frac{2\pi k}{T}x_m^P} - \frac{1}{TN}\sum_{n=1}^{N} \hat{w}(x_n^S,\alpha) e^{-i\frac{2\pi k}{T}x_n^S}. \tag{7.7}$$

In the last part of this equation, the two integrals are approximated by taking the empirical expectation based on the training data $\left\{x_n^S\right\}_{n=1}^N \sim p^S$ and test data $\left\{x_m^P\right\}_{m=1}^M \sim p^P$, respectively. Therefore, the unknown densities are no longer explicitly needed but their known samples can be used.

7.1.3 Optimization Problem

The original problem (7.1) is about finding an appropriate weight function. Employing $\hat{w}$ (6.1) for given parameter σ and center points $(\zeta_j)_{j=1}^Z$ and using the Fourier approximation (7.5) for a suitably chosen K one obtains the following optimization problem:

$$\min_{\alpha \geq 0} \sum_{l=1}^{L} \left| p^P(x_l) - \hat{w}(x_l, \alpha) p^S(x_l) \right| \approx \min_{\alpha \geq 0} \sum_{l=1}^{L} \left| \sum_{k=-K}^{K} \hat{c}_k^{(M,N)}(\alpha) e^{i \frac{2\pi k}{T} x_l} \right| . \qquad (7.8)$$

Due to the linearity, this problem can be expressed in matrix notation. Defining the matrix $A \in \mathbb{R}^{L \times Z}$ as $A = [A_1 | \dots | A_L]$, where the $A_l \in \mathbb{R}^Z$ are column vectors comprised, after inserting the explicit expression (6.1) for $\hat{w}$, of the entries:

$$(A_l)_j = \sum_{k=-K}^{K} \frac{1}{TN} \sum_{n=1}^{N} e^{-\frac{\|x_n^S - \zeta_j\|^2}{2\sigma^2}} e^{-i \frac{2\pi k}{T} x_n^S} e^{i \frac{2\pi k}{T} x_l}, \qquad j = 1, \dots, Z. \qquad (7.9)$$

Additionally a vector $b \in \mathbb{R}^L$ is obtained, that is defined as:

$$b_l = \sum_{k=-K}^{K} \sum_{m=1}^{M} \frac{1}{TM} e^{-i \frac{2\pi k}{T} x_m^P} e^{i \frac{2\pi k}{T} x_l}, \qquad l = 1, \dots, L. \qquad (7.10)$$

The problem (7.8) can now be stated as a L_1 minimization problem with side conditions in a compact notation by employing A and b

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

The latter expression is a $L1$-norm of system of linear equations and, therefore, convex [11].

7.1.4 Normalization Constraints

It is possible that a solution to the optimization problem (7.8) 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 data points will get importance. To compensate, an approach will be employed which is similar to the one introduced in [91]. Given expression $p^P(x) = w(x)p^S(x)$ derived from the expression $w(x) = {}^{p^P(x)}/_{p^S(x)}$, and taking the integral on both sides yields the natural side condition:

$$1 = \int p^P(x)dx = \int w(x)dp^S(x) \approx \frac{1}{N}\sum_{n=1}^{N} \hat{w}(x_n^S, \alpha),$$

again using the empirical samples and the approximation $\hat{w}$. This side condition ensures that the transformed p^S is again a valid density. By augmenting (7.8) one gets a new constrained optimization problem which can e.g. be solved with the solver *Yall1* [106] (Yall1 applies alternating direction algorithms for a diverse set of $l1$ problems):

$$\min_{\alpha \geq 0} \|A\alpha - b\|_1 \text{ s.t. } \frac{1}{N}\sum_{n=1}^{N} \hat{w}(x_n^S, \alpha) = 1. \tag{7.11}$$

An instruction for the implementation of the complete procedure is given by:

```
input  : Data $X^S \in \mathbb{R}^{N \times D}$ and $X^P \in \mathbb{R}^{M \times D}$
output: Vector $\alpha \in \mathbb{R}^N$ for obtaining weights from $\hat{w}(x, \alpha)$ (6.1)
Initialize with zeros $A \in \mathbb{R}^{M \times N}, b \in \mathbb{R}^M$ and $W \in \mathbb{R}^N$
for $l \leftarrow 1$ to $M$ do
    $b_l \leftarrow$ (7.10)
    for $j \leftarrow 1$ to $N$ do
        $A_{lj} \leftarrow$ (7.9)
    end
end
Calculate the constraint
for $j \leftarrow 1$ to $N$ do
    for $l \leftarrow 1$ to $M$ do
        $W_j \leftarrow W_j + e^{-\frac{||x_j^S - x_l^P||_2^2}{2\sigma^2}}$
    end
    $W_j \leftarrow \frac{1}{N} W_j$
end
For the final optimization problem: Randomly initialize $\alpha \in \mathbb{R}^N$
Apply $A, b, W$ to the solver Yall1 [106] with the configuration "(BP+)" and
solve $||A\alpha - b||_1$ w.r.t. the constraint $W\alpha = 1$
Retrieve a weight for arbitrary $x^*$ by applying the resulting $\alpha$s and $x^*$ to (6.1)
```

7.1.5 Kullback-Leibler Divergence

An advantage of the Fourier approach is that it can directly be applied to different divergence measures. To demonstrate this flexibility, a second Csiszár divergence namely the Kullback-Leibler divergence will be used, which also allows the comparison with

KLIEP [91]. Roughly following the KLIEP derivation gives:

$$\mathrm{KL}(p^P \| wp^S) = \sum_{l=1}^{L} p^P(x_l) \log\left(\frac{p^P(x_l)}{w(x_l)p^S(x_l)}\right)$$
$$= \sum_{l=1}^{L} p^P(x_l) \log\left(\frac{p^P(x_l)}{p^S(x_l)}\right) - \sum_{l=1}^{L} p^P(x_l) \log\left(w(x_l)\right).$$

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

$$\arg\min_{w} \mathrm{KL}(p^P \| wp^S) \approx \arg\min_{\alpha \geq 0} - \sum_{l=1}^{L} p^P(x_l) \log\left(\hat{w}(x_l, \alpha)\right), \tag{7.12}$$

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

$$\min_{\alpha \geq 0} \sum_{l=1}^{L} \sum_{k=-K}^{K} \hat{c}_k^{(M)}(\alpha)\, e^{i\frac{2\pi k}{T} x_l} \quad \text{s.t.} \quad \frac{1}{N} \sum_{n=1}^{N} \hat{w}(x_n^S, \alpha) = 1, \tag{7.13}$$

where

$$\hat{c}_k^{(M)}(\alpha) = -\frac{1}{TM} \sum_{m=1}^{M} \log\left(\hat{w}(x_m^P, \alpha)\right) e^{-i\frac{2\pi k}{T} x_m^P} \tag{7.14}$$
$$\approx \frac{1}{T} \int_t^{t+T} -\log\left(\hat{w}(x, \alpha)\right) e^{-i\frac{2\pi k}{T} x} dp^P(x) = c_k(\alpha).$$

Although the approach is very similar to the one suggested by [91], the optimization problem is different due to the Fourier approximation and also the divergence is estimated 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.

7.1.6 Euclidean Distance

The third distance measure that will be investigated is the Euclidean distance, which belongs to the class of Bregman divergences, and was also used for uLSIF [47]. The *Squared Euclidean* distance was derived in expression (2.7) by:

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

Thus, employing the data, the weight function $\hat{w}$ and applying the Fourier approximation the following optimization problem and corresponding normalization constraint is

```
input  : Data $X^S \in \mathbb{R}^{N \times D}$ and $X^P \in \mathbb{R}^{M \times D}$
output: Vector $\alpha \in \mathbb{R}^N$ for obtaining weights from $\hat{w}(x, \alpha)$ (6.1)
Initialized vector $\hat{c}$ of length $2K + 1$ with zeros
for $k \leftarrow -K$ to $K$ do
    $\hat{c}_k \leftarrow$ (7.14)
end
Calculate the constraint: Initialize $W \in \mathbb{R}^N$ with zeros
for $n \leftarrow 1$ to $N$ do
    for $p \leftarrow 1$ to $M$ do
        $W_n \leftarrow W_n + e^{-\frac{||x_n^S - x_m^P||_2^2}{2\sigma^2}}$
    end
    $W_n \leftarrow \frac{1}{N} W_n$
end
For the final optimization problem: Randomly initialize $\alpha \in \mathbb{R}^N$
Apply the calculated $\hat{c}_k$s and $W$ to (7.13) a solver, e.g. IPOpt
Retrieve weights for arbitrary $x^*$ by applying the resulting $\alpha$s and $x^*$ to (6.1)
```

Algorithm 1: Pseudo code for the KL Fourier method described in (7.13).

obtained:

$$\min_{\alpha \geq 0} \|A\alpha - b\|_2^2 \quad \text{s.t.} \quad \frac{1}{N} \sum_{n=1}^{N} \hat{w}(x_n^S, \alpha) = 1, \tag{7.15}$$

where A and b are defined as in (7.9) and (7.10) respectively. As for the KL setting above, this problem is also solved by applying *IPOpt* [101].

7.1.7 Hyperbolic Cross Approximation

Until now, only a one dimensional Fourier series was considered. The straightforward D-dimensional generalisation of a Fourier approximation for $f : \mathbb{R}^D \mapsto \mathbb{R}$ implies the problems discussed in section 3.1.1. Since this straight forward approach is practically infeasible, the Hyperbolic cross (3.2) will be applied to this problem. Therefore, the computational costs reduce from $(1+K)^D$ coefficients to $\mathcal{O}\left((1+K)\left(\log(1+K)\right)^{D-1}\right)$. Using a Hyperbolic cross one achieves for $f \in \mathcal{H}^s_{mix}$ the same order of approximation as the standard Fourier approximation. A question is if $p(x) - \hat{w}(x)q(x) \in \mathcal{H}^s_{mix}$ can be expected, 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 indications can be given that the assumption $p, q \in \mathcal{H}^s_{mix}$ is warranted, if one expects reasonably smooth probability distributions at all. Firstly, the mixed Sobolev spaces have an intrinsic tensor product structure with distinguished dimensions, each of which can be related 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, 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 [33, 68].

7.2 Investigation of Diverse Properties on Synthetic Data

The following sections investigate some properties of the newly introduced Fourier methods and shows advantages and discussed problems that might occur. It is also serves the purpose of illustration of the method.

7.2.1 Regularization Effect of the Fourier Approximation

In a first experiment, the behaviour of the procedure in regard the number of Fourier coefficients and the number of data will be studied in one dimension. For illustrative purposes, a standard normal distribution is considered. The example is based on the effect discussed in section 3.1. Three curves are shown in each subplot of figure 7.1. The black curve is the plot of the exact function, the normal distribution, plus two types of approximations.

The first is a Fourier series approximation of degree $K \in \{3, 5, 10, 15, 50\}$ where the integral for each Fourier coefficients c_k, as given in (3.2), is calculated by numerical integration, in this case adaptive Gauss-Kronrod quadrature, using the known exact distribution. The resulting Fourier approximation is shown by dashed red curve. The second is a Fourier approximation where the c_k coefficients are computed by Monte-Carlo integration, which is shown by the blue curve. Using a Monte-Carlo integration for the c_k coefficients can be viewed as taking the empirical mean according to the available data. Therefore, this approach is comparable to the previously derived optimization problems. Here weighted Monte-Carlo is used, i.e. a sampling according to the distribution.

For each column in Figure 7.1 a different number $N \in \{500, 5000, 50000\}$ of sample data/Monte-Carlo points for evaluating the empirical mean are taken. One can see from the plots that for lower K the obtained Fourier approximation is very smooth. As N is increased the accuracy of the approximation gets better, in particular for larger K. On the other hand, when K is increased for fixed N overfitting can be observed, which consequently would be reduced by additional datapoints for the calculation of the

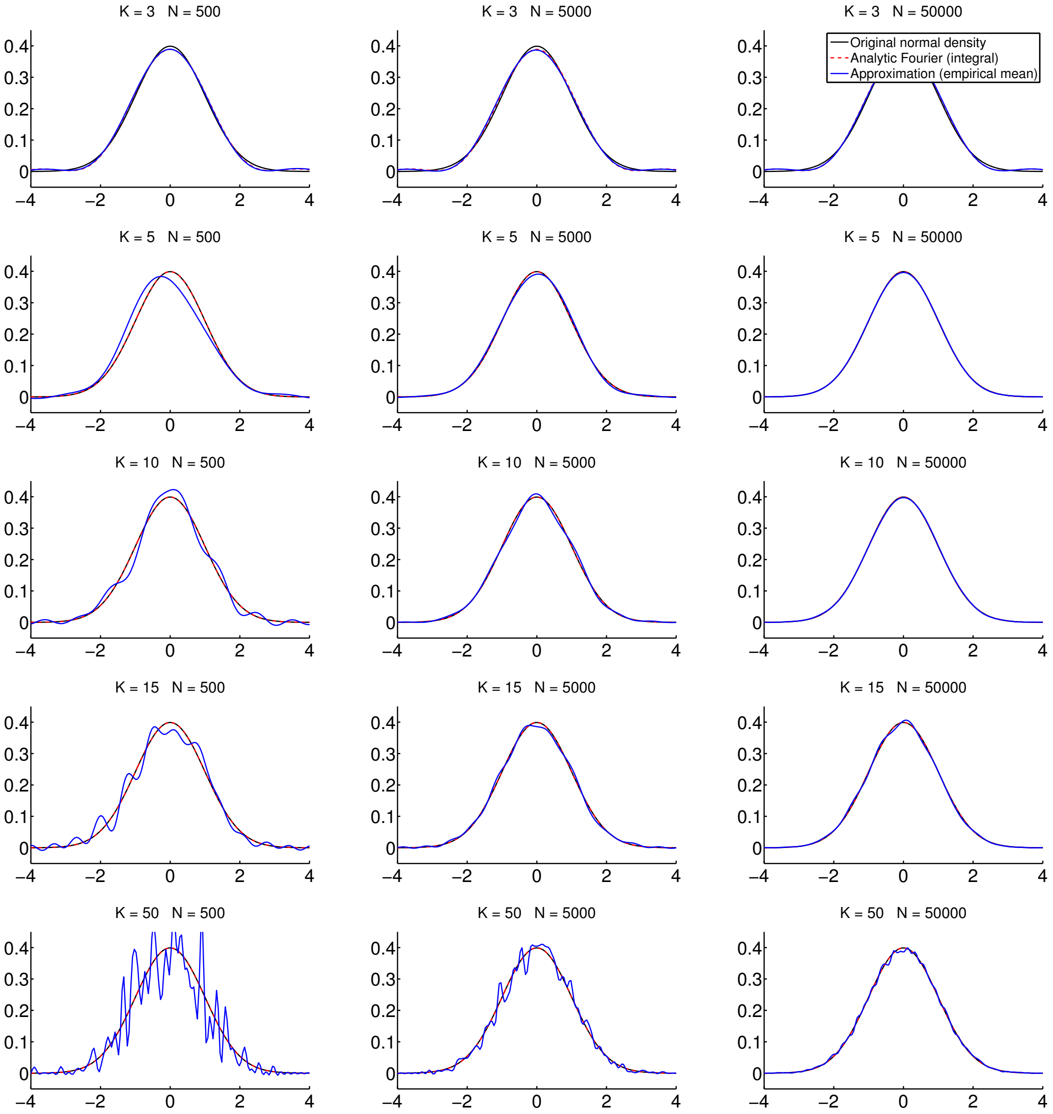


FIGURE 7.1: Fourier approximation of a standard normal distribution. The black line is the original analytic density function, which is the same in each plot. The dashed red curve is the approximation of this density by a truncated Fourier series of degree $K \in \{3, 5, 10, 15, 50\}$, this is the same for each N. The blue line shows the Fourier series where the Fourier coefficients c_k are approximated by the empirical mean, this curve varies in K and N

empirical mean. However, since it is assumed that no further data can be obtained, this cannot be a solution to the problem at hand.

Hence, by choosing a smaller value for K, and therefore considering only the lower frequencies and ignoring high frequencies, results in a more robust approximation of the original function by the truncated Fourier series approach when the c_k are computed using the data dependent sample mean. This beneficial effect of coarser resolutions for data-dependent problems is also known as regularization by discretization, or regularization by projection, going back to [60]. As can be seen, the Fourier approximation with exact c_k is for this example already almost the same as the original function for

$K = 5$. For more complex functions or divergence measures a higher degree K is in order to achieve a reasonable approximation.

7.2.2 Benefits of the Fourier Approximation

The following illustrative example shows the behaviour of the new Fourier approaches. The weight function $\hat{w}$ is chosen according to (6.1). In contrast to the previous setting where the focus was on the approximation quality of the empiricial approximation of the coefficients, this section will illustrate the actual approximation of the weight function $\hat{w}$ and compares the quality of fit with other approaches. Therefore, this toy example gives better insights of the consequences of the approximation for the weight function. For the sake of comparison, the Kullback-Leibler divergence and the Euclidean distance are used here.

As observed, estimating a function, i.e. the divergence measure, using the truncated Fourier approximation achieves a smoothing of the function, i.e. the weights. This becomes especially useful when a small bandwidth parameter σ is chosen for the weight function $\hat{w}$. As figure 7.2 illustrates, the weights learned by the Fourier methods are much smoother and stable than the weights learned by KLIEP [91] or uLSIF [47] which involve a much higher volatility. For the sake of comparison, in figure 7.2 the same bandwidth parameter σ was applied to the Fourier methods that was chosen by KLIEP. Although a parameter selection method that is not necessarily appropriate for the Fourier approaches was applied, the Fourier methods outperform KLIEP, in the sense of a less volatile weight function. The parameters for uLSIF have been determined by its own parameter estimation method.

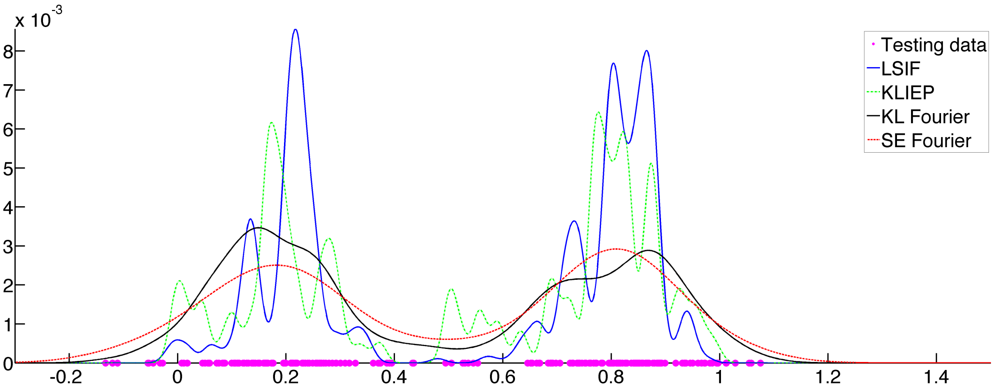


Figure 7.2: Plot of the learned weight functions $\hat{w}$ for a 1D toy example. Regions of low P data (magenta) density imply small weights while high density regions imply large weights. KLIEP (green) and uLSIF (blue) compute much more volatile weight functions than the Fourier methods. Total variation Fourier was omitted in this plot for clarity of the diagram and due to similarity to the other shown Fourier results.

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 smoothing of the Fourier approximation method. As already argued, 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 it becomes possible to learn more appropriate weights.

Another property of the approach is that the distance measures can be estimated on any point set. The secondary or the primal data are just one way, and not the only set of locations where to estimate the distances (7.11), (7.13), or (7.15). Merely the computation of the Fourier coefficients requires the S and P data. This is possible since firstly by empirically estimating the distance and secondly by applying the Fourier approximation. That is for example different to KLIEP where the error is calculated on the primal data and cannot be straightforwardly computed on the S data, or uLSIF where the S and P 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 secondary data is beneficial since the weights are employed for the secondary data in the regression algorithm and therefore for those points the distance should be small. This hypothesis is supported in section 7.4, where we calculated the distance using the secondary or the primal data for each Fourier method.

7.2.3 Properties Hyperbolic Cross Approximation

As mentioned in section 7.1.7 the application of the hyperbolic cross makes the method feasible in higher dimensional spaces. Insights on the appropriateness of the application of the Hyperbolic cross are given in the experimental results in the next sections as well as the following empirical study of the decay of the absolute values of the Fourier coefficients c_k once for the full complete Fourier series and once for the set of coefficients derived from the Hyperbolic cross. Therefore, a 2 dimensional problem is considered. To give a simple and clear illustration, the first problem will be the decay of the coefficients of the approximation of the euclidean distance as given in (7.15) for some normally distributed toy data. This toy data is sampled from two 2 dimensional normal distributions with means $\mu^P = (1, 1)$ and $\mu^S = (0, 0)$ for the primal and secondary distribution, respectively. The variance is equal in each dimension with $\eta^P = 3$ and $\eta^S = 1.5$. From each distribution 100 datapoints are sampled yielding $|X^P| = |X^S| = 100$ for each dataset sample. Given this data, expression (7.15) can be computed which yields a set of coefficients that can be plotted as a 2D heatmap. The colors of the heatmap denote the magnitude of the absolute value $|c_k| \in \mathbb{R}^+$ of each coefficient for each frequency combination (k_1, k_2) with $k_1, k_2 \in \{-K, \dots, K\}$. For the Hyperbolic cross, the set of coefficients denotes a subset of these frequency combinations. The full set of coefficients is shown in the left plot in figure 7.3. There, the largest values for the coefficients are located close to the center $k = (0, 0)$. Since only those large coefficients have an actual impact on the approximation of the original function by the Fourier series all other coefficients (that have a small value) could be neglected without having a huge loss of approximation quality. In fact, by looking at figure 7.3 one can see that the coefficients outside the "hotspot" tend to be noisy which might have a negative effect on the approximation. On the other hand, the Hyperbolic cross (figure 7.3, right plot) only pays attention to exactly those coefficients that are important for the approximation. Thus, the Hyperbolic cross not only reduces the amount of coefficients that have to be calculated but also removes the coefficients that could potentially harm the approximation due to being noisy. Also, from the plot, one can see that the actual upper bound for the frequency K can be set to a very small value. The reason for that is that the normal distribution is very smooth and can be captured by considering only a very low number of frequencies. However, this is specific for this toy example and can therefore not be generalized.

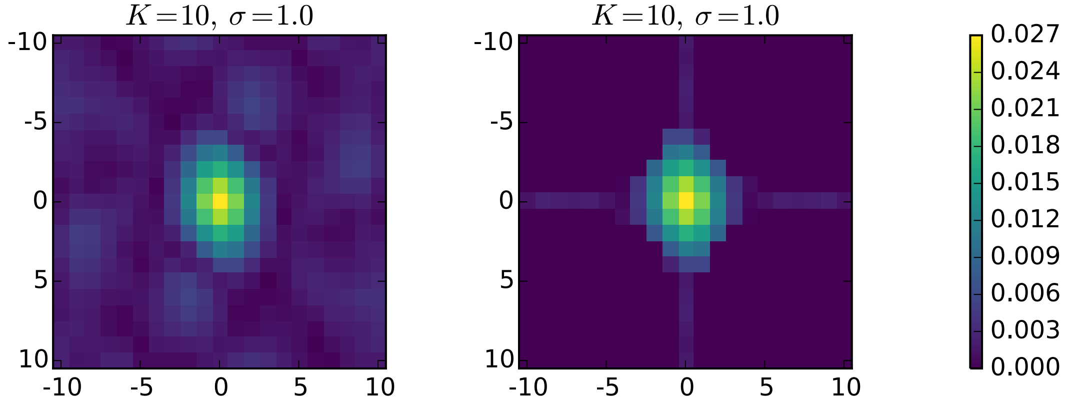


FIGURE 7.3: Comparison of the impact of each coefficient c_k of the frequency combinations $k = (k_1, k_2) \in \{-K, \dots, K\} \times \{-K, \dots, K\}$. Each combination of the x and y axis denote such a frequency combination. The left plot is the full set of coefficients and the right set is the Hyperbolic cross (note the characteristic shape of the coefficients). From the plots one can see that the Hyperbolic cross very accurately captures only those coefficients that are also large in the full set. Hence the Hyperbolic cross only applies the important coefficients and neglects higher frequencies which only seem to have a noisy influence.

By considering a toy example one can gain an intuition about when a Hyperbolic cross approximation might be beneficial. But since this does not reflect a realistic setting it is interesting to further consider an example from an actual real world dataset. Therefore, here a 2 dimensional subset of the earth quake dataset [5] which is also used in the experimental section is considered in order to demonstrate the effectiveness of the Hyperbolic cross approach. The dataset is 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, the dataset is augmented and gets assigned a separate dimension for each category, which turns one dimension into three. For figure 7.4 a 2D subset of magnitude and distance to the center is taken from the full dataset in order to provide a visual illustration. The plots show that the Hyperbolic cross approximation is very applicable here. One could even further reduce the amount of coefficients need by further reducing the set of coefficients. Therefore, in the real world example the amount of coefficients that need to be calculated can also be reduced without observing any negative impact on the approximation quality.

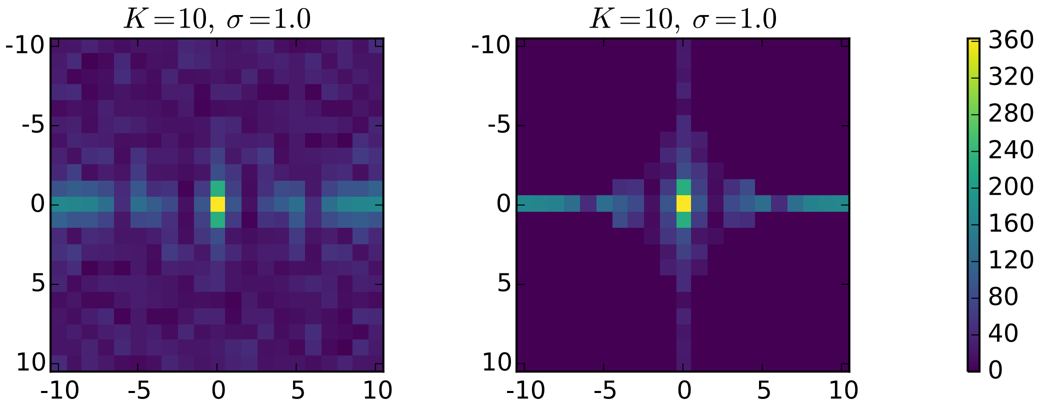
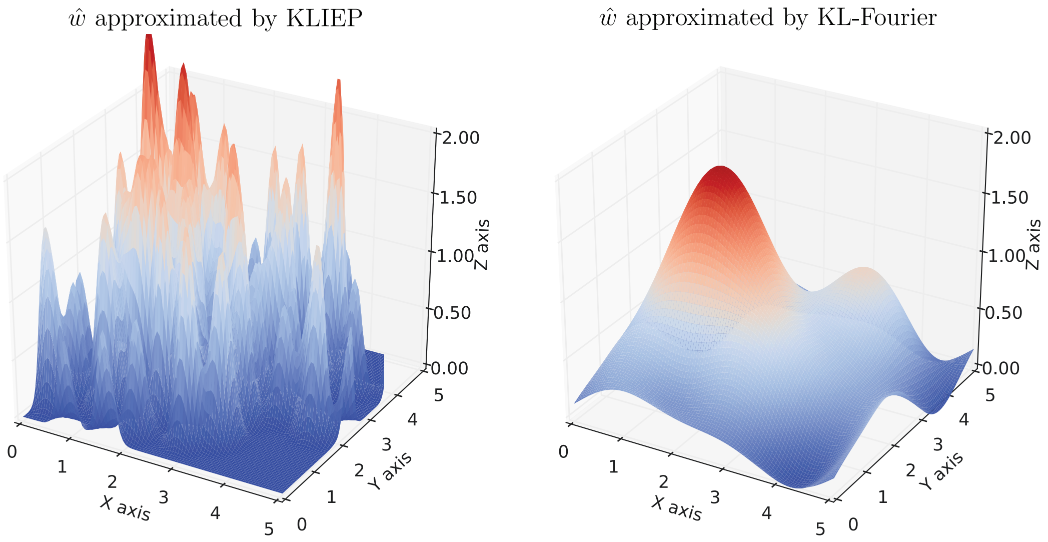


FIGURE 7.4: Comparison of the absolute value of the coefficients $|c_k|$ of the euclidian distance (7.15) for a 2D subset of the earth quake dataset [5]. The legend reads as in figure 7.3. However, here a real world problem is consider. The left plot shows, again, the full Fourier coefficients set and the right those of the Hyperbolic cross. In contrast to the toy data plot (7.3) some coefficient at higher frequencies have some impact. However, the most important coefficients lie again around the center such that the Hyperbolic cross captures the important information very well. In fact, from the shown plots one could also say that still the Hyperbolic cross considers too many coefficients.

Another question that arises is the approximation quality of the function $\hat{w}$ given the amount of data for each method. To answer this, another toy example will be applied to demonstrate the implications by measuring the difference between the real analytical importance function $w(x) = {}^{p^P(x)}/_{p^S(x)}$ and the approximation $\hat{w}(x)$ as it was defined in 6.1. The specification of the analytical importance function $w(x)$ requires the knowledge of the exact distributions p^P and p^S. Therefore, both distributions are considered to be normal distributions with $p^P \sim \mathcal{N}(\mu_1, \eta)$ and $p^S \sim \mathcal{N}(\mu_2, \eta)$, where the means $\mu_1, \mu_2 \in \mathbb{R}^2$ are $\mu_1 = (0,0), \mu_2 = (1,1)$ and the variance is $\eta = 1.0$. The S data is sampled once and then kept fix such that it consists of 1000 data samples. The bandwidth parameter σ in $\hat{w}(x)$ is chosen to be $\sigma = 0.1$ in order to provide a more consistent experimental setting. The P data is varied such that for the P data samples of



$\hat{w}$ approximated by KLIEP

$\hat{w}$ approximated by KL-Fourier

Figure 7.5: Different results for the approximation of the weight function $\hat{w}$. The data is a sample of two gaussian distributions with identical variance but different mean. The left plot shows the weight surface of the weight function approximated by KLIEP. The right plot is the approximated weight function as a result of the KL Fourier method which employs the Hyperbolic cross. The plot shows the consequences of the Fourier smoothing as a 3D surface plot (compare with figure 7.2).

$100, 500, 1000, 2000$ and 5000 data points are applied. Then, for each combination of the S and P data the weight approximation function $\hat{w}$ is estimated once for KLIEP, for KL-Fourier with $K = 10$ and KL-Fourier with $K = 50$. The three resulting approximations of w are then evaluated by calculating the Mean Square Error (MSE) on a separately sampled and fixed set of 10000 equidistant datapoints $\tilde{X} \subset \mathbb{R}^D$ that fully cover the S and P data, i.e. $^1\!/_{10000} \sum_{i=1}^{10000} ||w(\tilde{x}_i) - \hat{w}(\tilde{x}_i)||_2^2$. This type of calculation reduces the error of the actual P data sampling.

Figure 7.6 shows the results. As can be seen from the figure, not surprisingly, the more data available, the smaller the error. This holds true especially for KLIEP. However, the figure also suggests that in the beginning, a lower number of data points might have a negative effect on the calculation of the approximation $\hat{w}$ when no type of any smoothing is applied. On the other hand, the KL-Fourier method with $K = 10$ performs better at a low number of data points but then becomes relatively worse when more and more data becomes available. The reason for that might be the filtering property of the truncation of the Fourier series. This effect could also be responsible for the higher error at #P-Data $= 5000$. Although the number of data points grows the fitting does not get better when considering more data points because the structure for $K = 10$ is not able to capture more details. This hypothesis appears to be supported by comparing the KL-Fourier method with $K = 10$ versus $K = 50$. For $K = 50$ the error is at the beginning higher but at the end lower. This might be due to a mild overfitting at the

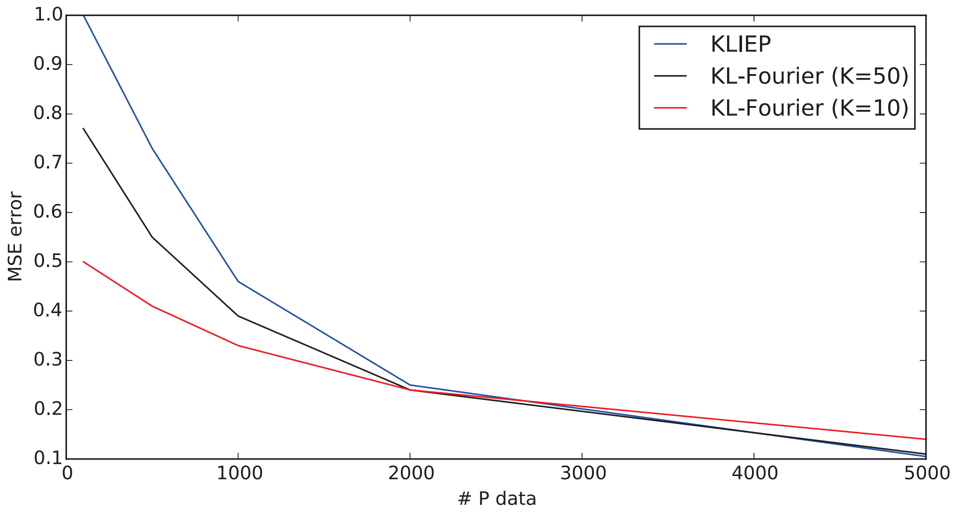


Figure 7.6: MSE between the real analytical function $w(x)$ and it's approximation $\hat{w}(x)$ measured on 10000 equidistant datapoints that completely cover the S and P data. As more data becomes available all methods become better. However, at the beginning the smoothing property of the Fourier approaches yields a better approximation while at a higher number of data points a Fourier method with truncated at a low K lacks of capturing more details. The effect is less present when considering Fourier with $K = 50$ but then the overfitting at the start has a higher impact. The errors have been normalized to the initial error of KLIEP (100% error) in order to make the comparison easier.

start and a lack of exactness at the end due to the filtering properties.

To better understand this effect, another toy example is considered. It applies the same setting as it was used for figure 7.6. However, in figure 7.7 the data is sampled once and then kept fix. Instead, the x axis now depicts different values of $k = \{10, 20, 30, 40, 50, 75, 100\}$ and plots the MSE error between the KLIEP and Fourier approximation and the real analytical weight function $w(x)$. For these experiments the, data samples are # S data = 1000 for both plots and # P data = 1000 for the left plot and # P data = 5000 for the right plot. Since the error between KLIEP and w is constant for each sample, the MSE of KLIEP will be denoted by 100% of the error. Thus figure 7.7 shows the relative performance of the Fourier method in comparison to KLIEP.

The left plot in figure 7.7 shows that in a setting with a very low number of datapoints the reduction of the K has a beneficial impact on the error made. However, if K is increased and more and more higher frequencies are considered the approximation $\hat{w}$ lines up to the KLIEP approximation. On the other side, the plot on the right side shows that if a lot of data is already given, then, the approximation at low values of K becomes worse since then the noise gets a lower impact and the weight function learned by KLIEP becomes better. These given toy examples give more insights that when

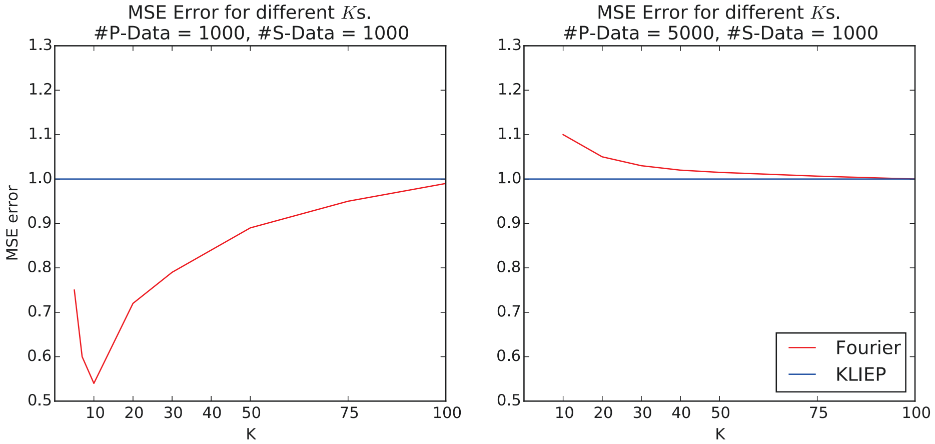


Figure 7.7: Comparison of the error between the approximation $\hat{w}(x)$ and the real analytical importance function $w(x)$ retrieved by applying KLIEP (100% of the error) and the error of the KL-Fourier method at different frequency cuts K. The left plot suggests that not enough data is available for KLIEP to infer an appropriate approximation. However, the smoothing property of the KL-Fourier method becomes beneficial since the noise (contained in higher frequencies) gets a lower impact. If K gets increased the error of the KL-Fourier approach lines up with KLIEP. The right plot shows the opposite: if enough data is given already, then KLIEP infers a better weight approximation function since noise has an overall lower impact on the inference.

dealing with a very low number of data points (either P data or both S and P data) the Fourier methods should be preferably applied. On the other hand, if more data is available, it is advisable to increase the truncation K such the complex structures are better fit.

7.3 Convergence of the Empirical Fourier Approximation

The functions $f(x)$ that are considered in the context of the Fourier approximation involve at least in parts a density $p(x)$. Furthermore, the given samples, the data points, $\{x_1, \dots, x_L\} \subset \mathbb{R}^{D \times L}$ are from this distribution p. The integral (3.3) therefore involves a density, see e.g. (7.6), so the integral can be rewritten as being in respect to that density and then the employed empirical samples are considered as a kind of Monte Carlo integration.

As a consequence, the result from section 3.3 can be applied. Equipped with these theoretical preparations provided there, the statistical error bounds can now be derived for the Fourier coefficients that are employed. For the total variation distance and the squared Euclidean distance the same integral is used for the Fourier coefficients and obtained by its empirical/Monte Carlo estimation:

$$\hat{c}^{(M,N)}_{k_1,\ldots,k_D}(\alpha) = \sum_{m=1}^{M} \frac{e^{-i2\pi \sum_{d=1}^{D} \frac{k_d}{T_d} x^P_{dm}}}{M \prod_{d=1}^{D} T_d} - \sum_{n=1}^{N} \frac{w(x^S_n, \alpha) e^{-i2\pi \sum_{d=1}^{D} \frac{k_d}{T_d} x^S_{dn}}}{N \prod_{d=1}^{D} T_d}.$$

Both sums are normally distributed with empirical variances η^2_M and η^2_N And the $(1-\gamma)\%$ confidence interval becomes:

$$\left[\hat{c}^{(M,N,0)}_{k_1,\ldots,k_D}(\alpha) - z_{(1-\frac{\gamma}{2})}\sqrt{\frac{\eta^2_M}{M} + \frac{\eta^2_N}{N}}\ ,\ \hat{c}^{(M,N,0)}_{k_1,\ldots,k_D}(\alpha) + z_{(1-\frac{\gamma}{2})}\sqrt{\frac{\eta^2_M}{M} + \frac{\eta^2_N}{N}}\ \right].$$

The index $(M, N, 0)$ denotes that the empirical mean has been centered by the real mean. Similarly for the Kullback-Leibler Fourier method one gets as the $(1-\gamma)\%$ confidence interval:

$$\left[\hat{c}^{(M,0)}_{k_1,\ldots,k_D}(\alpha) - z_{(1-\frac{\gamma}{2})}\frac{\eta_M}{\sqrt{M}}, \hat{c}^{(M,0)}_{k_1,\ldots,k_D}(\alpha) + z_{(1-\frac{\gamma}{2})}\frac{\eta_M}{\sqrt{M}}\right]$$

where η^2_M is the empirical variance of expression (7.14) minus the mean $c_{k_1,\ldots,k_D}$. As N tends to infinity these confidence intervals tend to zero. The resulting conclusion is that with a certain probability a given computed approximate Fourier coefficient has an error of ϵ in regard to the exact Fourier coefficient, where this error goes to zero with increasing number of points.

To directly combine the result of this section with the approximation result from 3.7 one would need an estimate for the pointwise variances jointly over the whole domain, which cannot be easily given. Nevertheless, the observations in the last two sections give justification that in the limit, both in the number of data and the Fourier resolution, the employed Hyperbolic Cross Fourier approximation converges to the exact difference function under consideration.

7.4 Experiments

The following sections will show the benefits of the Fourier approaches and that the new approach can compete with methods for compensating the covariate shift that are currently state of the art. First, the Fourier based approach is compared to other methods on some benchmark datasets, where the distance is estimated either on the S data or the P data. And secondly, results on a real world dataset are listed. The Fourier method is applied the following divergence measures: the total variation distance (TV), the Kullback-Leibler divergence (KL), and the squared Euclidean distance (SE).

7.4.1 Benchmark Datasets

The first experiments taken into account are performed on artificially generated data. This way, it is possible to compare the Fourier methods on standard datasets and in the same way other state of the art methods have been tested. Thus, for reasons of comparison, the same method for the creation of a synthetically generated covariate shift dataset is used as described in [91]. First, the dataset is normalized to $[0, 1]^D$ and then 100 datasets of 100 S data points and 500 P data points are created each. The P data samples are obtained by choosing a data point (x_l, y_l) randomly and accepting it with a sampling factor of $\min(1, 4(x_{dl})^2$, where x_{dl} is the dth element of x_l. For each of the 100 datasets the dimension $d \in \{1, \ldots, D\}$ is chosen randomly but kept fixed. Every randomly chosen x_l is removed from the pool even if it was not accepted. The secondary dataset is sampled uniformly from the remaining data. During the learning, the methods will only use the S data $\{x_n^S, y_n^S\}_{n=1}^N$ and the P data points without labels $\{x_m^P\}_{m=1}^M$. The P labels $\{y_m^P\}_{m=1}^M$ are used for performance measurements.

7.4.2 Parameter Estimation

In the experiments, a set of best parameters for a SVR and a SVM without weights (uniform) is estimated with classic cross-validation. Then the weights are calculated once with the Fourier based approach and once with the KLIEP, uLSIF and the Kernel Mean Matching (KMM) method. These weights are then employed to the weighted SVR and weighted SVM (as described in [45]) with RBF kernels and a new set of best parameters by using IWCV (Importance Weighted Cross-Validation) [90] is estimated. 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.

The 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 the experiments a larger K does not result in significantly different performance, whereas with smaller K the results degrade as one would expect. In other words, the experiments indicate that a large enough K can be easily selected. For further insights on the influence of this parameter, also, consider the synthetic experiments in section 7.2.3. The other parameter is σ, the kernel width in the weight function (6.1). A method for estimating a good σ parameter is now suggested.

The idea is that an appropriate parameter combination will minimize the expressions (7.11), (7.13), and (7.15). For given σ the corresponding αs have been determined

Table 7.1: Results for regression benchmark datasets. The results are obtained by taking the average of 100 mean errors on the P data. The values in the parentheses denote the standard deviation. All errors have been normalized by the uniform result (no weights).

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

by minimizing (7.11), (7.13), and (7.15). The lowest value of the objective functions obtained during the optimization for different σ parameters will be chosen.

To get a more stable result, a method that is similar to cross-validation is used, but will not use any label information. Given the original datasets, X^P and X^S, the P dataset is splitted into five parts, $(X_j^P)_{j=1}^5$. Each split X_j^P should contain enough samples of P data since they can normally be obtained quite easily. Each of the $j = \{1, \dots, 5\}$ folds is constructed by $X_j := X^P \setminus X_j^P$. Now for a fixed parameter σ expressions (7.11), (7.13), and (7.15) are minimized for each dataset combination $\{X^S, X_j\}$. The means of these five minima are calculated and the parameter that corresponds to the lowest average is chosen.

Minimizing the differences of the distributions of the covariates (7.11), (7.13), and (7.15) are independent of the labels of the P data. Therefore, one can explicitly make use of the locality of the P data here. This way, a simple method for estimating adequate parameters is obtained.

7.4.3 Experimental Results

For the experiments artificial covariate shift data was created as described in section 7.4.1. Data from the DELVE repository and the abalone dataset for regression was used. For classification experiments the IDA datasets which is available on mldata.org is applied. For each of the datasets 100 subdatasets were created and the P data is set as the center points of the weight function (6.1). For all datasets the mean error on the

Table 7.2: Results for classification benchmark datasets. As in table 7.1 results are obtained by taking the average of 100 mean errors on the P data. The values in the parentheses denote the standard deviation. All errors have been normalized by the uniform result (no weights).

Dataset	twonorm	waveform	ringnorm	image data	average
Dimension	20	21	20	18	-
Uniform	1.00	1.00	1.00	1.00	1.00
Fourier:TV (S)	0.92 (0.079)	0.91 (0.055)	0.96 (0.091)	0.92 (0.092)	0.92
Fourier:TV (P)	0.97 (0.072)	0.96 (0.061)	0.96 (0.081)	0.93 (0.080)	0.95
Fourier:SE (S)	0.89 (0.083)	0.90 (0.045)	0.95 (0.081)	0.90 (0.082)	0.91
Fourier:SE (P)	0.93 (0.068)	0.95 (0.052)	0.98 (0.083)	0.94 (0.069)	0.95
Fourier:KL (S)	0.91 (0.067)	0.89 (0.056)	0.96 (0.096)	0.93 (0.076)	0.92
Fourier:KL (P)	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

P data is calculated and normalized by the mean P data error of the uniform SVR or SVM, respectively. Note that the computing times of the Fourier methods and KLIEP are roughly the same, whereas uLSIF is slightly faster.

The results in tables 7.1 and 7.2 show that employing weights improves the prediction performance. The Fourier approach measuring the distance on the S data is always better than the corresponding one using the P data. A reason for the slightly poorer results on the P data might be due to the fact that for SVR and SVM one is interested in calculating weights for the S data. Therefore, it seems to be preferable to use the S data for the distance estimation to achieve on these a small distance between the P and reweighted S distribution.

The best method varies over the datasets, but on average the Fourier based approaches measuring the distance on the S data are better than KLIEP, uLSIF, and KMM for both the regression and the classification data. When comparing the results using KL, it can be observed that the Fourier based approaches when measuring the distance (7.12) on the P data is on average comparable to KLIEP, which also estimates the distance on the P data, whereas measuring the distance on the S data slightly improves the results.

The second experiment is performed on the earthquake regression dataset a real world dataset [5] which is described in section 7.1.7. The label to predict is the so called PGA (Peak Ground Acceleration) value.

The achieved model for prediction is learned on the California data and applied on the Japan earthquake data. Again Gaussian kernels are used in the normal SVR with no

Table 7.3: Results for the earthquake dataset [5]. Weighted SVR significantly improves the prediction on the P data.

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

weights (uniform) and the weighted SVR method described in section 6.5. As in the previous experiments, the results are normalized 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 7.4.2. It turns out that learning a weighted SVR improves the prediction result on the Japan dataset, as shown by Table 7.3. 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. The 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.

7.5 Summary

This chapter introduced a new method for measuring and compensating the covariate shift. A new formulation for finding appropriate importance weights is derived by using a Fourier approximation of the divergence measure between the P distribution and the reweighted S 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. 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. An individual treatment of each dimension might improve the method further. 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 [35]. Such an approach would allow the treatment of even higher dimensional problems.

Chapter 8

Inductive Transfer Learning (ITL)

The following chapter contains and extends parts of [34].

This chapter investigates methods for compensating a dataset shift in both the covariates x and the dependent variables y. Such a situation is known as Inductive Transfer Learning and was introduced in section 6.6 and 5.2.2. Motivated by the concept of importance sampling, two approaches are investigated for improving regression in the ITL setting by assigning each instance in the S data a weight. The first one called Direct ITL (DITL) is a supervised and the second Kullback-Leibler ITL (KLITL) an unsupervised method. The resulting weights are then used in a modified ridge regression (explained in section 6.6.1) in combination with the S and P data in order to improve the prediction quality of the P data. Experiments show that both new approaches yield good results.

Section 6.6 presents some instance based approaches but additionally other numerous approaches have been developed. Kernel based ideas have been presented by [83] and [17], where a special kernel matrix is learned that reflects the similarities between the S and P data. A further method is given in [75], in which an informative prior is constructed from the S data in order to improve a model on the P data. An additional advance is feature representation transfer [6]. This method learns a projection of the S and P data onto a lower dimensional subspace such that the common or shared information of both the P and S data can be used for the model on the P data. Learning feature representation is in particular common in the domain of natural language processing (NLP). Differences in vocabulary and writing style imply a bias in distribution such that normal learning approaches tend to perform worse in different domains. In this area, [29] proposed a simple, but often well performing, kernel-mapping function for NLP problems, which maps the data from both source and target domains to a high-dimensional feature space, where standard discriminative learning methods are used.

Model transfer or hypothesis transfer learning comprise another class of approaches for treating ITL. In the model transfer setting, a model parameter θ_S is learned on the S data. Assuming that the models should be similar, the idea is to regularize the model parameter for the P data θ_P with the help of the parameter θ_S. Recent work on this topic is given by [53] and [96].

8.1 Problem Formulation

For inductive transfer learning, it is now a situation assumed where the two datasets (X^S, Y^S), the S data, and (X^P, Y^P), the P data, are given by:

$$(X^S, Y^S) \sim p^S(x, y) \quad \text{and} \quad (X^P, Y^P) \sim p^P(x, y).$$

Further, the number of P data is assumed to be much smaller than the number of S data, and the two distributions from which the data was sampled are not equal, i.e. $p^P(x, y) \neq p^S(x, y)$. Nevertheless, a further assumption is that the two datasets are somehow related to each other, so that in some parts of the domain the distributions are similar (or even equal), i.e.:

$$p^S(\tilde{x}, \tilde{y}) \approx p^P(\tilde{x}, \tilde{y}) \text{ for some } (\tilde{x}, \tilde{y}).$$

Therefore, one can employ data points from the S data to improve the prediction on the P data. By assumption, having $p^P(x, y) \neq p^S(x, y)$ implies that the S and P data cannot be simply combined. The crucial part is to determine points from the S data that contribute positively to the P data prediction and neglect points that have a negative influence. A solution to this problem is based on a measure of similarity between the two distributions. A common way to achieve this is importance sampling (chapter 2). Defining the importance weight function as $w(x, y) := \frac{p^P(x,y)}{p^S(x,y)}$ one could reweight the S data distribution by:

$$p^P(x, y) = w(x, y)p^S(x, y) = \frac{p^P(x, y)}{p^S(x, y)}p^S(x, y). \tag{8.1}$$

With the help of the function $w(x, y)$ it becomes possible to assign each S datapoint (x^S, y^S) an individual and appropriate weight. A weight close to one indicates a preferable point, while a weight far from one indicates the opposite. Hence this approach seems suitable for tackling the induction transfer learning setting. However, this definition of the importance function requires knowledge of both distributions, which is not available. Therefore, an approximation of the importance function $w(x, y)$ is needed instead. By employing an appropriate approximation, the idea of importance sampling offers a guideline for solving the task of ITL.

8.2 New Instance Based Approach

8.2.1 Reweighting of the Prediction Function

By assuming that some given data $(X, Y) \subset \mathbb{R}^{L+1\times D}, L \in \mathbb{N}$ is distributed according to an (unknown) distribution $p(x, y)$, it can be expressed differently by applying the chain rule:

$$p(x, y) = p(y|x)p(x) \quad \text{or} \quad p(x, y) = p(x|y)p(y).$$

Although the suggested method can be applied to both cases, the discriminative (left) and the generative one (right), the following will refer to the first equation for the discriminative approach. Predictions are obtained by:

$$\hat{y}^* = \text{argmax}_y \left(p(y|x^*)p(x^*)\right).$$

By assumption, the new data x^* and its corresponding (unknown) label y^* is distributed according to $p(x, y)$, and therefore, loosely speaking, the best prediction one can make is the y with the highest probability given the data x^*.

However, in the setting of inductive transfer learning, two different distributions are given which for some primal data point $(x^P, y^P) \sim p^P$ yields the following two expressions for the prediction of y^P :

$$\begin{aligned} y^P_P &= \text{argmax}_y \left(p^P(y|x^P)p^P(x^P)\right) \\ y^P_S &= \text{argmax}_y \left(p^S(y|x^P)p^S(x^P)\right). \end{aligned}$$

In general, the prediction of y^P based on p^S for the S data, namely y^P_S, can differ arbitrarily from the prediction y^P_P based on the P distribution. Therefore, in order to make better predictions for the P data using the distribution of the S data, the S distribution will now be reweighted as suggested in (8.1):

$$\begin{aligned} y^P &= \text{argmax}_y \left(p^P(y|x^P)p^P(x^P)\right) \\ &= \text{argmax}_y \left(\frac{p^P(y|x^P)p^P(x^P)}{p^S(x^P, y)} p^S(y|x^P)p^S(x^P)\right) \\ &= \text{argmax}_y \left(w(x^P, y)p^S(y|x^P)p^S(x^P)\right). \end{aligned} \tag{8.2}$$

From this derivation one can see that this also is an unbiased estimator for the P data.

8.2.2 Estimation of the Weight Function

Due to the lack of knowledge about the true distributions p^P and p^S one cannot obtain the correct importance function directly. Instead the model $\hat{w}(x, y)$ as given in 6.1 is used for the inference of an appropriate approximation. To determine suitable weights $\hat{w}$ two approaches for the estimation will be introduced. The first one will be referred to as the direct method or DITL (Direct ITL) because it will directly rely on the prediction performance of the model learned on the S data. The goal of the model is to minimize the prediction error, i.e.

$$\min ||Y^P - \hat{Y}^P||^2$$

where Y^P is the vector of the real labels $\{y_i\}_{i=1,\ldots,M}$ and $\hat{Y}^P$ the model predictions. Therefore, by following this approach, and with the help of expression (8.2), an optimization problem for the estimation of a weight function can be stated as:

$$\min_{\hat{w}} \sum_{i=1}^{M} \left(y_i^P - \operatorname{argmax}_y \left(\hat{w}(x_i^P, y) p^S(y|x_i^P) p^S(x_i^P)\right)\right)^2 .$$

The idea behind this approach is that the computation of the weights $\hat{w}$ is performed with respect to the known labels Y^P in the context of a reweighted model for the S data. Therefore this approach provides a supervised method for adjusting the weights $\hat{w}$. Since for a given point x^P the argmax does not depend on $p^S(x^P)$ that term can be omitted, which leads to:

$$\min_{\hat{w}} \sum_{i=1}^{M} \left(y_i^P - \operatorname{argmax}_y \left(\hat{w}(x_i^P, y) p^S(y|x_i^P)\right)\right)^2 . \tag{8.3}$$

Additionally, a second method is proposed which does not depend directly on prediction models and can be regarded as an unsupervised approach. Following the idea of [91] the Kullback-Leibler divergence will be minimized between two distributions. This straightforwardly extends the approach [91] for covariate shift by also taking the labels into account:

$$\begin{aligned} &\text{argmin}_{\hat{w}}\text{KL}(p^P(x,y)||\hat{w}(x,y)p^S(x,y)) \\ = &\text{argmin}_{\hat{w}}\left(\int p^P(x,y)\log\left(\frac{p^P(x,y)}{\hat{w}(x,y)p^S(x,y)}\right)dxdy\right) \\ = &\text{argmin}_{\hat{w}}\left(-\int p^P(x,y)\log\left(\hat{w}(x,y)\right)dxdy\right). \end{aligned}$$

Here, the dependence on the S data lies in the approximation $\hat{w}$ which will use the S data as the centerpoints. The last expression can be approximated by the empirical mean:

$$\Rightarrow \min_{\hat{w}} \sum_{i=1}^{M} -\log\left(\hat{w}(x_i^P, y_i^P)\right). \tag{8.4}$$

Additionally, the following constraint is obtained:

$$\begin{aligned} p^P(x,y) &= w(x,y)p^S(x,y) \\ \Rightarrow 1 = \int p^P(x,y)dxdy &= \int w(x,y)p^S(x,y)dxdy \\ \Rightarrow N &= \sum_{j=1}^{N} \hat{w}(x_j^S, y_j^S). \end{aligned} \tag{8.5}$$

As for the Fourier setting in section 7.1.4, this constraint ensures that the transformed p^S is again a density. This approach will be refered to as the indirect method or KLITL (Kullback-Leibler ITL).

8.3 Determination of Individual Weights

The following sections will describe methods for computationally obtaining weights for both approaches.

8.3.1 Weight Function

As for the covariate shift case, this approach will apply the approximation for the importance weight function introduced in section 6.1. However, here in addition the labels will be explicitly included into the formula since the source component shift is not restricted to the covariates, i.e.:

$$\hat{w}^{\alpha}(x,y) = \sum_{j=1}^{N} \alpha_j \exp\left(-\frac{||(x,y)-(x'_j,y'_j)||^2}{2\sigma^2}\right). \tag{8.6}$$

Here, the index α is added to the notation in order to emphasize the dependence on the αs.

8.3.2 Direct Approach (DITL)

Following the abstract modelling of a prediction function in a standard machine learning setting, one obtains for the discriminative case:

$$\hat{y}^* = \text{argmax}_y p(y|x^*). \tag{8.7}$$

Here, x^* denotes a data point to be predicted on, and $\hat{y}^*$ the prediction. For (8.7) one needs a concrete model $f(x)$ that can actually be calculated. Such a f can be the prediction function of the kernel ridge regression. The derivation of f is similar derivation for the weighted kernel ridge regression (6.7) and is given by:

$$\text{argmax}_y p(y|x^*) \approx f(x^*) = a^t \Bbbk(x^*), \tag{8.8}$$

where $\Bbbk(x^*) := (\Bbbk(x_1,x^*),\ldots,\Bbbk(x_L,x^*))^t$ is the kernel map of the new datapoint x^* and the data $X \subset \mathbb{R}^{L\times D}$ on which the model has been learned, with $\Bbbk(x_l,x^*) := \phi(x_l)^t\phi(x^*)$, and $a \in \mathbb{R}^L$ is the vector of coefficients for the linear combination in the feature space. Hence for (8.3) one needs a different mathematical approximation:

$$\text{argmax}_y \left(\hat{w}(x^*,y)p(y|x^*)\right) \approx f_{\hat{w}(x^*,y)}(x^*) \tag{8.9}$$

where the model f now also depends on the weight function $\hat{w}$.

Derived from the kernel ridge regression approximation, a new weighted prediction model is now suggested. Considering the weighted kernel ridge regression problem:

$$J_W(\theta) = \frac{1}{2}\sum_{l=1}^{L} w_l \left(y_l - \theta^t\phi(x_l)\right)^2 + \frac{\lambda}{2}||\theta||^2 \tag{8.10}$$

where $\theta \in \mathbb{R}^D$ again denotes the model parameter, ϕ is the feature map and w_l is a weight coefficient for each data point x_l. By the process of dualization of the ridge regression [13], one gets the weighted prediction function as:

$$0 = \nabla J_W(\theta) \Leftrightarrow \theta = \sum_{l=1}^{L} \underbrace{w_l \left(-\frac{1}{\lambda}(y_l - \theta^t \phi(x_l)) \right)}_{=:\hat{a}_l} \phi(x_l).$$

Here, $\hat{a}_l = a_l w_l$ are the coefficients for the linear combination in the feature space. Analogously to (8.8), this prediction function can be taken as an approximation for the weighted prediction, i.e.:

$$\begin{aligned} \text{argmax}_y \left(\hat{w}(x^*, y) p(y|x^*) \right) &\approx f_{\hat{w}(x^*,y)}(x^*) \\ &= a^t \hat{W}(x^*, y) \Bbbk(x^*) \end{aligned} \tag{8.11}$$

where, as in (8.8) $\Bbbk(x^*) := (\Bbbk_1(x^*), \ldots, \Bbbk_L(x^*))^t$ with $\Bbbk_l(x^*), l \in \{1, \ldots, L\}$ being a compact notation for $\Bbbk_l(x^*) = \Bbbk(x_l, x^*) := \phi(x_l)^t \phi(x^*)$ and $\hat{W}$ denotes a diagonal matrix where each entry is a weight function $\hat{w}$ as given in (8.6). The centerpoints $(x'_j, y'_j)_{j=1}^N$ will be set to the S data points. The reason for this choice is that in (8.12) one optimizes over the P data; using the P data as centerpoints would exhibit a higher risk of overfitting.

Obviously, this prediction function contains the label that is to be predicted. Therefore, label prediction for new data points is not possible with (8.11). However, this model is actually not intended for making predictions; rather one would like to estimate appropriate weights for the subsequent step, in which the weights are applied to learn a model on the P data combined with the weighted S data. (8.3) provides a framework for getting the best possible weights by conditioning the expression to the labels of the P data. Inserting (8.11) into (8.3) yields:

$$\begin{aligned} &\min_{\hat{W}} \sum_{m=1}^{M} \left(y_m^P - a^t \hat{W}(x_m^P, y_m^P) \Bbbk(x_m^P) \right)^2 \\ = &\min_{\hat{w}} \sum_{m=1}^{M} \left(y_m^P - \sum_{l=1}^{L} a_l \hat{w}(x_m^P, y_m^P) \Bbbk_l(x_m^P) \right)^2 . \end{aligned} \tag{8.12}$$

In the latter expression, the diagonal matrix is $\hat{W} := \text{diag}\left(\hat{w}(x_m^P, y_m^P), \ldots, \hat{w}(x_m^P, y_m^P)\right)$ where each diagonal entry corresponds to a w_l as given in (8.6) and L kernel functions $\Bbbk_1(x_m^P), \ldots, \Bbbk_L(x_m^P)$ are taken into account. Now, by making the approximation (8.6) one replaces the $x_1, \ldots, x_L$ by the S data. Then the resulting weight function depends only on the given set of αs and optimization is straight forward by optimizing w.r.t. the αs. However, in some experiments an unregularized version of (8.12) sometimes returns αs that are comprised of only one or very few elements that dominate. In order to account for such an overfitting, a regularization term is added to (8.12) which penalizes large coefficients:

$$\begin{aligned}&\min_{\alpha\geq 0}\sum_{m=1}^{M}\Big(y_m^P - a^t\hat{W}^{\alpha}(x_m^P, y_m^P)k(x_m^P)\Big)^2 + \gamma||\alpha||^2 \qquad (8.13)\\ =&\min_{\alpha\geq 0}\sum_{m=1}^{M}\left(y_m^P - \sum_{n=1}^{N}a_n\hat{w}^{\alpha}(x_m^P, y_m^P)k_n(x_m^P)\right)^2 + \gamma||\alpha||^2.\end{aligned}$$

The final expression (8.13) shows that one is now dealing with a weighted L^2 regression problem, since the $\hat{w}$ function can be moved outside the parenthesis which yields:

$$\min_{\alpha\geq 0}\sum_{m=1}^{M}\left(y_m^P - \hat{w}^{\alpha}(x_m^P, y_m^P)a^t k(x_m^P)\right)^2 + \gamma||\alpha||^2. \qquad (8.14)$$

The estimated αs will then be subsequently used in the weights for the actual ITL-KRR. Expression (8.14) can also be interpreted from a different point of view. In (8.14) the datapoints x_m^P are fixed, but also, the model vector a of the S model is fixed. On the other hand, the optimization is performed over the α's. By looking at and altering expression $\hat{w}^{\alpha}(x_m^P, y_m^P)$ one gets:

$$\hat{w}^{\alpha}(x_m^P, y_m^P) = \sum_{j=1}^{N}\alpha_j e^{-\frac{||(x_m^P, y_m^P)-(x'_j, y'_j)||^2}{2\sigma^2}} = \alpha^t k(x_m^P, y_m^P).$$

Here, k replaces the Gauss kernel on the left side of the equation. This change in notation shows, that expression (8.14) can be, again, interpreted as a weighted kernel ridge regression. However, in this particular case, the weights are given by the factors $a^t k(x_m^P)$. Therefore, in this sense, (8.14) is again a weighted regression problem by itself.

Learning the weights and a better model from the combined P data and weighted S data requires a three step procedure. Problem (8.13) depends on a model of the S data for adjusting the αs. Therefore, the first step requires the inference of a model solely on the S data, which returns the coefficients a for the prediction function (8.11). With this a a solution to (8.13) has to be found which yields proper αs. These αs are then used in (6.5) for calculating the weight for each S data point. The procedure can be stated as:

1. Learn a model a for the normal kernel ridge regression using solely the S data and ignore any P data.
2. Use the coefficients vector a from step 1 to determine appropriate αs for the weight function (8.6) by using the weighted prediction model (8.11) and solve (8.13).
3. After having determined the αs in step 2, use these to calculate the weight for the application of the ITL-KRR (6.5). Use the resulting model to make predictions for new P data.

The optimization in step 2 is w.r.t. the α's which denote the coefficients. Since the sum in $\hat{w}$ is convex as well as the quadratic function $(\cdot)^2$ the optimization in step 2 is convex and therefore guarantees a single optimal solution. Good parameters in each step are estimated by performing standard cross-validation on the P data. We employ Gaussian kernels in the kernel ridge regression, therefore we need to estimate η (the bandwidth parameter for the kernel function) and λ in step 1 and 3 similarly to the two parameters γ and σ (bandwidth parameter of the weight function) in step 2.

8.3.3 Indirect Approach (KLITL)

In addition to the direct approach, a further procedure for the indirect approach can be derived. Following the derivation in section 8.2.2, using expression (8.4) as the objective and expression (8.5) as the constraint, the suggested method is:

1. Optimize the following problem with a standard solver for constraint problems:

$$\begin{aligned} &\max_{\alpha} \frac{1}{M} \sum_{i=1}^{M} \log\left(\hat{w}^{\alpha}(x_i^P, y_i^P)\right) \\ &\text{s.t. } N = \sum_{j=1}^{N} \hat{w}^{\alpha}(x_j^S, y_j^S) \text{ and } \alpha \geq 0. \end{aligned} \tag{8.15}$$

2. Use the αs from step 1 to compute the weights $\hat{w}$ of each S data point for the optimization of the ITL-KRR (6.5). Use the resulting model to make predictions for new P data.

Here, the same representation of the weight function (8.6) is used as for the direct approach. For the estimation of a good σ in (8.15) a modified version of cross-validation is applied that is explained in the experimental section 8.5.1.

8.3.4 Comparison of the Direct and Indirect Approach

Comparing the two approaches, an advantage for the indirect approach is that it does not require the estimation of a model on the S data. This might be advantageous when a lot of S data is available. Additionally, the method requires the estimation of just one parameter σ for the kernel width used in the weight function. However, on the downside is the fact that this is an unsupervised method. By this, a method is meant that does not consider an objective cost function for the parameter inference. Therefore it is less likely to obtain robust or reliable estimations for α. On the other hand DITL applies a supervised optimization problem that takes a subset of the target labels in order to assess the quality of parameter inference. As mentioned further in section 8.3.2

the additional regularization term allows a higher control of the fitting process. As a consequence the DITL method is much more robust in compensating the dataset shift. The experimental section shows the conditions under which this becomes advantageous. The disadvantage are a higher calculation costs since it requires the calculation of an additional model on the S data and the parameters σ and γ.

8.4 Theoretical Analysis

8.4.1 RKHS Introduction

In order to investigate different properties of the ITL KRR it is necessary to introduce some further theoretical concepts. In particular, an introduction to the so called Reproducing Kernel Hilbert Spaces (RKHS) is required. The theory of RKHS has been developed by Nachman Aronszajn and Stefan Bergman [7]. Further analysis of the theory can be found in [102], [103] and [84]. The following section gives a brief overview of the theory of Reproducing Kernel Hilbert Spaces.

The section starts with the definition of a kernel k : A kernel is a function that maps two arguments into the real space, i.e. $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$, where $\mathcal{X}$ denotes an arbitrary set. It is symmetric $k(x, y) = k(y, x)$ and it is called positive semidefinite (psd) if:

$$\int_{\mathcal{X}\times\mathcal{X}} k(x, x')f(x)f(x')d\mu(x)d\mu(x') \geq 0$$

for any function $f \in \mathcal{L}^2(\mathcal{X}, \mu)$. Further there must exist a Hilbert space $\mathcal{H}$ and a map $\Phi : \mathcal{X} \mapsto \mathcal{H}$ such that for every $x, y \in \mathcal{H}$:

$$k(x, y) = \langle\Phi(x), \Phi(y)\rangle_{\mathcal{H}}.$$

This map $\Phi(\cdot)$ is also called feature map and the corresponding space $\mathcal{H}$ the feature space. Then the RKHS can be defined as follows [77]:

Definition (Reproducing Kernel Hilbert Space). Let $\mathcal{H}$ be a Hilbert space of real valued functions f on an arbitrary set $\mathcal{X}$ with an inner product given by $\langle\cdot, \cdot\rangle_{\mathcal{H}}$. The space $\mathcal{H}$ is called *Reproducing Kernel Hilbert Space* if the following two requirements are fulfilled:

- there exists a function $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ such that for every arbitrary but fixed $x \in \mathcal{X}$ the function $k(x, x')$ is a function of x' and belongs to $\mathcal{H}$ and
- k satisfies the reproducing property, which is defined by $\langle f(\cdot), k(\cdot, x)\rangle = f(x)$.

Note that for a given kernel function $\mathscr{k} \in \mathcal{H}$ the reproducing property implies: $\langle \mathscr{k}(x,\cdot), \mathscr{k}(\cdot, x') \rangle_{\mathcal{H}} = \mathscr{k}(x, x')$. The next theorem [7] gives insight on the relationship between a kernel $\mathscr{k}$ and RKHS.

Theorem (Moore-Aronszajn theorem). Given a set $\mathcal{X}$ and a symmetric, positive definite kernel $\mathscr{k}$ then there exists a unique Hilbert space $\mathcal{H}$ of functions f on $\mathcal{X}$ such that $\mathscr{k}$ is a reproducing kernel.

In other words the theorem states that a kernel is isomorph to a corresponding associated RKHS.

In order to better understand the reproducing property the so-called *Dirac* functional [32] can be considered. Given the delta function

$$\delta(x) := \begin{cases} +\infty & x = 0 \\ 0 & x \neq 0 \end{cases}$$

with $\int_{-\infty}^{\infty} \delta(x)dx = 1$ the *Dirac* functional can be derived by defining a distribution that satisfies:

$$\int_{-\infty}^{\infty} f(x)\delta(dx) = f(0).$$

Although the *Radon-Nikodym* derivative [62] does not exist, a convenient and frequently used but incorrect notation is:

$$\int_{-\infty}^{\infty} f(x)\delta(x)dx = f(0).$$

The *Riesz* representation theorem gives an idea on how the *Dirac* functional can be understood in terms of the reproducing property:

Theorem (Riesz Representation [80]). Given T a bounded linear functional on a Hilbert space $\mathcal{H}$, there exists a unique vector $v \in \mathcal{H}$ such that $T(f) = \langle f, v \rangle_{\mathcal{H}}$

This theorem can also be applied to the *Dirac* functional. Thus for each δ_x there exists a unique vector $\mathscr{k}_x \in \mathcal{H}$ such that $\delta_x(f) = f(x) = \langle f, \mathscr{k}_x \rangle_{\mathcal{H}}$.

It is possible to construct a RKHS by applying the so called *Mercer's* theorem [56]. The theorem applies the eigenfunctions of a given kernel $\mathscr{k}$, i.e. $\int \mathscr{k}(x, x')\phi(x')d\mu(x') = \lambda\phi(x')$ for all x where λ is the corresponding eigenvalue to the eigenfunction ϕ with respect to the given measure μ. Using the dot product notation this relation can be expressed as $\langle \mathscr{k}(x,\cdot), \phi \rangle_{\mathcal{H}} = \lambda\phi$. These preparations lead to a theorem that allows the characterization of a kernel $\mathscr{k}$ in terms of eigenvalues and eigenfunctions:

Theorem (Mercer's theorem). For a given psd kernel $\Bbbk$ and associated linear operator $T_{\Bbbk}(\varphi)(x) = \int_{\Omega} \Bbbk(x,y)\varphi(y)dy, \varphi \in \mathcal{L}^2(\Omega)$ there exists an infinite sequence of eigenfunctions $\{\phi_i\}_{i\in I}$, $I \subset \mathbb{N}$ and corresponding non-negative eigenvalues $\{\lambda_i\}_{i\in I}$ with $\lambda_i \geq \lambda_{i+1}$ of $\Bbbk$ such that the $\{\phi_i\}_{i\in I}$ are an orthonormal basis of $\mathcal{L}^2(\Omega)$. Then $\Bbbk$ has the representation:

$$\Bbbk(s,t) = \sum_{i\in I} \lambda_i \phi_i(s)\phi_i(t)$$

where convergence is absolute and uniform.

As a consequence one can express the feature map $\Phi(x)$ as:

$$\begin{aligned} \Phi : \mathcal{X} &\to \ell^2(I) \\ x &\mapsto \left\{\sqrt{\lambda_i}\phi_i(x)\right\}_{i\in I} \end{aligned}$$

Thus one gets:

$$\Bbbk(x,y) = \langle \Phi(x), \Phi(y)\rangle_{\mathcal{H}} = \left\langle \sqrt{\lambda_i}\phi_i(x), \sqrt{\lambda_i}\phi_i(y)\right\rangle_{\ell^2(I)}.$$

With this, a RKHS can be constructed by applying the eigenfunctions ϕ_i of the integral operator $T_{\Bbbk}$.

Theorem. Let $\mathcal{X}$ be a compact metric space and $\Bbbk : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ a continuous kernel with the representation $\Bbbk(x,y) = \sum_{i\in I} \lambda_i \phi_i(x)\phi_i(y)$ according to *Mercer's* theorem, $\mathcal{H}_{\Bbbk}$ the corresponding RKHS (by Moore-Aronszajn theorem) and define:

$$\mathcal{H} := \left\{ f(x) = \sum_{i\in I} f_i\phi_i(x) : \left\{\frac{f_i}{\sqrt{\lambda_i}}\right\} \in \ell^2(I) \right\}$$

with inner product given by:

$$\langle f, g\rangle_{\mathcal{H}} = \left\langle \sum_{i\in I} f_i\phi_i(x), \sum_{i\in I} g_i\phi_i(x) \right\rangle_{\mathcal{H}} = \sum_{i\in I} \frac{f_i g_i}{\lambda_i}.$$

Then $\mathcal{H} = \mathcal{H}_k$.

Proof: It is easy to verify that the inner product meets all the necessary requirements. Thus, $\mathcal{H}$ is a Hilbert space. Due to *Mercer's* theorem the decomposition $\Bbbk(\cdot, x) = \sum_{i\in I} \lambda_i\phi_i(x)\phi_i(\cdot)$ is given, and therefore:

$$\sum_{i\in I} \left| \frac{\lambda_i\phi_i(x)}{\sqrt{\lambda_i}} \right|^2 = \sum_{i\in I} \lambda_i\phi_i(x)\phi_i(x) = \Bbbk(x,x) < \infty.$$

Hence, $k(\cdot, x) \in \mathcal{H}, \forall x \in \mathcal{X}$. Set $f(\cdot) = \sum_{i\in I} a_i\phi_i(\cdot) \in \mathcal{H}$ with $\left\{\frac{a_i}{\sqrt{\lambda_i}}\right\} \in \ell^2(I)$ then:

$$\langle f(\cdot), k(\cdot, x)\rangle_{\mathcal{H}} = \left\langle \sum_{i\in I} a_i\phi_i(\cdot), \sum_{i\in I}(\lambda_i\phi_i(x))\phi_i(\cdot)\right\rangle_{\mathcal{H}} = \sum_{i\in I} \frac{a_i\lambda_i\phi_i(x)}{\lambda_i} = f(x).$$

Thus, $\mathcal{H}$ is a Hilbert space of functions that has a reproducing kernel k. By the uniqueness of the RKHS this must be the $\mathcal{H}_k$. This concludes the proof.

As a consequence, the norm $||f||^2_{\mathcal{H}}$ is given by $\sum_{i\in I} \frac{f_i^2}{\lambda_i}$. Briefly summarized, the idea for the constuction of a RKHS for given psd kernel k follows from the following chain of implications:

$$k \xrightarrow{\text{Mercer}} \{\lambda_i\}_{i\in I}, \{\phi_i\}_{i\in I} \xrightarrow{\text{Eigenfunctions}} \text{dot product} \Longrightarrow \text{RKHS} \xrightarrow{\text{Moore-Aronszajn}} k.$$

8.4.2 More data implies better model

Based on the properties of the RKHS in the last section one can show some theoretical properties of the ITL-KRR. The following entities will be used: firstly, the regression function $\Psi(x)$ which will be defined by:

$$\Psi(x) = \mathbb{E}_{p^P}[y|x] = \int_{\mathcal{Y}} y dp^P(y|x), \quad \mathcal{Y} \subset \mathbb{R} \tag{8.16}$$

the exact function w.r.t. the distribution p^P. Here, the term exact function means, that this is the exact functional relationship (expressed in terms of an expectation) one wants to infer by solving the regression problem. However, this regression function Ψ depends on the unknown distribution p^P and the model is inferred based on the sampled (and perturbed) y values from this function. Important to note is that the conditional distribution of the ys given the xs is considered and therefore, the labels are implicitly contained in the regression problem. The next entities needed are empirical labels $y^P \in \mathbb{R}^M$ and $y^S \in \mathbb{R}^N$ two vectors of perturbed samples from the real (unknown) function and $\hat{f}$, the actual model, i.e. the ITL-KRR solution (based on the empirical data), which minimizes the following functional:

$$\hat{f} = \text{argmin}_f \mathcal{J}[f] = \frac{1}{2}\sum_{i=1}^{M}\left(f(x_i^P) - y_i^P\right)^2 + \frac{1}{2}\sum_{i=1}^{N} w(x_i^S, y_i^S)\left(y_i^S - f(x_i^S)\right)^2 + \frac{\lambda}{2}||f||^2_{\mathcal{H}}. \tag{8.17}$$

Here, the data comes from the P and S data, i.e. $x_i^P, y_i^P \in \left\{X^P, Y^P\right\}$ and $x_i^S, y_i^S \in \left\{X^S, Y^S\right\}$ and each data pair (x_i^S, y_i^S) gets a weight factor $w(x_i^S, y_i^S)$ assigned. In the following, it is assumed that one has given the exact importance function, i.e. $w(x,y) = \frac{p^P(x,y)}{p^S(x,y)}$.

Since by assumption the given P data is limited but a large S data is available, a question to ask could be: What happens to the performance of the model if more and more S data is taken into account? Or expressed differently: What happens if the number of data $N \to \infty$? To answer this question, one needs to consider the expected error made w.r.t. the p^P distribution, i.e.:

$$\begin{aligned}\mathbb{E}_{p^P}\left[\frac{1}{N}\sum_{i=1}^{N}\left(y_i^S - \hat{f}(x_i^S)\right)^2\right] =& \mathbb{E}_{p^S}\left[\frac{1}{N}\sum_{i=1}^{N} w(x_i^S, y_i^S)\left(y_i^S - \hat{f}(x_i^S)\right)^2\right] \\ =& \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_{p^S}\left[w(x_i^S, y_i^S)\left(y_i^S - \hat{f}(x_i^S)\right)^2\right] \\ =& \int w(x,y)\left(y - \hat{f}(x)\right)^2 dp^S(x,y). \end{aligned} \tag{8.18}$$

The latter equality holds since, each x_i^S is *also* a random variable distributed according to p^S such that one gets N times the same expectation. To see this, consider, that the particular samples are sampled from a random variable that is distributed according to p^S. Similarly, the error on the P data:

$$\mathbb{E}_{p^P}\left[\frac{1}{M}\sum_{i=1}^{M}\left(y_i^P - \hat{f}(x_i^P)\right)^2\right] = \int \left(y - \hat{f}(x)\right)^2 dp^P(x,y). \tag{8.19}$$

By applying the regression function $\Psi(x)$ (8.16) w.r.t the distribution p^P and further setting $y - \hat{f} = (y - \Psi) + (\Psi - \hat{f})$ one gets for expression (8.18):

$$\begin{aligned}\int w(x,y)\left(y - \hat{f}(x)\right)^2 dp^S(x,y) = & \int w(x,y)\left(y - \Psi(x)\right)^2 dp^S(x,y) + \\ & \int w(x,y)\left(\Psi(x) - \hat{f}(x)\right)^2 dp^S(x,y) + \\ & \int w(x,y)\left(y - \Psi(x)\right)\left(\Psi(x) - \hat{f}(x)\right) dp^S(x,y)\end{aligned}$$

Due to the definition of Ψ and that the integral cancels out the perturbations in the ys, the ys match the regression function Ψ and the cross term vanishes, i.e.:

$$\begin{aligned}
&\int w(x,y)\,(y-\Psi(x))\left(\Psi(x)-\hat{f}(x)\right)dp^S(x,y)\\
=&\int (y-\Psi(x))\left(\Psi(x)-\hat{f}(x)\right)dp^P(x,y)\\
=&\int \left(y\Psi(x)-y\hat{f}(x)-\Psi(x)^2+\Psi(x)\hat{f}(x)\right)dp^P(x,y)\\
=&\int \left(\Psi(x)\underbrace{(y-\Psi(x))}_{=0}\right)dp^P(x,y)+\int \left(\hat{f}(x)\underbrace{(y-\Psi(x))}_{=0}\right)dp^P(x,y)=0.
\end{aligned}$$

Further, the first term:

$$\int w(x,y)\,(y-\Psi(x))^2\,dp^S(x,y)$$

is constant since it does not depend on the model $\hat{f}$ and can therefore be omitted. The remaining term is:

$$\int w(x,y)\left(\Psi(x)-\hat{f}(x)\right)^2 dp^S(x,y)=\int \left(\Psi(x)-\hat{f}(x)\right)^2 dp^P(x,y) \tag{8.20}$$

This expression denotes the expected error between the model and the regression function and can be considered as an alternative expression for the difference of the observed labels y and the model $\hat{f}$. On the other side, this expression is fully specified in terms of a distribution p which is required for the next analytical steps.

The latter expression can be identified as the expected error made by the first two empirical terms in the ITL-KRR (8.17) by considering an expression for each of those two terms. Part one is given by:

$$\frac{1}{2}\sum_{i=1}^{M}\left(f(x_i^P)-y_i^P\right)^2\approx\frac{M}{2}\int\left(\Psi(x)-\hat{f}(x)\right)^2 dp^P(x,y).$$

Analogously, for the second term one gets:

$$\frac{1}{2}\sum_{i=1}^{N}w(x_i^S,y_i^S)\left(y_i^S-f(x_i^S)\right)^2\approx\frac{N}{2}\int w(x,y)\left(\Psi(x)-\hat{f}(x)\right)^2 dp^S(x,y). \tag{8.21}$$

By combining these two expressions one gets an analytical model for (8.17) except for the regularization term. The regularization in expression (8.17) denotes a Tikhonov regularization [95]. This type of regularization restrains the limits of freedom of the model by reducing the variance of the model vector θ. From a Bayesian point of view, this means that the error of the given data is normally distributed with zero mean and variance η^2, i.e. $y_i = x_i + \epsilon_i$ with $\epsilon_i \sim \mathcal{N}(0, \eta^2)$, $\forall i$. Thus, the regularization term can be interpreted as the inverse noise variance or expressed differently, given the data and the a priori distribution of the data the solution of the ridge regression problem (8.17) is, according to Bayes' theorem, the most probable solution [18]. Therefore, following this idea, the regularization term can be simply added in order to get an analytical model that corresponds to the empirical model. As the following proof shows, this has no impact on the convergence of the method such that it can be safely added here:

$$\mathcal{J}[\hat{f}] = \frac{M}{2}\int \left(\Psi(x) - \hat{f}(x)\right)^2 dp^P(x,y) + \frac{N}{2}\int w(x,y) \left(\Psi(x) - \hat{f}(x)\right)^2 dp^S(x,y) + \frac{1}{2}||\hat{f}||^2_{\mathcal{H}}.$$

This expression can now be used to determine the behavior of the approximation if more data is taken into account. Therefore, given now a psd kernel k, then thanks to *Mercer's* theorem one gets an orthonormal basis of eigenfunctions $\phi_i(x)$ (w.r.t. probability measure p^P) that span the RKHS. Thus one can represent the functions Ψ and $\hat{f}$ by a linear combination of these eigenfunctions, i.e. $\Psi(x) = \sum_{i=1}^{\infty} \psi_i \phi_i(x)$ and $\hat{f} = \sum_{i=1}^{\infty} \hat{f}_i \phi_i(x)$. Further, from the reproducing property of a RKHS, for $||f||^2_{\mathcal{H}}$ one gets:

$$\langle \hat{f}(\cdot), k(\cdot, x)\rangle_{\mathcal{H}} = \sum_{i=1}^{\infty} \frac{\hat{f}_i \lambda_i \phi_i(x)}{\lambda_i} = \hat{f}(x) \Rightarrow ||\hat{f}||^2_{\mathcal{H}} = \sum_{i=1}^{\infty} \frac{\hat{f}_i^2}{\lambda_i} < \infty \tag{8.22}$$

where the λ_is are the eigenvalues of the kernel given in *Mercer's* theorem. Plugging this into the equation above one gets:

$$\begin{aligned} \mathcal{J}[\hat{f}] \approx\ & \frac{M}{2}\int \left(\sum_{i=1}^{\infty} \psi_i \phi_i(x) - \sum_{i=1}^{\infty} \hat{f}_i \phi_i(x) \right)^2 dp^P(x,y) + \\ & \frac{N}{2}\int w(x,y) \left(\sum_{i=1}^{\infty} \psi_i \phi_i(x) - \sum_{i=1}^{\infty} \hat{f}_i \phi_i(x) \right)^2 dp^S(x,y) + \frac{1}{2}||\hat{f}||^2_{\mathcal{H}} \end{aligned} \tag{8.23}$$

The first term can be transformed in the following way:

$$\begin{aligned} & \frac{M}{2}\int \left(\sum_{i=1}^{\infty} \psi_i \phi_i(x) - \sum_{i=1}^{\infty} \hat{f}_i \phi_i(x) \right)^2 dp^P(x,y) \\ =& \frac{M}{2}\int \left(\sum_{i=1}^{\infty} \left(\psi_i - \hat{f}_i\right) \phi_i(x) \right)^2 dp^P(x,y) \\ =& \frac{M}{2}\sum_{i=1}^{\infty} \left(\psi_i - \hat{f}_i\right)^2 \int \left(\phi_i(x)\right)^2 dp^P(x,y) \end{aligned}$$

Since the convergence of the sum is uniform, the integral and sum can be interchanged. Due to *Mercer's* theorem the ϕ's comprise an orthonormal basis of eigenfunctions such that the term $\int \left(\phi_i(x)\right)^2 dp^P(x,y) = 1$. By applying the same steps to the second term of expression (8.23) one gets:

$$\begin{aligned}\mathcal{J}[\hat{f}] \approx\ & \frac{M}{2}\sum_{i=1}^{\infty}\left(\psi_i - \hat{f}_i\right)^2 \int (\phi_i(x))^2\, dp^P(x,y) + \\ & \frac{N}{2}\sum_{i=1}^{\infty}\left(\psi_i - \hat{f}_i\right)^2 \underbrace{\int w(x,y)\,(\phi_i(x))^2\, dp^S(x,y)}_{=1 \text{ since } wp^S = p^P} + \frac{1}{2}\sum_{i=1}^{\infty}\frac{\hat{f}_i^2}{\lambda_i} \\ =\ & \frac{N+M}{2}\sum_{i=1}^{\infty}\left(\psi_i - \hat{f}_i\right)^2 + \frac{1}{2}\sum_{i=1}^{\infty}\frac{\hat{f}_i^2}{\lambda_i}.\end{aligned}$$

The integral and sum expressions can be swapped due to the result of *Mercer's* theorem which states uniform convergence. Due to the transforming importance function w, the orthogonality of the ϕ's in the scalarproduct given by the integral w.r.t. p^P translates to the p^S term. The minimum of this expression is obtained by taking the derivative w.r.t. $\hat{f}_i$ and setting it to zero. Hence one gets:

$$\frac{\hat{f}_i}{\lambda_i} - (N+M)\left(\psi_i - \hat{f}_i\right) = 0 \Leftrightarrow \hat{f}_i = \frac{\lambda_i}{\lambda_i + \frac{1}{N+M}}\psi_i \xrightarrow[N\to\infty]{} \psi_i. \tag{8.24}$$

This proof is an extension of the convergence of a Gaussian process [77] such that it considers P & S data. Thus, the consequences on the convergence are essentially the same as in the Gaussian process setting. From the last expression (8.24), one can see that as more and more data, either from the P data or the S data, is taken into account the coefficients for the model function $\hat{f}$ converge to the coefficients of the real function Ψ. Another thing that can be derived is that the regularization term (or prior in term of a Bayesian viewpoint) looses influence as more and more data is coming in. From the proof, one might get the impression that it would suffice to just add more S data such that new P data could be ignored completely. This is the case if the real importance function $w(x,y)$ is known which is required here, to state a valid proof. However, in the practical setting, where one does not know the exact w one has to rely on additional P data.

8.5 Experiments

In the experimental section, the performance of the direct (DITL) and indirect (KLITL) approaches versus the boosting for transfer learning method , another instance-weighted approach, described in [66] will be investigated. Further, a method called "Frustratingly Easy Domain Adaptation" by [29] is applied, a simple, but often well performing feature learning approach, in combination with kernel ridge regression (in the following referred to as FS-KRR). Two more methods that both are based on Gaussian process

(GP) regression are considered for comparison. The first one, referred to as ATL [17], calculates a special correlation matrix for the GP and the second, called SMTR [111], is a GP-based multi-task method where the common knowledge of all tasks is reflected in form of a shared prior. Additionally, the performance of a normal kernel ridge regression for regression problems learned from the three dataset combinations: P data, S data, and P & S data is provided. As a weighted baseline KLIEP [91], presented in chapter 7, is taken into account for determining instance weights, as an alternative Kernel Mean Matching (KMM) [45] is also employed.

This experimental section also consider the previously discussed earthquake data. However, in contrast to the covariate shift setting where the full dataset of Japan (without labels) was employed in the learning procedure, this section will assume a different situation. Here, it is assumed that only very few data from the Japan dataset including the labels is given. This is fundamentally different from the covariate shift setting where it is assumed that the full Japan dataset is available but without any label information. Yet, this dataset can still be applied as an example for the source component shift, as the experimental section will show. Therefore, if the data provided for Japan is scarce, the model learned on the Japan data might not provide a good prediction quality. Even if the distributions for the California data and Japan data differ in general, it is reasonable to assume that in some respects the distributions are very similar or almost equal. Therefore, it might be helpful to augment the Japan data with some data from the California data to improve prediction quality for the Japan dataset. For the earthquake example the data for California would be the S data, and the Japan data would be the P data.

8.5.1 Parameter Selection

DITL applies a kernel ridge regression (KRR), a weight estimation procedure and the ITL-KRR. In each of the three steps, a 5-fold standard cross-validation is performed for the parameter estimation. The KRR and the ITL-KRR will apply RBF kernel functions for the calculation of the kernel matrix $\mathcal{K}$. Denoting the bandwidth parameter of the RBF kernels with η two parameters have to be calculated η and the regularization parameter λ in step 1 (KRR) on the S data, and furthermore step 3 (ITL-KRR) on the S and P data. In the second step DITL requires the estimation of the parameters σ (the bandwidth for the importance function approximation) and γ (the regularization parameter for the α vector). Since all problems are quadratic, one can use standard algorithms for quadratic programming.

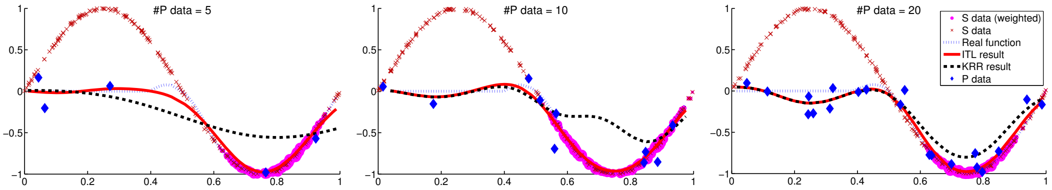


Figure 8.1: Illustrative toy example for DITL. From left to right the number of P data is: 5, 10 and 20 datapoints. The location of an S datapoint is marked by a red cross '×'. The round purple points indicate how much weight an S datapoint gets assigned. The thicker the point the more weight it has. As can be seen from the example, in one dimension 20 datapoints are already dense enough to learn a reliable kernel ridge regression.

KLITL is different in the parameter estimation from the DITL method. KLITL requires just two steps. In the first step, problem (8.15) is solved; i.e. a simple maximization of the sum constrained by the expectation equal to N.

In order to get a good estimate for σ a selection criteria is proposed that will choose the σ from all the proposed σ values that maximizes (8.15). Since KLITL in the first step is unsupervised, a similar method to cross-validation is used to get a more stable selection result. Given the original S dataset, $\left(X^S, Y^S\right)$, the dataset is splitted into five disjoint parts, $\left(X^S, Y^S\right)_{l=1}^{5}$. Each split $\left(X^S, Y^S\right)_l$ should contain enough samples of the S data but due to the assumption of a sufficiently large S dataset, this should not be a problem. Now for a fixed parameter σ expression (8.15) is maximized for each dataset combination $\{\left(X^S, Y^S\right)_l, \left(X^P, Y^P\right)\}$. The parameter with the highest mean of these five maximas is picked. Therefore. in this way, a more robust method for estimating an adequate parameter is obtained.

8.5.2 Datasets

First, for illustration purposes, by using a toy example it is shown how the proposed DITL algorithm learns weights, and how these weights influence the model prediction. The performance of the new methods is then verified on some standard benchmark datasets that have been slightly modified. Finally, the methods are applied to three real world datasets, the earthquake dataset that was previously used for the covariate shift methods, a second new one describing delays of aircrafts and a third new describing radio signal strengths from WiFi access points for indoor location estimation. However, in contrast to the covariate shift setting, here the earthquake dataset will also employ the labels and less P data is given.

8.5.2.1 Toy Examples

The toy example mainly serves as an illustrative demonstration of how and where the DITL algorithm learns weights for the S data, and shows the consequences for the prediction of the P data when taking additional S data into account. Similar results can be obtained by applying KLITL, but are not exposed due to redundancy.

The dataset is generated by sampling data points from two functions that are - as assumed for the methods - partially almost identical. The S data is sampled as:

$$f_s(x) = sin(2\pi x) + \eta_S \mathcal{N}(0,1), \tag{8.25}$$

where η_S is a factor for controlling the influence of the variance (in the experiments $\eta_S = 0.1$). The P data is sampled according to:

$$f_p(x) = \begin{cases} 0 + \eta_P \mathcal{N}(0,1) & 0 \leq x \leq 1/2 \\ \sin(2\pi x) + \eta_P \mathcal{N}(0,1) & 1/2 < x \leq 1 \end{cases} \tag{8.26}$$

where η_P, as in the case for the S data, is the sample variance (in the experiments $\eta_P = 0.4$). Parameter selection is performed as described in the previous section 8.5.1. The experiments in figure 8.1 only apply a very small number of P data points (just 5,10 and 20). The reason for this is that, for the example, the performance of a standard KRR is already very good at 20 data points. This is due to the fact that in one dimension one gets a non-sparse dataset very quickly. Since the aim of the example is the illustration that the lack of data points (as by assumption), and hence sparseness of data, leads to models that perform poorly on predicting new data, this setting for the toy example is reasonable. However, in high dimensions the situation is different and the number of P data points can be much larger, in parts due to the empty space phenomenon.

8.5.2.2 Benchmark Datasets

In this section DITL, KLITL, ATL, FS-KRR, SMTR and TL-boosting are applied to standard benchmark datasets. The experimental setup is as follows: The following standard benchmark datasets for evaluation are taken: abalone, elevators[4], and the kin family datasets[5]. From the kin dataset the so-called *n datasets* (n for nonlinear) with 8 dimensions are used. The *nm data* (non linear medium variance) is used as the S data and *nh data* (non linear high variance) as the P data. Since abalone and elevators do not necessarily comprise a dataset shift the S and P data is determined according to a

[4] abalone and elevators can be found on mldata.org

[5] kin datasets are part of the delve dataset repository

special selection criteria. The selection process is performed upfront and independently of the ITL method. In the first step, the covariates X are normalized to $[0,1]$ for each dimension. Then the following three values are calculated randomly; First, a dimension $d \in \{1,\dots,D\}$ is selected randomly. In the same way a threshold value $\vartheta \in [0,1]$ is chosen randomly and finally with a selection probability $p_{select} \in [0,1]$ it is sampled. All values are selected according to a uniform distribution on the corresponding domain. After that, these three values are fixed for the actual data generation process. For the dataset generation a data point (x,y) is selected from the set (X,Y), $x \in X$ is taken and then the value for dimension d, i.e. x_d, is considered. If x_d is larger than the threshold ϑ this (x,y) combination is added with probability p_{select} to the S data (X^S, Y^S), and to the P dataset (X^P, Y^P) otherwise. That way, 50 instances of the data sets are generated randomly for each individual experiment with a drift, i.e. a covariate shift, in the distribution. In order to get also a shift in the labels the function

$$f(y) = y + \nu \sin(2\pi y)\,,\ \nu \in [0,1] \tag{8.27}$$

is applied to the labels of the P data only. For instance $\nu = 0$ means no shift in the labels. This way a dataset is generated that accounts for the ITL setting and, due to the ν parameter, gives control about the strength of the shift such that S and P data still have something in common.

Tables 8.1 and 8.2 show the results for each method for a different number of P data. For illustration results with $\nu = 0$ are stated in table 8.1, i.e. with only a covariate shift. As one would expect, a standard KRR using both S and P data performs best, since for $\nu = 0$ the datasets only contain a covariate shift. Nevertheless, this experiment verifies that the introduced ITL methods learn proper weights in order to employ the right S data points for improving prediction of the P data. Their prediction performance is best over all approaches which aim to take a shift into account, both the covariate shift procedures and the full dataset shift procedures. Experimental results are qualitatively the same on each dataset.

Table 8.2 shows results when an artificial shift of $\nu > 0$ is added to the labels. By adding such a shift one gets a full dataset shift setting and the situation is as expected differently from the covariate shift setting. The KRR learned exclusively on the S data does not show any performance gain by adding P data. This is to be expected since the P data has no influence on the learning procedure but only serves as an evaluation dataset. On the other hand, if learned on $P \cup S$ the results improve slightly but they are still biased by the S data. Over all approaches, as the proportion of the P data grows the error gets reduced. FS-KRR and ATL show comparable errors, this can be explained by the similarity in these approaches, by construction both do not use weights for each instance

Table 8.1: Results on different benchmark datasets for mean square error. Sampling of the S and P data is explained in the text. Each experiment has been performed 50 times and the results have been normalized by the error on the P data. Therefore, each number in the other columns denotes the proportion in percent. Further comments on the results can be found in the text. Error calculation has been performed on a randomly sampled P_{eval} for each trial. Best results are marked as bold text.

Number of P data	# P50	# P100	# P 200	# P 300
Abalone $\nu = 0$ (no additional label shift), error on $\lvert P_{eval}\rvert = 1000$ and $\lvert S\rvert = 1000$				
KRR (on P)	0.0017 / 1.00	0.0016 / 1.00	0.0014 / 1.00	**0.0012 / 1.00**
KRR (on S)	0.88	0.94	0.98	1.03
KRR (on $S \cup P$)	**0.72**	**0.77**	**0.85**	**0.99**
FS-KRR	0.87	0.94	0.97	1.02
KMM	0.91	0.95	0.96	**1.00**
ATL	0.85	0.92	0.96	**1.00**
TL-boosting	0.83	0.90	0.98	**1.00**
KLIEP	0.90	0.94	0.98	**1.00**
KLITL	0.84	0.89	0.96	1.01
DITL	0.78	0.85	0.89	**0.99**
Elevators $\nu = 0$ (no additional label shift), error on $\lvert P_{eval}\rvert = 1000$ and $\lvert S\rvert = 2000$				
KRR (on P)	5.4e-6 / 1.00	5.2e-6 / 1.00	4.4e-6 / 1.00	**3.5e-6 / 1.00**
KRR (on S)	0.67	0.71	**0.80**	**0.99**
KRR (on $S \cup P$)	**0.66**	0.71	**0.79**	**1.00**
FS-KRR	0.76	0.81	0.93	1.02
KMM	0.91	0.95	0.98	1.01
ATL	0.77	0.80	0.91	1.01
TL-boosting	0.75	0.79	0.90	1.01
KLIEP	0.88	0.92	0.97	**1.00**
KLITL	0.73	0.74	0.86	**0.99**
DITL	**0.65**	**0.68**	0.81	**1.00**
kin dataset $\nu = 0$ (no additional label shift), error on $\lvert P_{eval}\rvert = 1000$ and $\lvert S\rvert = 2000$				
KRR (on P)	0.054 / 1.00	0.050 / 1.00	0.046 / 1.00	**0.042 / 1.00**
KRR (on S)	**0.90**	**0.92**	**0.96**	**1.00**
KRR (on $S \cup P$)	**0.91**	**0.92**	0.97	**1.00**
FS-KRR	**0.91**	0.95	0.97	1.03
KMM	0.96	0.97	1.00	1.02
ATL	0.92	0.94	0.98	**1.00**
TL-boosting	0.93	0.95	1.00	**1.00**
KLIEP	0.95	0.97	0.98	1.01
KLITL	0.93	0.94	0.97	1.01
DITL	0.92	**0.93**	0.97	**1.00**

but one weight for the correlation of P and S data. Consequently, each S data point has an equal influence. For KMM the $\frac{p^P(x,y)}{p^S(x,y)}$ is considered for the ratio calculation since that

Table 8.2: Results on different benchmark datasets for mean square error. The data has been augmented by adding an artificial shift to the labels. The other settings for these experiments are the same as in table 8.1. Best results are marked as bold text.

Number of P data	# P50	# P100	# P 200	# P 300
Abalone $\nu = 1/2$ (artificial label shift), error on $\lvert P_{eval} \rvert = 1000$ and $\lvert S \rvert = 1000$				
KRR (on P)	0.0024 / 1.00	0.0019 / 1.00	0.0016 / 1.00	0.0013 / 1.00
KRR (on S)	1.53	1.41	1.42	1.45
KRR (on $S \cup P$)	1.46	1.38	1.27	1.20
FS-KRR	0.92	0.93	0.96	1.01
KMM	0.93	0.96	1.01	1.00
ATL	0.89	0.91	0.94	**0.99**
TL-boosting	0.81	0.85	0.93	**0.99**
KLIEP	1.48	1.38	1.25	1.21
KLITL	0.80	0.87	0.92	1.00
DITL	**0.76**	**0.80**	**0.89**	**0.97**
Elevators $\nu = 1.0$ (artificial label shift), error on $\lvert P_{eval} \rvert = 1000$ and $\lvert S \rvert = 2000$				
KRR (on P)	6.5e-6 / 1.00	5.7e-6 / 1.00	4.1e-6 / 1.00	**3.6e-6 / 1.00**
KRR (on S)	1.61	1.51	1.42	1.49
KRR (on $S \cup P$)	1.51	1.40	1.38	1.29
FS-KRR	0.91	0.97	0.99	1.01
KMM	0.89	0.95	0.98	1.02
ATL	0.88	0.91	0.97	1.01
TL-boosting	0.74	0.78	0.94	**1.00**
KLIEP	1.53	1.45	1.35	1.30
KLITL	0.76	0.79	**0.90**	1.01
DITL	**0.68**	**0.71**	**0.89**	**0.99**
kin dataset $\nu = 1/4$ (artificial label shift), error on $\lvert P_{eval} \rvert = 1000$ and $\lvert S \rvert = 2000$				
KRR (on P)	0.065 / 1.00	0.056 / 1.00	0.050 / 1.00	**0.044 / 1.00**
KRR (on S)	1.30	1.34	1.32	1.30
KRR (on $S \cup P$)	1.28	1.23	1.19	1.15
FS-KRR	0.88	0.91	0.95	1.03
KMM	0.90	0.93	0.95	**1.00**
ATL	0.87	0.89	0.94	**1.00**
TL-boosting	0.83	0.88	**0.92**	**1.00**
KLIEP	1.27	1.24	1.18	1.12
KLITL	0.84	0.88	**0.91**	**1.00**
DITL	**0.79**	**0.84**	**0.91**	**1.00**

better fits the ITL setting. KMM does not provide a method for parameter selection, and it is unsupervised since it does not use a subset of the target labels to adjust the parameters, which overall makes it less robust and shows moderate performance. KLIEP used as a baseline covariate shift approach shows a poor performance, which is reasonable since it is not adapted to the ITL setting. The performance differences to the other methods show that it makes sense to treat ITL and covariate shift as two separate problem classes. Further, it demonstrates that one should be careful in the choice of the algorithm in the presence of a dataset shift. Other (related) methods for covariate shift [89] were also considered in the experiments, their performance is similar to KLIEP and therefore are not reported here in detail. TL-boosting and KLITL show a similar performance. DITL performs best, which might be due to the supervised way for estimating the weights. Nearly all methods eventually converge to a value of 1.00 because, as demonstrated by the toy example in section 8.5.2.1, with some data set size the P data provides enough information about its structure to allow a good prediction performance.

8.5.2.3 Real World Datasets

This section investigates the more interesting situation of real data that very likely contains a distribution shift. The first dataset [5] describes measurements taken during earthquakes in Japan and California. The features describe values that have the same measurements as in the experimental section of the covariate shift methods.

The second real world dataset describes the flight arrival and departure details for all commercial flights within the USA[6]. The complete dataset contains records from October 1987 to April 2008. Data from 2007 is taken as the S data and from 2008 data as the P data. In this case, one can argue that the measurement taken in 2008 are different to 2007 due to a shift in time. The predicted value is the delay of a particular flight.

The third dataset [109] comprises data for indoor location estimation from radio signal strengths received by a user device (like a PDA) from various WiFi Access Points. The measurements are taken at different locations and therefore contain a dataset shift. The results are shown in table 8.3. Besides FS-KRR and ATL all approaches which take a shift into account consistently improve the result in comparison to the baseline approach of KRR on P (and/or S).

Adjusting for a covariate shift with KLIEP only slightly improves the result, whereas approaches which also adjust with weights stemming from a dataset shift view achieve much better performance. The supervised approach DITL consistently performs best, with KLITL and TL-boosting as second.

In a final experiment (see table 8.4) additional distortions are added to the labels with (8.27) and thereby the shift is artificially increased in the labels. The purpose of this additional shift is to investigate the robustness of the methods, assuming that with a stronger shift, the methods become more sensitive in the weight calculation, which might lead to a higher error rate. The results confirm this expectation, but also show that it is reasonable to assume that DITL provides a better robustness to stronger shifts than other methods.

8.6 Summary

This chapter suggested two new approaches for tackling the problem of inductive transfer learning. The first one DITL, a supervised method, is motivated by a reweighted and unbiased prediction function of the S data. The second method uses an approximation of the Kullback-Leibler divergence to measure the difference in the distributions of the S

[6] Flight dataset available at http://stat-computing.org/dataexpo/2009/

Table 8.3: Results for the mean square error on the real world datasets. Since these datasets exhibit real dataset shifts the advantage of applying weighted S data becomes obvious. Best results are marked as bold text.

Number of P data	# P 20	# P 30	# P 50	# P 70
Earthquake $\nu = 0$, error on $\lvert P_{eval}\rvert = 1000$ and $\lvert S\rvert = 841$				
KRR (on P)	0.0138 / 1.00	0.0106 / 1.00	0.0076 / 1.00	**0.0064 / 1.00**
KRR (on S)	1.13	1.15	1.20	1.23
KRR (on $S \cup P$)	1.12	1.07	1.04	1.04
FS-KRR	1.13	1.14	1.19	1.24
KMM	0.91	0.95	0.98	1.02
ATL	1.04	1.09	1.13	1.16
TL-boosting	0.64	0.88	0.96	**0.99**
KLIEP	0.97	0.99	1.00	1.01
KLITL	0.60	0.83	0.95	1.02
DITL	**0.51**	**0.78**	**0.93**	**1.00**

Number of P data	# P 50	# P 200	# P 400	# P 800
Flight Data $\nu = 0$, error on $\lvert P_{eval}\rvert = 1000$ and $\lvert S\rvert = 2000$				
KRR (on P)	898.01 / 1.00	611.39 / 1.00	265.97 / 1.00	**211.12 / 1.00**
KRR (on S)	0.96	1.02	1.36	1.41
KRR (on $S \cup P$)	0.95	0.99	1.23	1.36
FS-KRR	1.01	1.04	1.35	1.42
KMM	0.88	0.92	0.99	1.01
ATL	0.92	0.99	1.10	1.14
TL-boosting	0.53	0.78	0.89	1.01
KLIEP	0.92	0.96	0.97	1.02
KLITL	0.55	0.76	0.90	**1.00**
DITL	**0.51**	**0.71**	**0.86**	**0.99**

Number of P data	# P 50	# P 100	# P 200	# P 400
Wireless $\nu = 0$, error on $\lvert P_{eval}\rvert = 1000$ and $\lvert S\rvert = 2000$				
KRR (on P)	256.83 / 1.00	230.74 / 1.00	197.23 / 1.00	153.21 / 1.00
KRR (on S)	1.02	0.98	1.10	1.13
KRR (on $S \cup P$)	0.96	1.00	1.12	1.15
FS-KRR	0.99	1.01	1.05	1.10
KMM	0.91	0.95	0.97	1.03
ATL	0.93	0.97	1.02	1.08
TL-boosting	0.74	0.82	0.93	0.99
KLIEP	0.95	0.97	0.99	1.01
KLITL	0.71	**0.78**	0.89	0.98
DITL	**0.69**	**0.79**	**0.87**	**0.96**

Table 8.4: Results for the mean square error on the artifically augmented real world datasets. The robustness of the methods become apparent when the shift is artificially intensified (i.e. (8.27) with $\nu > 0$). Best results are marked as bold text.

Number of P data	# P 20	# P 30	# P 50	# P 70
Earthquake $\nu = 1/3$ (with additional label shift), error on $\|P_{eval}\| = 1000$ and $\|S\| = 841$				
KRR (on P)	0.0261 / 1.00	0.0189 / 1.00	0.0160 / 1.00	**0.0141 / 1.00**
KRR (on S)	1.90	1.98	1.73	1.82
KRR (on $S \cup P$)	1.40	1.35	1.32	1.28
FS-KRR	1.30	1.24	1.19	1.16
KMM	1.08	1.10	1.06	1.03
ATL	1.21	1.18	1.14	1.09
TL-boosting	0.83	0.89	0.95	**1.00**
KLIEP	1.30	1.32	1.38	1.41
KLITL	0.81	0.90	0.96	**1.00**
DITL	**0.60**	**0.83**	**0.92**	**1.00**

Number of P data	# P 50	# P 200	# P 400	# P 800
Flight Data $\nu = 1$ (with additional label shift), error on $\|P_{eval}\| = 1000$ and $\|S\| = 2000$				
KRR (on P)	1432.12 / 1.00	1151.32 / 1.00	813.21 / 1.00	**350.94 / 1.00**
KRR (on S)	1.45	1.51	1.39	1.47
KRR (on $S \cup P$)	1.25	1.19	1.12	1.08
FS-KRR	1.11	1.09	1.06	1.02
KMM	1.01	1.03	0.99	**1.00**
ATL	1.12	1.08	1.05	**1.00**
TL-boosting	0.79	0.88	0.92	**0.99**
KLIEP	1.21	1.24	1.30	1.38
KLITL	0.79	0.89	0.94	1.01
DITL	**0.59**	**0.82**	**0.88**	**0.99**

Number of P data	# P 50	# P 100	# P 200	# P 400
Wireless $\nu = 1$ (with additional label shift), error on $\|P_{eval}\| = 1000$ and $\|S\| = 2000$				
KRR (on P)	431.23 / 1.00	398.19 / 1.00	354.21 / 1.00	**299.85 / 1.00**
KRR (on S)	1.78	1.65	1.77	1.59
KRR (on $S \cup P$)	1.17	1.13	1.10	1.07
FS-KRR	1.20	1.14	1.09	1.04
KMM	1.10	1.05	1.07	1.02
ATL	1.18	1.12	1.10	1.05
TL-boosting	0.86	0.90	0.97	1.01
KLIEP	1.34	1.38	1.40	1.45
KLITL	0.84	0.88	0.94	**0.99**
DITL	**0.74**	**0.83**	**0.92**	**1.00**

and P data. The results indicate that both methods are suitable to account for dataset shifts while the supervised method performs better.

Chapter 9

Conclusions and Outlook

Dataset shifts in machine learning describe problems that - depending on the structure of the data - can be very hard to solve. The term is very general such that it is required to consider special classes of dataset shifts. Since the standard model in the machine learning setting is described by the expression $p(x, y)$ the different types of dataset shifts can be categorized into covariate shift (4.4.1), prior probability shift (4.4.2) and source component shift (4.4.5). Each shift can become infeasible to compensate in situations where either the sample size is too small or the difference inbetween (i.e. the actual shift itself) is too large or even both cases. Therefore, if the transformation of the data generating process is unknown and cannot be stated as an accurate informed mathematical model it becomes necessary to apply statistical methods that require at least some existing connections between the shifted datasets.

Instance based approaches provide an intuitive way of facing such situations. Further they are analytically well justified since they aim to approximate the exact transformation function which is the actual Radon-Nikodym derivative. The approximation itself is then infered based on the sampled datasets available. In this theses a linear combination of gauss kernels (6.1) is applied since such an approach provides a flexible yet simple model and which has been proven to be reliable in serveral other works [57, 71, 89, 91] previously.

The thesis introduced several new algorithms for compensating the covariate shift and source component shift. The methods for compensating a covariate shift are based on other existing work but extend them in a new way by applying the Fourier series approximation. The advantage of the approximation is that the estimation of the weight function becomes more robust in terms of the number of data available. The Fourier series is truncated which implies the neglection of the higher frequencies that mainly only capture the noise that is contained within the data. Therefore, the Fourier series methods put the emphasize on lower frequencies that capture the main structure of the shift. Additionally, in order to make the Fourier series applicable to higher dimensional problems the Hyperbolic cross is applied.

The second type of dataset shift that is considered is the source component shift. Two new methods have been stated. A direct approach (DITL) (8.3.2) that is derived from a weighted kernel ridge regression (6.6.1) and the KL-ITL (8.3.3) method. DITL provides a frameworks that assesses the quality of fit by applying cross validation. Therefore, DITL is a very robust method which is shown in the experimental section 8.5. The second algorithm KLITL is based on the idea of KLIEP and can be considered being an unsupervised approach. The advantages of both methods are that they are very easily applicable by simultaneously providing good results.

All algorithms have been benchmarked on diverse well known datasets and put in comparison with other algorithms that have been developed for the same problem set (8.5, 7.4).

However, still, a lot of unanswered questions remain. It could be interesting to apply the algorithms, discussed in this thesis, to other divergence measures. Or, the question about the kind of weight function approximation that should be used. The currently applied approximation (6.1) is simple and very general. For instance, the approach for compensating covariate shift is not limited to the current choice of a linear combinations of (Gaussian) kernels for the weight function. An interesting possibility would be the use of a sparse grid-based approach [33, 68], 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. Therefore, this approach is an interesting alternative to the application of the Hyperbolic cross. Further, in situations in which a little bit more is known about the transforming process one could consider this knowledge in form of an informed model for the weight approximation which, therefore, would assign more reliable weights. Another question is the robustness of the methods. One could ask: How much similarity in between the data is required in order to apply any of the methods currently available? This implies the next important question: What criteria could indicate when to use shift compensating techniques for which datasets?

Finally, from a wider and more abstract perspective, the results presented in this thesis might also be interesting for other research approaches in the machine learning area. For instance, a high level question is: why do humans learn quicker with less examples than traditional algorithms [99]? One hypothesis is that they have access to an intelligent (human) teacher which has previous experience and knowledge about a particular problem. While traditionally only data x and outcome y is given for the inference a teacher might provide additional information. For that reason, special teacher student interactions can lead to a quicker learning process. One way to model these interactions is by considering knowledge transfer techniques that are similar to those presented in the previous chapters. Therefore, another interesting topic would be the combination of existing teacher student models with these new approaches.

Bibliography

[1] Robert Alexander Adams. *Sobolev spaces*. Pure and applied mathematics. Academic Press, New York, 1978.

[2] Arvind Agarwal and Hal Daumé, III. A geometric view of conjugate priors. *Mach. Learn.*, 81(1):99–113, October 2010.

[3] Samir Al-Stouhi and Chandan K. Reddy. Adaptive boosting for transfer learning using dynamic updates. In *ECML/PKDD*, volume 6911 of *Lecture Notes in Computer Science*, pages 60–75. Springer, 2011.

[4] S. M. Ali and S. D. Silvey. A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society. Series B (Methodological)*, 28(1):131–142, 1966.

[5] 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.

[6] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, December 2008.

[7] N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68, 1950.

[8] Smith Aronszajn, Nachman. Theory of bessel potentials. I. *Annales de l'institut Fourier*, 11:385–475, 1961.

[9] K. Atkinson and W. Han. *Theoretical Numerical Analysis: A Functional Analysis Framework*. Springer New York, 2001.

[10] K. I. Babenko. Approximation by trigonometric polynomials in a certain class of periodic functions of several variables. *Soviet Mathematics Doklady*, 1:672–675, 1960.

[11] Francis Bach, Rodolphe Jenatton, Julien Mairal, and Guillaume Obozinski. Convex optimization with sparsity-inducing norms.

[12] Arindam Banerjee, Srujana Merugu, Inderjit S. Dhillon, and Joydeep Ghosh. Clustering with bregman divergences. *Journal of Machine Learning Research*, 6:1705–1749, December 2005.

[13] Christopher M. Bishop. *Pattern Recognition and Machine Learning.* Springer-Verlag New York, 2006.

[14] John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman. Learning bounds for domain adaptation. In *Advances in Neural Information Processing Systems*, volume 20, 2007.

[15] L. M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7:200–217, 1967.

[16] P. L. Butzer and K Scherer. *Approximationprozesse und Interpolationsmethoden.* Bibliographisches Institut Mannheim, 1968.

[17] Bin Cao, Sinno Jialin Pan, Yu Zhang, Dit-Yan Yeung, and Qiang Yang. Adaptive transfer learning. In *AAAI*. AAAI Press, 2010.

[18] Gavin C. Cawley, Nicola L. C. Talbot, and Olivier Chapelle. Estimating predictive variances with kernel ridge regression. In *Machine Learning Challenges, Evaluating Predictive Uncertainty, Visual Object Classification and Recognizing Textual Entailment, First PASCAL Machine Learning Challenges Workshop, MLCW 2005, Southampton, UK, April 11-13, 2005, Revised Selected Papers*, pages 56–77, 2005.

[19] D.C. Champeney. *A Handbook of Fourier Theorems.* Cambridge University Press, 1989. ISBN 9780521366885.

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

[21] Ciprian Chelba and Alex Acero. Adaptation of maximum entropy capitalizer: Little data can help a lot. *Computer Speech & Language*, 20(4):382–399, 2006.

[22] N. N. Chentsov. *Statistical decision rules and optimal inference / N.N. Cencov ; [translated from the Russian by the Israel Program for Scientific Translations ; translation edited by Lev J. Leifman].* American Mathematical Society Providence, R.I, 1982.

[23] Andrzej Cichocki, Anh Huy Phan, and Rafal Zdunek. *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation.* Wiley, Chichester, 2009.

[24] Michael Collins, Robert E. Schapire, and Yoram Singer. Logistic Regression, Adaboost and Bregman Distances. In *Machine Learning*, pages 158–169, 2000.

[25] I. Csiszár. Eine informationstheoretische Ungleichung und ihre Anwendung auf den Beweis der Ergodizität von Markoffschen Ketten. *Publications of the Mathematical Institute of Hungarian Academy of Sciences*, 8:95–108, 1963.

[26] Wenyuan Dai, Qiang Yang, Gui rong Xue, and Yong Yu. Boosting for transfer learning. In *International Conference on Machine Learning (ICML)*, pages 193–200, 2007.

[27] Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu. Self-taught clustering. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 200–207, 2008.

[28] Ingrid Daubechies. *Ten Lectures on Wavelets.* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1992.

[29] H. Daumé III. Frustratingly easy domain adaptation. In *ACL*, pages 256–263, 2007.

[30] George S. Fishman. *Monte Carlo : concepts, algorithms, and applications.* Springer series in operations research. Springer, New York, Berlin, 1996.

[31] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of online learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139, 1997.

[32] F.G. Friedlander and M.S. Joshi. *Introduction to the Theory of Distributions.* Cambridge University Press, 1998.

[33] 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.

[34] Jochen Garcke and Thomas Vanck. *Importance Weighted Inductive Transfer Learning for Regression*, pages 466–481. 2014.

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

[36] Zoubin Ghahramani and Matthew J. Beal. Variational inference for bayesian mixtures of factor analysers. In *In Advances in Neural Information Processing Systems 12*, pages 449–455. MIT Press, 2000.

[37] Tom Goldstein and Stanley Osher. The split bregman method for l1-regularized problems. *SIAM J. Img. Sci.*, 2(2):323–343, April 2009.

[38] M. Griebel and S. Knapek. Optimized approximation spaces for operator equations. Technical Report 568, SFB 256, Univ. Bonn, 1999.

[39] Quanquan Gu, Zhenhui Li, and Jiawei Han. Learning a kernel for multi-task clustering. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA, August 7-11, 2011*, 2011.

[40] Shaobo Han, Xuejun Liao, and Lawrence Carin. Cross-domain multitask learning with latent probit models. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*, 2012.

[41] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. 2001.

[42] E. Hellinger. Neue begründung der theorie quadratischer formen von unendlichvielen veränderlichen. *Journal für die reine und angewandte Mathematik*, 136:210–271, 1909.

[43] J.M. Hilbe. *Logistic Regression Models*. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis, 2009.

[44] Jean Honorio and Dimitris Samaras. Multi-task learning of gaussian graphical models. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, pages 447–454, 2010.

[45] Jiayuan Huang, Alexander J. Smola, Arthur Gretton, Karsten M. Borgwardt, and Bernhard Schölkopf. Correcting sample selection bias by unlabeled data. In *NIPS*, pages 601–608, 2006.

[46] O. Kallenberg. *Foundations of Modern Probability*. Applied probability. Springer, 2002.

[47] Takafumi Kanamori, Shohei Hido, and Masashi Sugiyama. A least-squares approach to direct importance estimation. *Journal of Machine Learning Research (JMLR)*, 10:1391–1445, 2009.

[48] Andreas Kirsch. *An Introduction to the Mathematical Theory of Inverse Problems.* Springer-Verlag New York, Inc., New York, NY, USA, 1996.

[49] 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`.

[50] S. Knapek. *Approximation und Kompression mit Tensorprodukt-Multiskalenräumen.* Dissertation, Universität Bonn, April 2000.

[51] Shu Kong and Donghui Wang. Transfer heterogeneous unlabeled data for unsupervised clustering. In *ICPR*, pages 1193–1196. IEEE, 2012.

[52] S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[53] Ilja Kuzborskij and Francesco Orabona. Stability and hypothesis transfer learning. In *International Conference on Machine Learning*, June 2013.

[54] Ilja Kuzborskij, Francesco Orabona, and Barbara Caputo. From n to n+1: Multiclass transfer incremental learning. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, June 2013.

[55] B.B. Mandelbrot and R.L. Hudson. *Misbehavior of Markets.* Basic Books, 2004.

[56] J. Mercer. Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society, London*, 209:415–446, 1909.

[57] 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.

[58] Tetsuzo Morimoto. Markov processes and the H-Theorem. *Journal of the Physical Society of Japan*, 18(3):328–331, March 1963.

[59] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain Generalization via Invariant Feature Representation. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, pages 10–18. 2013.

[60] F. Natterer. Regularisierung schlecht gestellter Probleme durch Projektionsverfahren. *Numer. Math.*, 28:329–341, 1977.

[61] Andrew Y. Ng and Michael I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in Neural Information Processing Systems 14*, pages 841–848. MIT Press, 2002.

[62] Otton Nikodym. Sur une généralisation des intégrales de m. j. radon. *Fundamenta Mathematicae*, 15:131–179, 1930.

[63] Peter Oswald. *On discrete norm estimates related to multilevel preconditioners in the finite element method.* Proc. Internat. Conf. on the Theory of Functions. 1992.

[64] Sinno Jialin Pan and Qiang Yang. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, October 2010.

[65] Sinno Jialin Pan, James T. Kwok, and Qiang Yang. Transfer learning via dimensionality reduction. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2*, AAAI'08, pages 677–682, 2008.

[66] David Pardoe and Peter Stone. Boosting for regression transfer. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, pages 863–870, 2010.

[67] Alexandre Passos, Piyush Rai, Jacques Wainer, and Hal Daumé III. Flexible modeling of latent task structures in multitask learning. In *ICML*, 2012.

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

[69] B. Priestley. *Non-linear and Non-stationary Time Series Analysis.* Probability and mathematical statistics. Academic Press, 1988.

[70] Novi Quadrianto, James Petterson, and Alex J. Smola. Distribution matching for transduction. In *Advances in Neural Information Processing Systems 22*, pages 1500–1508, 2009.

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

[72] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, pages 257–286, 1989.

[73] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. 2007.

[74] Piyush Rai and Hal Daumé III. Infinite predictor subspace models for multitask learning. In *AISTATS*, pages 613–620, 2010.

[75] Rajat Raina, Andrew Y. Ng, and Daphne Koller. Constructing informative priors using transfer learning. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 713–720, New York, NY, USA, 2006. ACM.

[76] Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y. Ng. Self-taught learning: Transfer learning from unlabeled data. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 759–766, 2007.

[77] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning).* The MIT Press, 2005.

[78] D.G. Rees. *Essential Statistics, Fourth Edition.* Chapman & Hall texts in statistical science series. Taylor & Francis, 2000.

[79] M.D. Reid and R.C. Williamson. Composite binary losses. *Journal of Machine Learning Research*, 11, September 2010.

[80] F. Riesz. Sur une espèce de géométrie analytique des systèmes de fonctions sommables. 144:1409–1411, 1907.

[81] L.C.G. Rogers and D. Williams. *Diffusions, Markov Processes, and Martingales: Volume 1, Foundations.* Cambridge Mathematical Library. Cambridge University Press, 2000.

[82] Reuven Y. Rubinstein. *Simulation and the Monte Carlo Method.* John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1981.

[83] Ulrich Rückert and Stefan Kramer. Kernel-based inductive transfer. In *ECML/PKDD 2008*, pages 220–233. Springer, 2008.

[84] Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press, Cambridge, MA, USA, 2001.

[85] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.*, 10(5):1299–1319, July 1998.

[86] B.W. Silverman. *Density Estimation for Statistics and Data Analysis.* Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis, 1986.

[87] S. Smolyak. Quadrature and interpolation formulas for tensor products of certain classes of functions. *Soviet Mathematics, Doklady*, 4:240–243, 1963.

[88] 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.

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

[90] 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.

[91] Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul Von Bünau, and Motoaki Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *NIPS*, 2008.

[92] Ben Tan, Erheng Zhong, Evan Wei, and Xiang Qiang Yang. Multi-transfer: Transfer learning with multiple views and multiple sources. In *SDM*, 2013.

[93] Ben Taskar, Simon Lacoste-Julien, and Michael I. Jordan. Structured prediction, dual extragradient and bregman projections. *Journal of Machine Learning Research*, 7:1627–1653, 2006.

[94] V Temlyakov. Approximation of periodic functions, 1993.

[95] A.N. Tikhonov. On the stability of inverse problems. *Dokl. Acad. Nauk USSR*, 39:195–198, 1943.

[96] Tatiana Tommasi, Francesco Orabona, and Barbara Caputo. Safety in numbers: Learning categories from few examples with multi model knowledge transfer. In *CVPR*, pages 3081–3088. IEEE, 2010.

[97] Hans Triebel. *Interpolation theory, function spaces, differential operators.* North-Holland mathematical library. North-Holland Pub. Co., 1978.

[98] 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.

[99] Vladimir Vapnik and Rauf Izmailov. Learning using privileged information: Similarity control and knowledge transfer. *Journal of Machine Learning Research*, 16: 2023–2049, 2015.

[100] Vladimir Vapnik, Igor Assis Braga, and Rauf Izmailov. A constructive setting for the problem of density ratio estimation. page 434–442, 2014.

[101] A. Wächter and L. T. Biegler. On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106:25–57, 2006.

[102] Grace Wahba. *Spline models for observational data*, volume 59 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), 1990.

[103] E. J. Wegman. Reproducing kernel hilbert spaces. 8:81–84, 1982.

[104] Lei Xu and Michael I. Jordan. On convergence properties of the em algorithm for gaussian mixtures. *Neural Computation*, 8:129–151, 1995.

[105] Ya Xue, Xuejun Liao, Lawrence Carin, and Balaji Krishnapuram. Multi-task learning for classification with dirichlet process priors. *Journal of Machine Learning Research*, 8:2007, 2007.

[106] J. Yang and Y. Zhang. Alternating direction algorithms for l1-problems in compressive sensing. *SIAM Journal on Scientific Computing*, 33:250–278, 2011.

[107] Jiyan Yang, Vikas Sindhwani, Haim Avron, and Michael W. Mahoney. Quasi-monte carlo feature maps for shift-invariant kernels. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 485–493, 2014.

[108] Pei Yang and Wei Gao. Multi-view discriminant transfer learning. In *IJCAI*. IJCAI/AAAI, 2013.

[109] Qiang Yang, Sinno Jialin Pan, and Vincent Wenchen Zheng. Estimating location using wi-fi. *IEEE Intelligent Systems*, 23(1):8–13, 2008. ISSN 1541-1672.

[110] Dan Zhang, Jingrui He, Yan Liu, Luo Si, and Richard D. Lawrence. Multi-view transfer learning with a large margin approach. In *KDD*, pages 1208–1216. ACM, 2011.

[111] Yu Zhang and Dit-Yan Yeung. Semi-supervised multi-task regression. In *ECML/PKDD 2009*, pages 617–631, 2009.

[112] Yu Zhang and Dit-Yan Yeung. Multi-task learning in heterogeneous feature spaces. In *AAAI*, 2011.

[113] Yu Zhang, Dit-Yan Yeung, and Qian Xu. Probabilistic multi-task feature selection. In *NIPS*, 2010.