Active Learning by Discrepancy Minimization

A Comparison of Active Learning Methods Motivated by Generalization Bounds

T. J. Viering

Technische Universiteit Delft

TU Delft Delft University of Technology

Challenge the future

Active Learning by Discrepancy Minimization

A Comparison of Active Learning Methods Motivated by Generalization Bounds

by

T. J. Viering

in partial fulfillment of the requirements for the degree of

Master of Science
in Computer Science.

at the Delft University of Technology,
to be defended publicly on Wednesday August 24, 2016 at 13:00.

Student number:	4333055
Master programme:	Media and Knowledge Engineering (MKE)
Specialization:	Pattern Recognition
Faculty:	Electrical Engineering, Mathematics and Computer Science
Supervisors:	Prof. dr. M. Loog
	Dr. J. Krijthe

Thesis committee:

Chair	Prof. dr. ir. M.J.T. Reinders,	PRB, EEMCS, TU Delft
University Supervisor	Prof. dr. M. Loog,	PRB, EEMCS, TU Delft
University Co-Supervisor	Dr. J. Krijthe,	ME, LUMC, PRB, EEMCS, TU Delft
Committee Member	Prof. dr. E. Eisemann	CGV, EEMCS, TU Delft

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft Delft University of Technology

Preface

During my bachelor in physics, I discovered my true passion when I did my minor in computer science: artificial intelligence. That is why I decided to pursue a masters degree in computer science at the Technical University of Delft. My favorite courses were pattern recognition and machine learning. It fascinated me that these subjects are very theoretical, yet these concepts are also applicable in practice.

I decided to graduate in the research group Pattern Recognition and Bioinformatics at the faculty for Electrical Engineering, Mathematics and Computer Science. Initially I struggled in choosing a topic for my master thesis since my interests are so broad. First I conducted interesting research under the supervision of Hamdi Dibeklioglu. I'm grateful for his support. However, I realized I wanted a more fundamental research project. So I made a new start with prof. dr. Marco Loog and dr. Jesse Krijthe in 'Active Learning'.

The main motivation behind my thesis was to find out more about theoretically well-founded active learning methods. We found that a lot of active learning methods in literature were justified by intuition or 'hand waving' arguments. For many of these methods it was unclear in which cases these methods work and when they don't: their assumptions were vague or hard to check empirically. With active learning it is in practice costly or impossible to detect if a method works or doesn't for the specific problem at hand. Therefore we looked into 'conservative' methods based on generalization bounds, since these methods account for worst case scenarios they are likely to work even in these worst case scenarios. Furthermore it is straightforward to verify whether or not the assumptions of these generalization bounds hold. This thesis is the result of this investigation.

First and foremost I would like to thank Marco Loog and Jesse Krijthe for being great supervisors. I always looked forward to our weekly meetings. We've had many interesting discussions on machine learning, maths, and the research community in general. Only near the end of my research project were we able to limit our meeting times to an hour. Thank you for the many mathematical insights, critical questions, support and your ability to keep me focused. Last not and least, thank you for all your advice on scientific writing and thank you for taking the time to proofread my work.

To my committee members Marcel Reinders and Elmar Eisemann, I am grateful for taking the time to read my thesis and attending my thesis presentation. Finally, I would like to thank my parents, friends and girlfriend for all their support.

T. J. Viering
Delft, August 2016

Contents

1

Introduction

Recently we have seen many successful applications of machine learning. For example face recognition, music recognition, object detection, and landmark detection are possible nowadays with high accuracy. In these fields it is relatively easy to collect large labeled datasets, since it takes little time to annotate faces, objects or landmarks in photos. There are many large music databases that contain labeled music samples. If a lot of labeled data is available pattern recognition systems typically perform better, explaining their success in these examples.

However, in some fields most collected data is unlabeled. This makes it difficult to train accurate pattern recognition models for these applications. In fields such as speech recognition for rare languages, computational biology, bio-informatics, drug discovery, video classification, text categorization, relevance feedback, recommendation systems or medical diagnosis it can be expensive, difficult or time consuming to annotate data [1]. In some of these settings labels can only be provided by experts. For example if one wants to train a pattern recognition system to automatically suggest medical diagnoses based on patient records or other medical data, much labeled data is necessary to train this system. In this case the label is the diagnosis of a doctor. In speech recognition for rare languages, trained linguists need to annotate speech samples which is a very time consuming task. For computational biology, bio-informatics or drug discovery, biological or chemical samples need to be evaluated to obtain labels which could require precious laboratory time and materials. For recommendation systems, such as Netflix movie recommendations, user feedback is costly to obtain, since users do not like to rate many movies. For relevance feedback users need to indicate if a query returns relevant results, which is also a time consuming and difficult task. For video classification and text categorization it is also costly to obtain labels, since possibly lengthy videos need to be watched or lengthy texts needs to be studied to determine if the video or text addresses a certain topic. Each of these examples illustrates it may be costly or time consuming in practice to obtain large labeled datasets, and therefore for these settings it is often hard or unpractical to use pattern recognition systems.

However, pattern recognition systems for these fields are highly desirable and relevant for society and industry. Imagine a world where risk assessment of a medical condition could be performed with the same ease and accuracy as a smartphone can recognize faces. Such developments could be revolutionary for the health care industry. Furthermore, these pattern recognition models in other fields can be used to automate costly, difficult or repetitive tasks to reduce the workload of people.

Active learning aims to reduce labeling costs. Imagine we have trained a pattern recognition model on a small labeled dataset. To improve the model we may be able to obtain more labeled data. Generally, unlabeled samples are selected randomly from a large unlabeled dataset for labeling. Most predictive models even require that labeled data is selected randomly. In the active learning setting, instead of selecting data randomly, an algorithm proposes which objects should be labeled. In the words of B. Settles: “The key idea behind active learning is that a machine learning

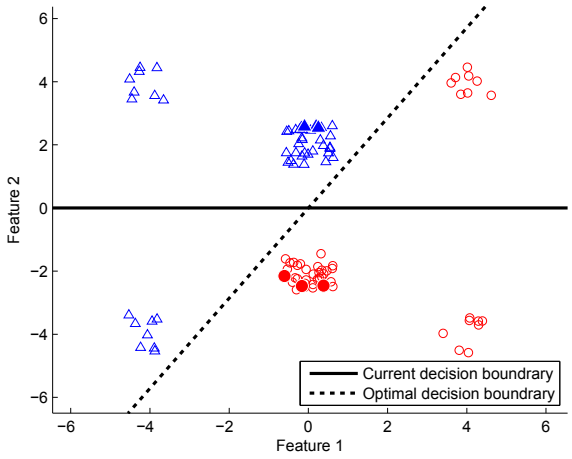


Figure 1.1: This figure illustrates when uncertainty sampling can query many irrelevant examples. The solid icons are labeled examples, the outlined icons indicate unlabeled data. The positive and negative class are indicated by blue triangles and red circles. Uncertainty sampling will select data near the current decision boundary for labeling. Because of this the boundary will change little: the boundary will remain approximately horizontal. Because of this strategy, it will take uncertainty sampling a long time before it discovers the remaining clusters. Observe that uncertainty sampling constructs a very biased sample. In this work we look at active learning methods that try to approximate the complete data distribution as fast as possible with labeled data. In this example these methods will discover all clusters directly.

algorithm can perform better with less [labeled] training [data] if it is allowed to choose the data from which it learns" [1]. Therefore, active learning could make it practical and economical to train accurate pattern recognition systems in fields where labeled data is costly to acquire.

We examine sequential pool based active learning [1] for binary classification. In this setting we are given a pool of unlabeled data, and the active learner can select objects one by one from this pool for labeling. After labeling, the object is added to the training set and the model is retrained. Our goal is to optimize the performance of the model with as little queries as possible.

We explore the use of generalization bound minimization for active learning. Generalization bounds give an upper bound on the generalization error of the model: the error of the model on previously unseen data. We study the performance of four active learning methods that minimize generalization bounds.

The structure of this introduction from here on is the following. First we motivate generalization bound minimization for active learning and we discuss the context in which we study these methods. Then we give the main motivation for this work: two novel active learning algorithms based on the minimization of a notion of discrepancy between empirical distributions. Finally we pose several research questions to gain more insights in these active learning methods and list our contributions.

We now motivate the usage of generalization bound minimization for active learning. Some methods in active learning focus too much on 'exploitation'. 'Exploitation' in the context of active learning means selecting objects near to the decision boundary for labeling[1]. Intuitively, this makes sense: the model is uncertain of the label for objects close to the decision boundary, thus these objects are informative for refining the decision boundary. However, in some cases this sampling method known as uncertainty sampling can can focus too much on querying irrelevant samples [1]. See Figure 1.1 for an example where this occurs. Because of this behavior uncertainty sampling may perform worse than random sampling. Note that this would be disastrous in a real world active learning setting, since this means the small and costly labeling budget has been wasted on irrelevant samples.

One of the reasons 'exploitative' strategies can perform worse than random sampling is that these methods can induce a large sampling bias [2]. By actively selecting data we do not sample from the original distribution but we obtain a biased sample. Because of this, the model obtained

may not generalize well to the original distribution. In Figure 1.1 this is also illustrated: in this example uncertainty sampling only samples from the two clusters in the center, and thus we obtain a biased sample. To correct for sample bias data is typically reweighted [3, 4]. This reweighting can be guided by minimizing a generalization bound.

In active learning most well known generalization bounds do not hold, because the training sample is not independent and identically distributed (i.i.d) because the data is selected by an active learner. A generalization bound originating from sample bias correction from [4] is applicable to active learning because this bound does not assume i.i.d. data. Instead of reweighting our data as in sample bias correction, we let the generalization bound guide the active learning process: we query the object that minimizes the bound. We introduce an active learning algorithm that minimizes this generalization bound and compare it to other methods that minimize generalization bounds. Before we discuss these other generalization bounds we first discuss the advantages of using bounds for active learning and the setting in more detail.

The generalization bounds that we study are non-probabilistic, meaning they always hold if their assumptions are satisfied. These bounds even hold in a worst-case scenario. By minimizing these bounds, we are assured we will not be too 'exploitative' or greedy like uncertainty sampling. Furthermore, by explicitly accounting for worst-case scenarios we will likely perform better in such scenarios. This is important since in active learning we could be in such a worst-case scenario without knowing it — this is impossible to detect without additional labeled data. Another advantage of minimizing a generalization bound is that the connection with the generalization performance is explicit. For other active learning methods this connection is not always clear. Furthermore, the bounds we study have assumptions which can be empirically verified.

All bounds we study do not require information of the labels, thus the obtained active learning strategies are independent of the observed labels. We limit our study to these so called non-adaptive active learners.

These label independent strategies have multiple advantages and one disadvantage. Because these methods are label independent, we can generate a list of all samples that should be labeled in advance. This speeds up the annotation process. The annotator in this setting does not have to wait for the active learner to suggest which object should be labeled next. Another advantage is that these active learners cannot be misled by incorrect labels, and are thus unaffected by label noise. However, active learners that *do* use label information may perform better because they have more information: this is the most important disadvantage of these methods.

We illustrate why these label independent strategies work for active learning. We explain the trends observed in the learning curves of these methods and characterize when these methods can improve upon random sampling.

The bounds we study give guarantees on the surrogate loss. This is the loss that the learning algorithm minimizes. In classification we are usually interested in the performance in terms of zero-one or misclassification loss. The zero-one loss measures whether or not the predicted class was correct. Learning models minimize the surrogate loss since minimizing the zero-one loss induces an NP-complete optimization problem [5]. The relation between the surrogate loss and the zero-one loss is not straightforward, except for the fact that the surrogate loss always upper bounds the zero-one loss. This upper bound may be very loose or the losses may be related in unexpected ways.

Unlike other work [6–10], we measure the performance of active learning methods in terms of the surrogate loss to exclude effects of the interaction between the zero-one loss and surrogate loss. Because of this our comparison of the bounds for active learning is more straightforward: we only see the effects caused by the bounds themselves. Also, when we chose our model, we have given a preference for a specific surrogate loss function. Therefore it makes sense to study performance in terms of this surrogate loss. For example when choosing the logistic regression model, the surrogate loss quantifies how accurate the posterior probabilities are, and thus if the logistic regression model is chosen it is sensible to measure performance in terms of the log loss.

The main quantities in the generalization bounds, the quantities that are minimized for active learning, only capture all behavior in the realizable setting. The realizable setting is an artificial scenario where the observed labels are generated by a model of our chosen model class (in this case

there is no 'model mismatch'). The realizable setting eliminates effects that arise due to model mismatch, and therefore we study this artificial setting to gain more insights in the behavior of these generalization bounds when they are used for active learning. Such an in depth comparison in the realizable setting was not considered in other works that minimize generalization bounds for active learning [6–10].

In this work we use the kernel ridge regression model which uses the squared loss as surrogate loss. It has been shown that kernel ridge regression can achieve state-of-the-art results in classification [5] and is computationally efficient for large datasets [11], and thus is a relevant model for modern machine learning research. Furthermore, some of the bounds that we study are applicable to any loss function. In future work these bounds may be applied to construct active learning methods for other models that use other losses, such as logistic regression or SVM's. All of our results are straightforwardly applicable to regression since we use the squared loss.

Now we have discussed the setting in detail, we will introduce the active learners and their generalization bounds that we will study and pose our research questions.

The first active learner we introduce minimizes the discrepancy. The discrepancy is a measure which was introduced by [12] for sample bias correction. The discrepancy quantifies the difference between empirical distributions. The discrepancy measure is related to another measure called the maximum mean discrepancy (MMD) which is also used for sample bias correction[3] like the discrepancy. Active learning based on minimization of the MMD has already been proposed [6, 9]. In [9] this strategy was motivated by the minimization of a generalization bound. We show their bound may not hold in the active learning setting and we give a modified version of their bound that does hold.

Recently it has been shown that sample bias correction based on discrepancy minimization performs better or as good as sample bias correction based on minimization of the MMD [4]. In [4] it is argued that the discrepancy performs better because its generalization bound takes the loss function and the hypothesis set of the model into account while the MMD does not. The hypothesis set in this context means the kernel used by the model. This is the main motivations of this work: do these results from sample bias correction generalize to the active learning scenario? In other words: can active learning based on the discrepancy improve upon the state-of-the-art MMD active learner? This is our main research question.

We give an in depth theoretical analysis comparing the bounds of the MMD and the discrepancy. We show that the MMD can be adapted to take the hypothesis set and loss into account like the discrepancy. We investigate whether this is useful for active learning. The remaining difference then between the MMD and the discrepancy is that the discrepancy takes the loss function of the learning algorithm explicitly into account while the MMD does not. Our second novel theoretical result is that the discrepancy generalization bound is always tighter than the MMD generalization bound under the same assumptions.

Our theoretical analysis thus suggests the discrepancy indeed is a better bound. Since the discrepancy bound is tighter it can estimate the generalization error more accurately. Does this then guarantee better performance in active learning? We investigate this question, and we also investigate if tighter generalization bounds translate to improved active learning performance in general. To this end we also introduce a new quantity which we call the nuclear discrepancy. The corresponding generalization bound of the nuclear discrepancy is looser, but we argue it might perform better in active learning using a probabilistic analysis. More generally, we investigate what features determine the success of generalization bounds for active learning.

Finally, we compare the performance of the MMD and our introduced discrepancy and nuclear discrepancy active learners to the active learning method based on Transductive Experimental Design (TED) [8]. This is a state-of-the-art label independent active learner that was shown to minimize a generalization bound [7]. The TED bound is derived especially for the ridge regression model, and thus takes loss and hypothesis set into account like the discrepancy. This bound is however derived in a completely different way than the MMD and (nuclear) discrepancy bounds: it depends on the analytical solution of the ridge regression model. The MMD and discrepancy bound can be applied to any loss function while TED cannot. Therefore it is interesting to investigate if the bound of TED is tighter and how the TED active learning strategy compares with the MMD

and (nuclear) discrepancy active learning strategy. This can tell us whether model dependent bounds can be more favorable for active learning in general.

Below we summarize our contributions:

- We give an in depth theoretical analysis of the discrepancy, MMD and TED generalization bounds. Furthermore, we derive a novel generalization bound for TED which always holds and which does not depend on any probabilistic assumptions.
- We derive a novel generalization bound for the MMD that is applicable to the active learning setting. We also derive a novel MMD generalization bound that always holds.
- We show how the MMD bound can be adapted to take the hypothesis set and loss into account like the discrepancy generalization bound.
- We show that this adapted MMD bound is comparable to the discrepancy bound, and that the discrepancy bound is always tighter under the same assumptions.
- We define a new quantity called the nuclear discrepancy and give a novel generalization bound in terms of this quantity.
- We introduce two novel active learning methods: the discrepancy active learner and the nuclear discrepancy active learner.
- We give a comparison of the active learning methods in terms of the surrogate loss in the realizable setting. Such a comparison was not considered in other works. This comparison can give us more insight in the behavior of the generalization bounds when they are used for active learning because we eliminate effects that do not originate from the bounds themselves. This way we can find out what properties of these generalization bounds are important for good active learning performance.

Besides this, we answer the following research questions:

Q1. Why and in what case can non-adaptive active learning methods improve upon random sampling?

Q2. Is it beneficial to take the hypothesis set and the loss into account for active learning based on the MMD?

Q3. Will our introduced discrepancy active learning strategy improve upon the MMD active learning strategy as suggested by our theoretical analysis? (main question)

Q4. How does the model dependent TED active learner and its generalization bound compare with the discrepancy and MMD active learners and bounds?

Q5. Is the tightness of a generalization bound related to performance in active learning?

1.1. Outline

First we describe the definitions, notation and setting in Section 1.2. Afterward, we discuss related work that uses the Maximum Mean Discrepancy (MMD) quantity, the discrepancy quantity, and the Transductive Experimental Design (TED) objective in Section 1.3. In particular we discuss the state-of-the-art MMD and TED active learners. We also briefly discuss other methods that minimize generalization bounds for active learning.

We discuss the MMD quantity, the discrepancy quantity, the TED objective, and the nuclear discrepancy quantity and their generalization bounds in Chapter 2 in depth. In Section 2.1 we describe the MMD quantity and derive a novel generalization bound in terms of the MMD that always holds in the active learning scenario. At the end of this section we present the most important result: we show how to adapt the MMD to take into account the hypothesis set and the loss of the learning algorithm. In Section 2.2 we describe the discrepancy quantity. We show

the similarities between the MMD and discrepancy, and present the generalization bound for the discrepancy which is directly comparable to the MMD bound. The most important result of this section is the comparison of these bounds: we show that in the realizable setting the discrepancy always results in a tighter bound. TED is described in detail in Section 2.3. We discuss the TED objective, its interpretation and generalization bound. We give an in depth comparison of the TED bound and the discrepancy and MMD bound. We give a detailed description and motivation of the novel nuclear discrepancy quantity in Section 2.4. We also give its novel generalization bound in this section. At the end of the chapter, in Section 2.5, we give a summary of the bounds and discuss how they are used to construct active learning algorithms.

We describe the experimental setup in Chapter 3. In Chapter 4 we perform extensive experiments comparing the active learning strategies. Using these experiments we answer the research questions posed in the introduction. The most important parts of this chapter are sections 4.3, 4.4 and 4.5. In Section 4.3 we compare the active learning methods on real world and artificial data in the realizable setting. In Section 4.4 we study the differences between our proposed discrepancy active learner and the existing MMD active learner. We show the shortcomings of the discrepancy active learner, which motivates the introduction of the nuclear discrepancy active learner. In Section 4.5 we compare the nuclear discrepancy active learner to the other active learners.

In Chapter 5 we revisit the research questions and answer them. We reflect on some choices we have made in the experiments and discuss their influence on the results. Afterward we discuss how the active learners discussed in this work could be improved. We also relate our results to the setting of sample bias correction and discuss the implications of our findings for other fields. Besides this we place our results in a broader context in the active learning research field and discuss what barriers remain for practical applications of these active learning techniques.

The main conclusions of this work are given in Chapter 6.

In the appendices we give several derivations that were too long for the main text, additional results, and additional details so all experiments are reproducible. The appendices also contain the proofs of all generalization bounds. The most important proof is given in Appendix C, where we show that the discrepancy bound is always tighter than the MMD generalization bound in the realizable case. In this appendix we also describe how the MMD can be adapted to take the loss and hypothesis set of the learning algorithm into account.

1.2. Definitions, Notation and Setting

Let $\hat{Q}$ be the labeled sample that is constructed by the active learner, and let $\hat{P}$ be the complete dataset we have access to. Thus $\hat{P}$ includes labeled and unlabeled data, and thus $\hat{Q} \in \hat{P}$. Both are samples from an unknown distribution P over the input space $\mathcal{X} = \mathbb{R}^d$. We assume $\hat{P}$ is an identically distributed and independent (i.i.d.) sample from P, but $\hat{Q}$ may be a non-i.i.d. sample since these labeled objects may have been chosen by the active learner. This work focuses on binary classification. We consider two settings: the realizable case, where the labels are generated by a model of our model class (an artificial setting), and the agnostic case where this may not be the case (this is a real world setting). For the agnostic setting we use the original binary labels of the dataset and thus the output space is $\mathcal{Y} = \{+1, -1\}$. In the realizable case the output space $\mathcal{Y}$ is a subset of $\mathbb{R}$ since the model outputs are real numbers.

The active learner has access to both sets $\hat{P}$ and $\hat{Q}$. In this work we limit ourselves to non-adaptive active learners, meaning that the active learners do not use label information from the labeled set $\hat{Q}$. We focus on sequential active learning: the active learner can select one sample for labeling in each iteration of the active learning process. The sample will be labeled and added to the training set $\hat{Q}$. Afterward, the model will be retrained on $\hat{Q}$ and is evaluated in terms of the squared loss on an unseen test set. The goal is to minimize the mean squared error on the unseen test set with as little queries as possible. This setting is summarized in Figure 1.2.

Batch selection methods might work better in practice than sequential selection: in such a setting multiple objects are selected for labeling at once by the active learner. But these selection strategies often select samples to minimize some (convex) relaxation of the original objective of the active learner. In case we compare batch methods, it might be the case that one active learner

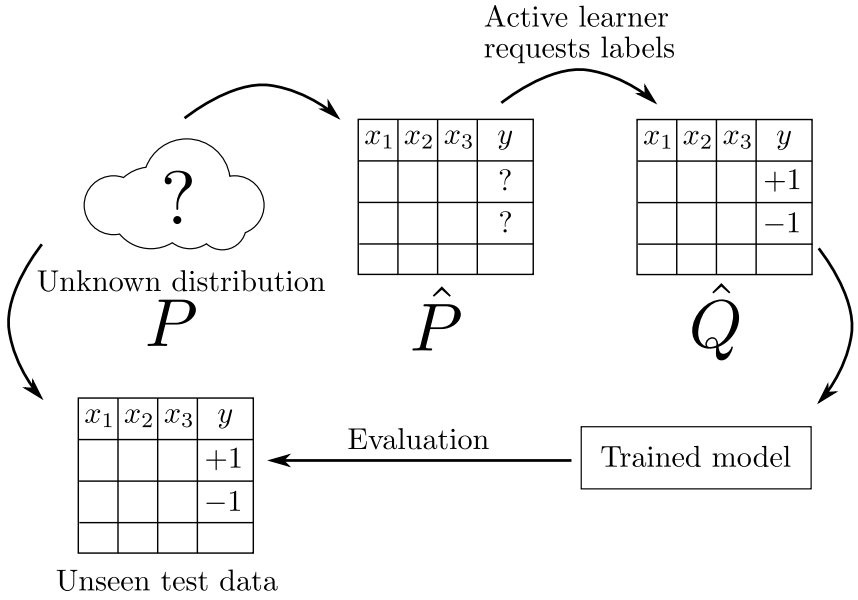


Figure 1.2: Schematic overview of datasets and active learning setting. The dataset $\hat{P}$ is the complete dataset the active learner has access to which is unlabeled. The dataset $\hat{Q}$ is the labeled dataset constructed by the active learner.

outperforms the other, because its optimization method is better. We are more interested in the underlying objectives instead of comparing optimization procedures of batch active learners. In fact, in appendix I we show that the state-of-the-art batch active learner of the MMD of [6] performs worse than the sequential MMD active learner in terms of the objective that is minimized: the MMD. Therefore we only study sequential active learners in this work.

We choose to evaluate the model in terms of the mean squared error, since this is the surrogate loss of our ridge regression model. We do not use the zero-one loss as is usual in classification, since the generalization bounds we study only give direct guarantees on the surrogate loss on the set $\hat{P}$. The surrogate loss upper bounds the zero-one loss but further the relation between this bound is unclear: it may be very loose or related to each other in unexpected ways. We choose to measure performance in the surrogate loss so we eliminate these effects that do not originate from the generalization bounds we study. Furthermore, since we have chosen the ridge regression model, it is also obvious to compare performance in terms of this loss, since this is the loss the model aims to minimize. By choosing this model we have given the preference for this surrogate loss to approximate the zero-one loss function.

We use the convention that all vectors are column vectors. x_i is the feature vector of object i. $X_{\hat{P}}$ and $X_{\hat{Q}}$ are the $n_{\hat{P}}$ by d and $n_{\hat{Q}}$ by d data matrices of the sets $\hat{P}$ and $\hat{Q}$. We assume there exists a deterministic labeling function $f : \mathcal{X} \to \mathcal{Y}$ that determines the labels to derive the generalization bounds. This assumption might seem restrictive since it does not allow label noise. This assumption is however only explicitly violated in case a dataset contains objects with the same feature vector x with different labels[1]. We use y_i or $f(x_i)$ to indicate the label of object x_i.

We choose the kernel ridge regression model without intercept term and we do not use class priors. Since we work with kernel ridge regression we need to introduce some notation concerning kernels. This work only considers positive definite symmetric (PSD) kernels K. For these kernels a reproducing kernel Hilbert space (RKHS) exists and we indicate the RKHS by $\mathcal{H}$. We require PSD kernels since we require the RKHS of K to derive the generalization bounds. Figure 1.3 on page 9 illustrates the used notation concerning kernels. Each object x in the input space $\mathcal{X}$ can be mapped to the RKHS $\mathcal{H}$ using the function ψ. $K(x, x')$ is the kernel function between object

[1]Furthermore, we believe all generalization bounds can easily be adapted to a probabilistic setting.

x and x', this kernel function can be computed as an inner product in the RKHS of the kernel:

$$K(x, x') = \langle \psi(x), \psi(x') \rangle_K$$

where $\langle ., . \rangle_K$ denotes the inner product in the RKHS $\mathcal{H}$ of K. The norm of a vector h in the RKHS is written as $||h||_K$ and is given by $||h||_K = \sqrt{\langle h, h \rangle}_K$. $K(x, x')$ can typically be computed only using x and x' as well. In this work we use the linear kernel, the squared kernel and the Gaussian kernel. For the linear kernel we compute $K(x, x')$ using:

$$K(x, x') = x^T x'$$

Note that for the linear kernel the RKHS $\mathcal{H}$ is equal to the input space $\mathcal{X}$. For the squared kernel we have that:

$$K(x, x') = (x^T x')^2$$

For the Gaussian kernel $K(x, x')$ is defined as:

$$K(x, x') = \exp\left(-\frac{||x' - x||_2^2}{2\sigma^2}\right)$$

The parameter σ is a parameter the kernel. It determines the smoothness of the functions $h \in \mathcal{H}$, where $\mathcal{H}$ is the RKHS of the Gaussian kernel. If σ is larger, h evaluated over the input space $\mathcal{X}$ will be smoother. The RKHS $\mathcal{H}$ of the Gaussian kernel can be shown to be infinite dimensional. Furthermore the Gaussian kernel is a normalized kernel: each vector mapped from $\mathcal{X}$ to the RKHS $\mathcal{H}$ of the kernel has norm one: $||\psi(x)||_K = 1$ for all $x \in \mathcal{X}$ (see [13, p. 96]).

Kernel ridge regression will give us a model $h \in \mathcal{H}$. This hypothesis or model is linear in $\mathcal{H}$, the RKHS of K, but due to the mapping ψ can be nonlinear in the input space $\mathcal{X}$. In case we work in a linear kernel, we will write the hypothesis h as w, and the prediction of the model h on x, $h(x)$, is in this case given by $w^T x$. In case we work in a different kernel K, the prediction $h(x)$ can be computed in the RKHS by $h(x) = \langle h, \psi(x) \rangle_K$.

We write $||.||_1$, $||.||_2$ and $||.||_\infty$ to indicate the l_1 norm, the l_2 norm and infinity norm of finite dimensional vectors, respectively. For matrix norms we use the same notation: we use $||.||_p$ to indicate the operator norms for matrices, which are given by:

$$||A||_p = \max_{||x||_p = 1} ||Ax||_p$$

In particular, the $p = 1$ operator matrix norm of a n by m matrix A is given by:

$$||A||_1 = \max_{1 \leq j \leq n} \sum_{i=1}^{m} |a_{ij}|$$

For real symmetric matrices B the $p = 2$ operator norm is given by:

$$||B||_2 = \max_i |\lambda_i|$$

Where λ_i are the eigenvalues of the matrix B. This norm is also known as the spectral norm. For real symmetric matrices B the Frobenius matrix norm is given by:

$$||B||_F = \sqrt{\sum_i \lambda_i^2}$$

Finally, the nuclear norm (also known as trace norm) of a real symmetric matrix B is given by:

$$||B||_* = \sum_i |\lambda_i|$$

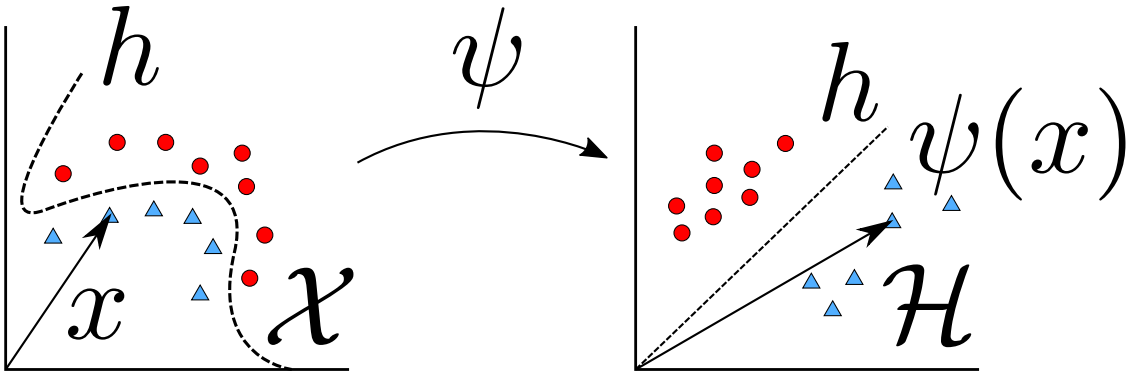


Figure 1.3: Illustration of the mapping a PSD kernel induces. Left the input space is shown, here a hypothesis h (shown using dashed lines) can be nonlinear. In $\mathcal{H}$, the RKHS of the kernel K, the hypothesis becomes a linear function: thus h can be described by a vector in $\mathcal{H}$. ψ is the mapping from the input space $\mathcal{X}$ to the RKHS $\mathcal{H}$.

We denote a kernel matrix by $K_{\hat{Q}\hat{P}}$ where each entry in the matrix is given by evaluating the kernel function between the corresponding samples of $\hat{Q}$ and $\hat{P}$, and is of size $n_{\hat{Q}}$ by $n_{\hat{P}}$. For the computations of the MMD we need a second kernel which is indicated with K' and we indicate its RKHS by $\mathcal{H}'$.

Kernel ridge regression minimizes the following objective for an hypothesis $h \in \mathcal{H}$ when trained on the sample $\hat{Q}$:

$$F_{(\hat{Q},f)} = \frac{1}{n_{\hat{Q}}} \sum_{i=1}^{n_{\hat{Q}}} (h(x_i) - y_i)^2 + \lambda ||h||_K^2 = L_{\hat{Q}}(h, y) + \lambda ||h||_K^2 \tag{1.1}$$

So kernel ridge regression minimizes $L_{\hat{Q}}(h, y)$ — the average squared loss evaluated on the set $\hat{Q}$ between the hypothesis h and the labels y. L with a subscript will always indicate the average loss on the set denoted by the subscript. We may also use other arguments in L to indicate the losses between different labeling functions or hypotheses. The notation $L(h(x), f(x)) = (h(x) - f(x))^2$ will be used to indicate the squared loss on a particular example x. In Equation 1.1 the parameter $\lambda > 0$ is a regularization parameter which controls the complexity of the obtained model. This second term is added to avoid overfitting of the model. Furthermore, if the Gaussian kernel is used, the parameter σ controls the complexity of the obtained solution as well. Since if σ is larger the resulting function $h(x)$ is smoother, and thus the resulting model is less complex. See Appendix H to see how to compute h that minimizes the objective of Equation 1.1.

Observe that if we have trained our ridge regression model on the set $\hat{Q}$ and we want to compute the model outputs on the set $\hat{P}$, this can be done by computing:

$$h_{\hat{P}} = \bar{H} f_{\hat{Q}}$$

Where we write $f_{\hat{Q}}$ as the vector of labels of the set $\hat{Q}$, and $h_{\hat{P}}$ as the vector of outputs of the model h on the set $\hat{P}$. The matrix $\bar{H}$ is the hat matrix[2] of the ridge regression model. For the definition of $\bar{H}$ see Appendix H.

We choose a subset of $\mathcal{H}$ as our hypothesis set: $H = \{h \in \mathcal{H} : ||h||_K \leq \Lambda = \frac{1}{\sqrt{\lambda}}\}$. The hypothesis set $\mathcal{H}$ is the set of all functions that may be outputted by the training procedure of the model. In appendix F it is shown[3] that this does not constrain ridge regression in choosing any hypothesis from $\mathcal{H}$, thus we do not need to introduce any additional constraints when minimizing

[2]Note that we define the hat matrix differently than is standard in most statistics texts. Usually the hat matrix places the 'hat' on the set that the model is trained on, however we are always interested in the predictions on the set $\hat{P}$ and therefore we redefined the hat matrix as above.

[3]In this appendix we also show that the proposed way of bounding the hypothesis space by [4] does not result in an informative bound on Λ.

Equation 1.1. This explicit determination of the hypothesis set H is necessary to compute the generalization bounds in Chapter 2. Furthermore, we often explicitly need a constant C that verifies $L(h(x), f(x)) \leq C$ for all $h \in H$ and all $x \in \mathcal{X}$ in the generalization bounds that we study. In Appendix F a value of C that verifies this condition is given as well.

1.3. Related Work

Several other works have proposed to minimize error bounds for pool based active learning. First, we discuss related work which use the Maximum Mean Discrepancy (MMD) measure, the discrepancy measure and the Transductive Experimental Design (TED) objective. In particular we discuss the works that introduce the state-of-the-art MMD and TED active learners. Afterward we briefly discuss three other directions of work that also minimize generalization bounds for active learning, that are either incomparable to our work or for which we think the theoretical justification is not entirely correct.

Originally the MMD was proposed for a statistical test to determine if two empirical samples originate from the same distribution [14]. The MMD has also been used in transfer learning, domain adaptation and sample bias correction to make training and test distributions more similar by sample reweighting [3]. Active learning by minimization of the maximum mean discrepancy (MMD) was first proposed by [6].

In [9] a generalization bound is given that motivates to minimize the MMD measure in active learning. We now give several arguments why this bound does not hold in the active learning setting. The bound assumes the labeled samples $\hat{Q}$ come from a distribution Q. However, we believe that strictly speaking the distribution Q may not exist in the active learning scenario. If we use a probabilistic active learner we could define a probability distribution Q according to which to sample, however this distribution will change every iteration. So in general we cannot speak of a single distribution Q from which all empirical samples $\hat{Q}$ are obtained. For deterministic active learners it is even more difficult to see how the distribution Q can exist.

Furthermore, [9] proposes to minimize the MMD between the distributions Q and P, and use empirical samples from $\hat{Q}$ and $\hat{P}$ to estimate the MMD between the probability distributions of Q and P. The estimators for the MMD between probability distributions require i.i.d. samples[4] from Q and P. Since the samples selected by the active learner depend on previously selected samples, this i.i.d. assumption is violated. The bound derived in [9] is therefore not appropriate for active learning. In Section 2.1 we prove a modified version of their bound for the MMD that does hold for non-i.i.d. samples and is thus applicable to active learning. Furthermore our bound is guaranteed to hold for deterministic labeling functions.

The works of [6] and [9] use batch active learning. In Appendix I we show that the batch active learner of [6] performs worse than a simple sequential MMD active learner in terms of the MMD objective, and therefore we only consider the sequential MMD active learner. The active learner of [9] is not label independent and therefore we do not include it in our comparisons. Different from this work, these works do not measure the performance in terms of the surrogate loss and also use a different model. Therefore our results are not directly comparable with their results.

In [12] the discrepancy measure is introduced in the context of domain adaptation. In [4] it is demonstrated empirically that the discrepancy improves upon MMD in sample bias correction in terms of the mean squared error on some datasets, and on other datasets it matches the performance of the MMD. In this work the discrepancy is also generalized to arbitrary kernels. In [15] more bounds are introduced in terms of the discrepancy and the generalized discrepancy measure is introduced for domain adaptation.

Transductive Experimental Design (TED) was introduced in [8] as an extension of methods of experimental design. In [16] a batch active learner was proposed to minimize the TED objective, however the proposed batch algorithm is only applicable to linear kernels. Because we mainly work with the Gaussian kernel we do not include this batch active learner in our comparison.

In [7] active learning and semi-supervised learning are combined for a model similar to the ridge regression model. They perform active learning by minimizing a generalization bound. This

[4] In the appendix of [9] they refer to Theorem 7 of [14] which assumes i.i.d. samples, see appendix A.2 of [14].

generalization bound results in the same objective as TED when the semi-supervised part of the model is ignored. Similar to the MMD and discrepancy, the generalization bound upper bounds the surrogate loss of the trained model. We discuss their generalization bound and extend it to a non-probabilistic setting. Our TED bound is guaranteed to always hold for deterministic labeling functions.

Now we mention several other works for active learning using generalization bounds. We think the following methods are not entirely theoretically justified, or these methods are not comparable to our work.

A generalization bound based on the Transductive Rademacher complexity is minimized in [17] to perform active learning. However, we believe the bound is used erroneously in this work and we will now briefly argue why. In [17] the function class is chosen after observing the set $\hat{Q}$ and $\hat{P}$, while the bound only holds for function classes that are chosen after observing the full sample. This means no knowledge of $\hat{P}$ and $\hat{Q}$ is allowed, only the set $\hat{P}+\hat{Q}$ may be known for choosing the function class. See [18, Theorem 2] for details. In [17] the function class is chosen to depend on the set $\hat{Q}$, and therefore we believe the bound will not hold. We believe the Transductive Rademacher complexity bound is not applicable to the active learning scenario. Their active learner however does perform well, and one may wonder if it is possible to adapt this generalization bound to hold in the active learning scenario.

Importance weighting error bounds from domain adaptation might seem attractive for active learning. However, importance weighting is usually based on the ratio between the probability densities of P and Q, while in active learning the probability density function Q does not strictly exist as argued before. Furthermore we studied the importance weighting generalization bound of [19] but found that the bounds given here do not apply to active learning since these bounds require i.i.d. samples. Importance weighting for active learning has been proposed in [20], however this strategy requires the use of the observed labels to determine the next query and this is therefore not comparable to our work.

In [10] a generalization bound is derived for logistic regression and this bound is minimized for active learning. However this generalization bound is not label independent and uses a different model, and therefore is also incomparable with this work.

2

Theory

In this chapter we review the MMD measure, the discrepancy measure, the TED objective and the nuclear discrepancy measure. We explain what these quantities mean, how they are computed and we discuss their generalization bounds. We give in depth comparisons of the bounds and measures. We first discuss the MMD measure and and follow up with the discrepancy, since the discrepancy is similar to the MMD and can be considered a more refined version of the MMD quantity. We discuss TED afterward so we can compare it in depth with the MMD and discrepancy. We note some advantages of the TED bound, and this leads us to introduce a novel quantity called the nuclear discrepancy. At the end of the chapter we summarize the measures, their bounds and their assumptions, and explain how we use these bounds to construct active learning algorithms.

2.1. Maximum Mean Discrepancy (MMD)

In this section we discuss in detail what the MMD measure is and how it is computed. The work of [9] uses the MMD measure for active learning but motivates this with a generalization bound that may not hold in the active learning scenario as we have argued in Section 1.3. We give a novel generalization bound in terms of the MMD that explicitly holds for the active learning scenario using a similar technique as [15]. We describe in detail the qualitative features of the bound. To compute the MMD a kernel K' has to be chosen. We discuss how the kernel is chosen in other work. In the last subsection we give the most important theoretical result of this section: we derive how the kernel of the MMD can be chosen depending on the kernel that is used by the learning algorithm in case of the squared loss. Using this analysis we obtain a bound similar to the discrepancy that takes the hypothesis set and the loss into account.

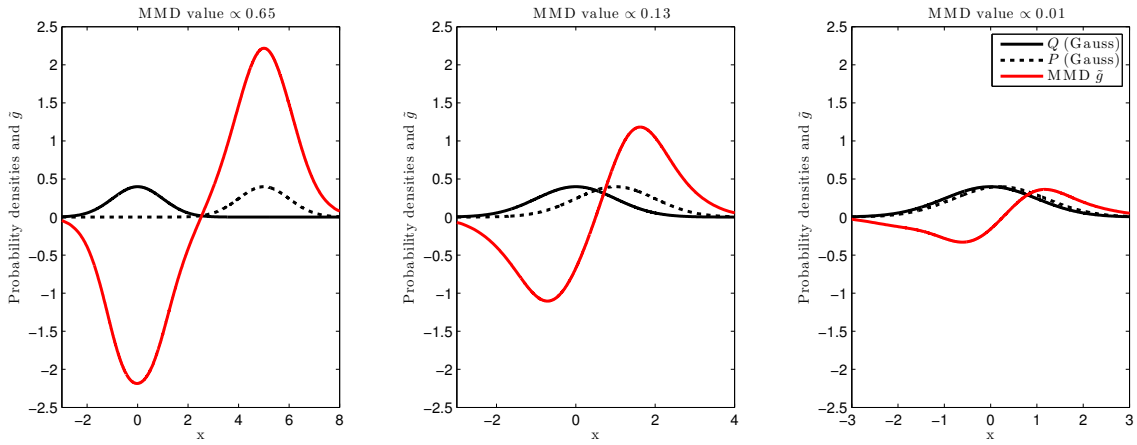
2.1.1. MMD Measure

In this subsection we discuss what the MMD measure means and how it is defined. We also look at multiple examples to get a better understanding of the MMD measure.

We define the MMD as:

$$\mathrm{MMD}(\hat{P}, \hat{Q}) = \max_{\tilde{g} \in H'} \left(\frac{1}{n_{\hat{P}}} \sum_{x \in \hat{P}} \tilde{g}(x) - \frac{1}{n_{\hat{Q}}} \sum_{x \in \hat{Q}} \tilde{g}(x) \right) \tag{2.1}$$

Here $\tilde{g}$ is the worst-case function from a set of functions H'. The set H' is usually chosen as a subset of functions in the RKHS $\mathcal{H}'$ of a universal kernel K' such as the Gaussian kernel. Similar to the hypothesis set we choose the set H' as $H' = \{h \in \mathcal{H} : ||h||_{K'} \leq \Lambda'\}$. In other works the constant Λ' is fixed as 1, we consider the more general case where $\Lambda' \neq 1$. This is useful for our comparison between the discrepancy and the MMD. Our definition of the MMD only differs by a constant from the usual definition of the MMD. In particular this choice has no effect on the active learning strategy, see Section 2.5.



(a) (b) (c)

Figure 2.1: This figure illustrates the function $\tilde{g}$ maximizing Equation 2.1. This is the function where the means of $\tilde{g}$ on $\hat{P}$ and $\hat{Q}$ are the most different. The Gaussian distributions remain the same except for the distance between their means, which is decreased from a) to c). We see that if the (empirical) distributions are more similar, the mean of $\tilde{g}$ will be more similar on both empirical samples $\hat{P}$ and $\hat{Q}$ due to the smoothness constraint of the Gaussian kernel, and thus the MMD will be smaller.

Our estimator of the MMD coincides with the 'biased MMD estimator' from [14]. This estimator is biased, since it has a bias when estimating the MMD between probability distributions. For our purpose this estimator is not biased since it estimates the exact empirical quantity that is required in our generalization bound that estimates the performance on empirical samples (which we introduce in the next subsection).

We will now illustrate the behavior of the MMD quantity using three examples shown in Figure 2.1. In this figure we see two probability distributions P and Q. In all examples P and Q are Gaussian distributions, the only difference between the three examples is the distance between the means of the Gaussian distributions. We sample an equal amount of samples from P and Q to obtain the empirical datasets $\hat{P}$ and $\hat{Q}$. We compute the MMD between these two empirical sets: we however don't show the empirical sets, but instead display the probability distributions since this is more convenient to display. The relative MMD quantities are shown in the title of the figures. The function $\tilde{g}$ maximizing Equation 2.1 is shown in red, appendix A.2 explains how to compute the function $\tilde{g}$. The Gaussian kernel was used to compute the MMD with $\sigma = 0.5$ which imposes certain smoothness constraints on the function $\tilde{g}$.

We observe the following about the function $\tilde{g}$ in general in Figure 2.1. $\tilde{g}$ is as positive as possible if $P(x) > Q(x)$, and $\tilde{g}$ is as negative as possible if $P(x) < Q(x)$. This is so that the empirical means of $\tilde{g}$ differ as much as possible on the set $\hat{P}$ and $\hat{Q}$ to maximize Equation 2.1. The amount of positivity and negativity $\tilde{g}$ can attain depends on the distributions $\hat{P}$ and $\hat{Q}$ and the smoothness conditions on $\tilde{g}$.

In Figure 2.1a we observe that the function $\tilde{g}$ can become more positive on $\hat{P}$ and more negative on $\hat{Q}$ since the distributions overlap little when compared to Figure 2.1c. In Figure 2.1c we observe that the function $\tilde{g}$ can hardly become positive on $\hat{P}$ and hardly negative on $\hat{Q}$. Because the distributions overlap more $\tilde{g}$ only has a small window to become positive when $P(x) > Q(x)$. The same argument holds for $\tilde{g}$ becoming negative. Figure 2.1b illustrates the intermediate scenario where $\tilde{g}$ can become somewhat positive and negative.

Now we relate the function $\tilde{g}$ to the MMD quantity. We observed that if the empirical distributions $\hat{P}$ and $\hat{Q}$ are more similar, the empirical mean of $\tilde{g}$ on $\hat{P}$ and $\hat{Q}$ will be more similar, because $\tilde{g}$ cannot attain large absolute values. Because of this nececarily the MMD quantity defined by Equation 2.1 becomes smaller, as can also be observed in the title of the Figure 2.1c. If the empirical distributions $\hat{P}$ and $\hat{Q}$ differ more, the difference between the means of $\tilde{g}$ under both distributions can be larger and the MMD will become larger as seen in the title of Figure 2.1a. This

example thus illustrates that the MMD measures the similarity between empirical distributions.

The MMD can also be interpreted in a different manner. It can be shown that the MMD is equal to the distance between the means of $\hat{P}$ and $\hat{Q}$ in the RKHS of K':

$$\text{MMD}(\hat{P}, \hat{Q}) = \Lambda' ||\mu_{\hat{P}} - \mu_{\hat{Q}}||_{K'}$$

See appendix A.1 for the derivation of this equation. In this appendix it is also shown how to compute the MMD using empirical quantities.

2.1.2. MMD Generalization Bound

In the previous subsection we have seen that the MMD quantity measures the difference between the empirical distributions $\hat{Q}$ and $\hat{P}$. In this subsection we discuss the generalization bound of the MMD.

We use a similar technique to the one employed in [15] to derive a novel generalization bound for the MMD that applies to the active learning scenario where the samples $\hat{Q}$ may not be independently and identically distributed (i.i.d.). The derivation of the generalization bound is given in appendix A.1. We give the bound in the theorem below:

Theorem 1 (MMD Generalization bound) *Given any hypothesis $h \in H$ and any deterministic labeling function $f(x)$ and given that the loss function $g(x) = L(h(x), f(x)) \in H'$, where $H' = \{h \in \mathcal{H} : ||h||_{K'} \leq \Lambda'\}$, we have that:*

$$|L_{\hat{P}}(h, f) - L_{\hat{Q}}(h, f)| \leq MMD(\hat{P}, \hat{Q})$$

This bound indicates that the average squared loss of any h on $\hat{P}$ and $\hat{Q}$ is more similar if the MMD is small between the empirical distributions $\hat{P}$ and $\hat{Q}$. Why would we want this? We give two interpretations that explain why this is desirable. Note that in this context we train on $\hat{Q}$, the (small) labeled sample in active learning, and we evaluate on $\hat{P}$, the complete empirical data distribution on which we want to perform well in terms of the squared loss.

Note that in the active learning setting it is useful that this bound holds for any $h \in H$, since after selecting a new example we do not know which model h we will obtain since before selecting this example we do not yet know its label.

The bound above upper bounds the following quantity for all $h \in H$:

$$|L_{\hat{P}}(h, f) - L_{\hat{Q}}(h, f)| \tag{2.2}$$

Why do we want this quantity to be small for all $h \in H$? Ideally, we want that training on the dataset $\hat{Q}$ and training on the dataset $\hat{P}$ results in the same model, since training on $\hat{P}$ is the best we can do in active learning since this would correspond to labeling all samples. When the loss for all $h \in H$ is the same on the datasets $\hat{P}$ and the $\hat{Q}$, training on $\hat{Q}$ *is* the same as training on $\hat{P}$, in which case the quantity in Equation 2.2 is zero for all $h \in H$. If the quantity in Equation 2.2 is small for all $h \in H$, the difference in the training procedure will be small as well, since the losses on $\hat{Q}$ and $\hat{P}$ will be similar for all h. Thus if the quantity in Equation 2.2 is small we will obtain a similar model if we train on $\hat{Q}$ and if we train on $\hat{P}$. Since Equation 2.2 is upperbounded by the MMD quantity for all $h \in H$, we want the MMD to be small. This argument is also illustrated in Figure 2.2 on page 16.

Now we consider the second interpretation of this bound, to this end we rewrite the bound in Theorem 1 in a more familiar form:

$$L_{\hat{P}}(h, f) \leq L_{\hat{Q}}(h, f) + \text{MMD}(\hat{P}, \hat{Q}) \tag{2.3}$$

Here we directly relate the performance on $\hat{P}$ to the empirical loss on $\hat{Q}$ and the MMD measure. From this point of view, we see that the training procedure minimizes the loss on $\hat{Q}$, and if the MMD is small, this low loss will generalize to $\hat{P}$. If we are more interested in the generalization performance on the distribution P, we can combine Equation 2.3 with a standard generalization

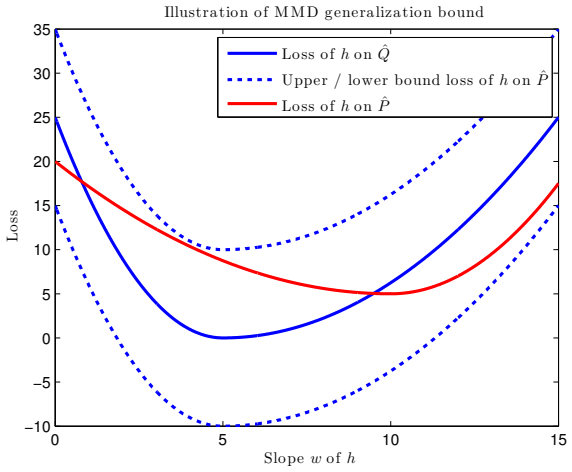


Figure 2.2: This illustrates the MMD generalization bound in one dimension for a linear kernel. We only look at the loss functions — there is no corresponding dataset. Minimization of the loss on $\hat{Q}$ results in the hypothesis where the blue line is minimal, $w = 5$. We actually would want to train on $\hat{P}$ since this corresponds training on the entire dataset. If this were possible we would obtain the hypothesis where the red line is minimal: $w = 10$. We see that if the bound is tighter, the difference between the red line (loss on $\hat{P}$) and the blue line (loss on $\hat{Q}$) becomes smaller. Thus the tighter the bound, the more the minimization of the loss on $\hat{Q}$ will result in a similar hypothesis as the the minimization on the set $\hat{P}$. This is what we what we want: we want to obtain a hypothesis that is as close as possible as the hypothesis that minimizes the loss on $\hat{P}$.

bound to relate the performance on $\hat{P}$ to the performance on P, such as a Rademacher generalization bound [13, Theorem 10.7]. Since the set $\hat{P}$ is an i.i.d. sample from P, and since the set $\hat{P}$ is relatively large, the error on $\hat{P}$ is a good estimate of the generalization error[1]. Therefore, a small error on $\hat{P}$ is desirable, since this would mean we likely obtain a model with good generalization performance.

The main assumption in this bound is that the kernel K' is 'rich' enough so its set H' contains the true loss function $g = L(h(x), f(x))$ for all $h \in H$ and the labeling function f. This is required because in deriving the bound we approximated the true loss function g by a worst-case function $\tilde{g} \in H'$. It is possible to relax this assumption using a similar approach as [15]. The proof is given in Appendix A.3, we give the novel bound below:

Theorem 2 (MMD Generalization bound that always holds) *Define the set H' as: $H' = \{h \in \mathcal{H} : ||h||_{K'} \leq \Lambda'\}$. This theorem applies to* any *loss function L. We assume the labeling function $f(x)$ is deterministic. Then for any hypothesis $h \in H$ the following bound holds:*

$$|L_{\hat{P}}(h, f) - L_{\hat{Q}}(h, f)| \leq MMD(\hat{P}, \hat{Q}) + \eta_{MMD}(\hat{P}, \hat{Q}, f, h)$$

Where $\eta_{MMD}(\hat{P}, \hat{Q}, f, h)$ is given by:

$$\eta_{MMD}(\hat{P}, \hat{Q}, f, h) = \min_{\tilde{g} \in H'} \left(\frac{1}{n_{\hat{P}}} \sum_{x \in \hat{P}} |L(h(x), f(x)) - \tilde{g}(x)| + \frac{1}{n_{\hat{Q}}} \sum_{x \in \hat{Q}} |L(h(x), f(x)) - \tilde{g}(x)| \right)$$

Observe that by relaxing the assumption $g \in H'$ we get an additional term $\eta_{\text{MMD}}(\hat{P}, \hat{Q}, f, h)$. This term measures the approximation error when approximating the true loss function g by a loss function $\tilde{g} \in H'$. Observe that this term cannot be computed in a real world active learning experiment, since it requires us to know all labels of the set $\hat{P}$.

One may wonder why there is an absolute value in the definition of the MMD. The reason is twofold: because of this we obtain a bound on the absolute value of $L_{\hat{P}}(h, f) - L_{\hat{Q}}(h, f)$ which is

[1] Note that this may not be necessarily true since overfitting may occur. This can be accounted for by the complexity term in the Rademacher generalization bound.

useful as illustrated by the analysis accompanying Figure 2.2. Another reason this is useful is because in this case the approximation error $\eta_{\text{MMD}}(\hat{P}, \hat{Q}, f, h)$ becomes straightforward to compute.

Using Theorem 2 we can rewrite the bound in a similar form as Equation 2.3. In this case we obtain the following equation:

$$L_{\hat{P}}(h, f) \leq L_{\hat{Q}}(h, f) + \text{MMD}(\hat{P}, \hat{Q}) + \eta_{\text{MMD}}(\hat{P}, \hat{Q}, f, h)$$

This illustrates that it is beneficial if the approximation error of the MMD is small or zero, since in this case the bound on $\hat{P}$ will be tighter as well.

In case $g \in H'$ the approximation error becomes zero since g does not need to be approximated. We may wonder when this is the case, this is however difficult to determine for the bound in this form. In Section 2.1.4 we return to this question and give a clear example when this approximation error vanishes.

2.1.3. How to Choose the Kernel of the MMD According to Literature

When comparing (empirical) distributions with the MMD measure the Gaussian kernel is used in most works. We briefly discuss how these works determine the bandwidth σ of the Gaussian kernel in this subsection.

In [14] the bandwidth σ is chosen as the median distance between all objects in the dataset. For domain adaptation, sample bias correction and transfer learning it is usual to set σ to the value of σ that is used by the Gaussian kernel used to train the model [3]. When the MMD is used for active learning as in [6] and [9] likely the same σ was used as for the kernel of the MMD and the learning algorithm, this is however not explicitly stated.

2.1.4. Choosing the Kernel of the MMD to Take the Hypothesis Set Into Account

According to [4] it is beneficial to take into account the hypothesis set and the loss when deriving generalization bounds like the MMD bound. This means that the quantities in the bound should depend on the kernel K and the loss used by the learning algorithm. In case of the MMD, the kernel K' of the MMD should depend on the kernel K and the loss of the learning algorithm. In this section we discuss how we can take the hypothesis set and squared loss into account when computing the MMD generalization bound.

Before we derive how to take the loss and kernel of the learning algorithm into account, we discuss why it might be beneficial to do so. Generally, it is known that certain machine learning models perform better than others for certain datasets. It is also generally known that certain kernels work better for some datasets than others. This illustrates why it is beneficial if the generalization bound takes the kernel and the model (or the surrogate loss of the model) into account. This possibly will allow us to distinguish between different models and kernels for the dataset at hand. In any case we know that the performance for different kernels and different models is different, and therefore we may want this to be reflected in the generalization bound. Including this information possibly will make the generalization bound tighter or more relevant for the setting considered.

Now we move on to deriving how we can take the model (or the surrogate loss of the model) and the kernel into account for the MMD. First we assume that we are in the realizable setting, and then we extend our argument to the agnostic setting as well. In the realizable setting f is in the hypothesis set of the learning algorithm: $f \in H$ (where H depends on K). Given this assumption we can derive which kernel K' should be chosen so that the assumption of the MMD bound in Theorem 1 is satisfied: $g \in H'$. Or in other words: given that we are in the realizable case and we use the squared loss, how should we choose K' to assure that the approximation error of the MMD $\eta_{\text{MMD}}(\hat{P}, \hat{Q}, f, h)$ is zero? In appendix C this derivation is given. We find that if we use a kernel K for the learning algorithm, the MMD should be computed with the kernel $K'(x, y) = K(x, y)^2$ in case we use the squared loss. We also find under these assumptions that $\Lambda' = 4\Lambda^2$ should be chosen to guarantee that $g \in H'$.

Of course, one is usually not in the realizable case. However, if one chooses a certain model class, it is logical to assume that this is the model class that best approximates the labeling function f. Since if one believes that a different model class can approximate f better, one should choose this different model class instead. Therefore, if we choose the set H as hypothesis set, and one believes this is the optimal model class for approximating f, then the model class H' should be chosen as above. If one chooses a different model class H' to compute the MMD than the model class prescribed above, then even in the realizable setting the approximation error may not be zero.

In case we use a linear kernel for the learning algorithm, the kernel K' should be chosen as a quadratic kernel. In case we use a Gaussian kernel for our learning algorithm with bandwidth σ, then the kernel K' corresponds to a Gaussian kernel with bandwidth $\sigma' = \frac{\sigma}{\sqrt{2}}$. This indicates that we should not choose the same bandwidth for the learning algorithm and the computation of the MMD, as is often done in other works as mentioned in the previous subsection. Since $\sigma > \frac{\sigma}{\sqrt{2}}$, if the Gaussian kernel with $\sigma' = \sigma$ is used to compute the MMD, we are using a function that is too smooth to approximate the true loss function. Therefore, if one uses $\sigma' > \frac{\sigma}{\sqrt{2}}$, the approximation error $\eta_{\text{MMD}}(\hat{P}, \hat{Q}, f, h)$ likely will not be zero, even in the realizable setting.

This thus prescribes how to choose K' depending on the hypothesis set and loss of the learning algorithm, similar to the discrepancy measure defined in the next section. To indicate the difference between both variants of the MMD, we will use 'MMD HS' to indicate the MMD quantity that takes the hypothesis set and loss into account. The only difference then between MMD HS and the discrepancy is that the discrepancy takes all aspects of the loss function into account, while MMD HS does not, which we will discuss in the next section.

2.2. Discrepancy

In this section we first review the discrepancy measure. We describe how it is computed and we indicate the similarities with the MMD measure. We give an easy to interpret bound of the discrepancy measure. This bound is directly comparable with the MMD bound given in Section 2.1.2. We compare both bounds in the realizable setting where the approximation errors of both bounds are equal to zero — this is the case where the MMD bound takes the hypothesis set and loss into account. In this setting we show that the discrepancy bound is always tighter. This theoretical result is novel and was not shown in other work concerning the discrepancy.

2.2.1. Discrepancy Measure

In this subsection we show how the discrepancy measure is defined, we illustrate what it measures using artificial examples, and we compare it with the MMD measure.

The discrepancy is defined as:

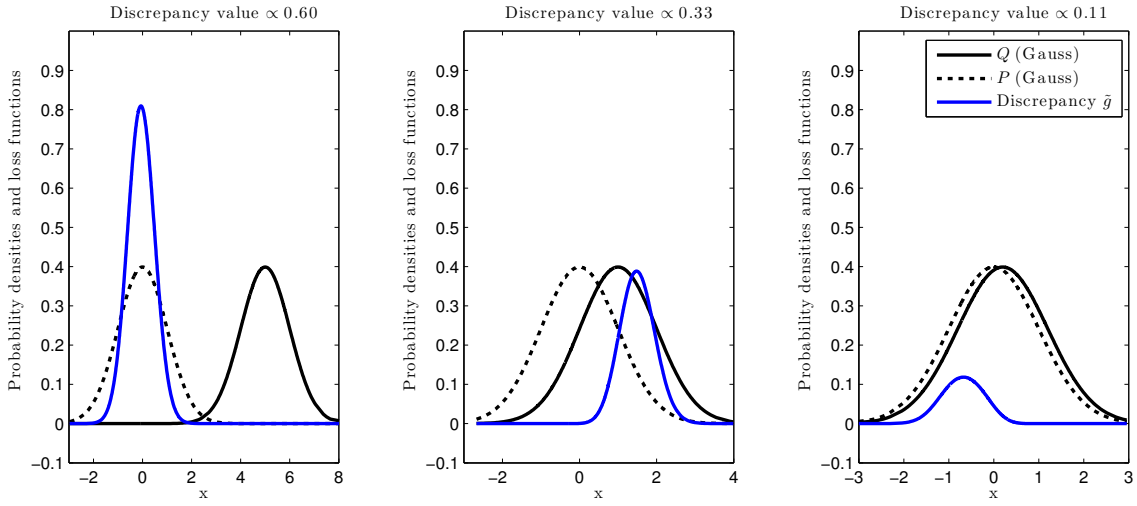
$$\text{disc}(\hat{P}, \hat{Q}) = \max_{h,h' \in H} |L_{\hat{P}}(h', h) - L_{\hat{Q}}(h', h)|$$

Note that in this case $L_{\hat{P}}(h, h')$ and $L_{\hat{Q}}(h, h')$ are defined as the average loss of h with respect to h' over the empirical samples $\hat{P}$ and $\hat{Q}$. The loss is the same loss as used by the learning algorithm, and H indicates the hypothesis set of the learning algorithm. We thus observe that the discrepancy measure explicitly depends on the hypothesis set and loss of the model. Because of this the generalization bound of the discrepancy that depends on the discrepancy quantity also takes the hypothesis set and loss of the model into account. Refer to the discussion in Subsection 2.1.4 for why this may be beneficial.

For the squared loss, we can rewrite the discrepancy as:

$$\text{disc}(\hat{P}, \hat{Q}) = \max_{\tilde{z} \in \mathcal{H}, ||\tilde{z}||_K \leq 2\Lambda} |\frac{1}{n_{\hat{P}}} \sum_{x \in \hat{P}} \tilde{z}(x)^2 - \frac{1}{n_{\hat{Q}}} \sum_{x \in \hat{Q}} \tilde{z}(x)^2| \tag{2.4}$$

See appendix B.1 for the derivation. We can write the discrepancy in the same form as the MMD



(a) (b) (c)

Figure 2.3: This figure illustrates the function $\tilde{g}$ maximizing Equation 2.1. This is the same example as was used in Figure 2.1. We see that if the (empirical) distributions are more similar, the mean of $\tilde{g}$ will be more similar on both empirical samples $\hat{P}$ and $\hat{Q}$ due to the smoothness constraint, and thus the discrepancy will be smaller.

quantity in Equation 2.1. To this end we define the function $\tilde{g}(x) = \tilde{z}(x)^2$. Then we obtain:

$$\text{disc}(\hat{P}, \hat{Q}) = \max_{\tilde{z} \in \mathcal{H}, ||\tilde{z}||_K \leq 2\Lambda} |\frac{1}{n_{\hat{P}}} \sum_{x \in \hat{P}} \tilde{g}(x) - \frac{1}{n_{\hat{Q}}} \sum_{x \in \hat{Q}} \tilde{g}(x)| \tag{2.5}$$

The difference with the MMD quantity is that for the discrepancy we have the constraint that $\tilde{g}(x) = \tilde{z}(x)^2$, and the maximization is over $\tilde{z}$ and not over $\tilde{g}$. The function $\tilde{g}$ plays the role of the worst-case loss function in both bounds.

Similar to the MMD quantity, we study the function $\tilde{g}$ maximizing Equation 2.5. See Figure 2.3. This figure uses the same example that was used to illustrate the MMD (see Figure 2.1) and was generated using the same approach. In particular note that we compute the discrepancy using empirical samples. The computation of $\tilde{g}$ for the discrepancy is described in appendix B.1.

We observe that the function $\tilde{g}$ in Figure 2.3 becomes as positive as possible on one of the distributions and as small as possible on the other. Because of this Equation 2.5 is maximized: the mean of the worst-case loss function $\tilde{g}$ on one distribution will be very large while on the other distribution it will be very small. Because of this the empirical average of $\tilde{g}$ will be as different as possible on the sets $\hat{P}$ and $\hat{Q}$. The empirical distributions $\hat{P}$ and $\hat{Q}$ together with the smoothness conditions on $\tilde{z}$ determine how large and small $\tilde{g}$ can become. Note that for this example which is symmetric the discrepancy can become positive on one of both distributions: this explains why the function $\tilde{g}$ sometimes is positive on the left distribution and sometimes on the right distribution in Figure 2.3. On which it becomes positive depends on the empirical samples obtained from $\hat{Q}$ and $\hat{P}$.

However, if one of the Gaussian distributions is broader than the other, the discrepancy will become positive on this broader distribution, since $\tilde{g}$ can become more positive on these empirical samples from this broader distribution since it has a larger window to increase and decrease (recall it needs this window to increase and decrease because of the smoothness conditions on $\tilde{z}$). The MMD can become positive or negative on either distribution. This is easy to show for the MMD, since if the function $\tilde{g}$ maximizes Equation 2.1, $(-\tilde{g})$ will always maximize Equation 2.1 as well. This is not the case for the discrepancy. If the function $\tilde{g}$ maximizes the discrepancy in Equation 2.5, the function $(-\tilde{g})$ for the discrepancy is instead not allowed, since the loss function cannot be negative.

Now we relate the function $\tilde{g}$ to the discrepancy quantity. We observe in Figure 2.3a that the

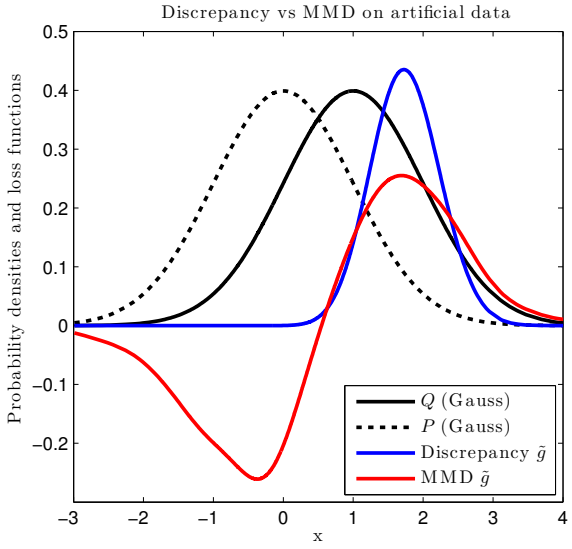


Figure 2.4: This figure illustrates the function $\tilde{g}$ maximizing Equation 2.5 for the discrepancy and the MMD.

function $\tilde{g}$ can become very large, this is because the distribution $\hat{Q}$ is very far away from $\hat{P}$, and thus $\tilde{g}$ can become as positive on $\hat{P}$ as possible. The function $\tilde{g}$ has a large window to increase and decrease. Therefore the empirical average of $\tilde{g}$ is very different on $\hat{P}$ and $\hat{Q}$ and consequently the discrepancy is relative large, see Equation 2.5. In Figure 2.3c there is only a small window where the density of P exceeds the density of Q, and thus $\tilde{g}$ has only a small window to increase. Because of the smoothness conditions on $\tilde{z}$ it cannot become very large. Because of this the empirical average of $\tilde{g}$ on $\hat{P}$ and $\hat{Q}$ is more similar, and the discrepancy quantity is smaller. Figure 2.3b illustrates the intermediate case.

Similar to the MMD, if the empirical distributions are more similar, the discrepancy is smaller. Thus like the MMD the discrepancy measures the difference between empirical distributions in some sense.

In Figure 2.4 we compare the functions $\tilde{g}$ of the MMD and discrepancy. Observe that the loss function $\tilde{g}$ of the MMD can become positive and negative, and because of this $\tilde{g}$ can possibly obtain a larger difference between the empirical means on $\hat{P}$ and $\hat{Q}$. The discrepancy loss function $\tilde{g}$ on the other hand must be positive since $\tilde{g}(x) = \tilde{z}(x)^2$. Now we directly observe that the discrepancy takes the squared loss more completely into account than the MMD: the worst-case loss function $\tilde{g}$ considered by the discrepancy is non-negative, as we would expect from a L_2 loss function. The worst-case loss function of the MMD can become negative, and therefore the worst case considered by the MMD might be overly pessimistic since the squared loss function can never become negative.

In appendix B.1 we show that for the squared loss in the linear kernel the discrepancy is given by:

$$\text{disc}(\hat{P}, \hat{Q}) = 4\Lambda^2 || \frac{1}{n_{\hat{P}}} X_{\hat{P}}^T X_{\hat{P}} - \frac{1}{n_Q} X_{\hat{Q}}^T X_{\hat{Q}} ||_2 = 4\Lambda^2 ||M||_2 \tag{2.6}$$

Here $||M||_2$ is the spectral norm of the matrix M, which is the same as the largest absolute eigenvalue of M since M is real and symmetric. Equation 2.6 indicates that if the covariance matrices of $\hat{P}$ and $\hat{Q}$ are similar as measured by the spectral norm of the difference of the matrices, the discrepancy measure will be small. In case kernels are used, the covariance matrices will be computed in the RKHS of K. In this case it is possible to express the discrepancy in terms of the kernel matrix, see appendix B.1.

We now discuss the meaning of Equation 2.6 in more detail using an artificial dataset where we use a model with a linear kernel, see Figure 2.5a which indicates the empirical distributions $\hat{P}$ and $\hat{Q}$. Typically the discrepancy is computed using a universal kernel such as the Gaussian

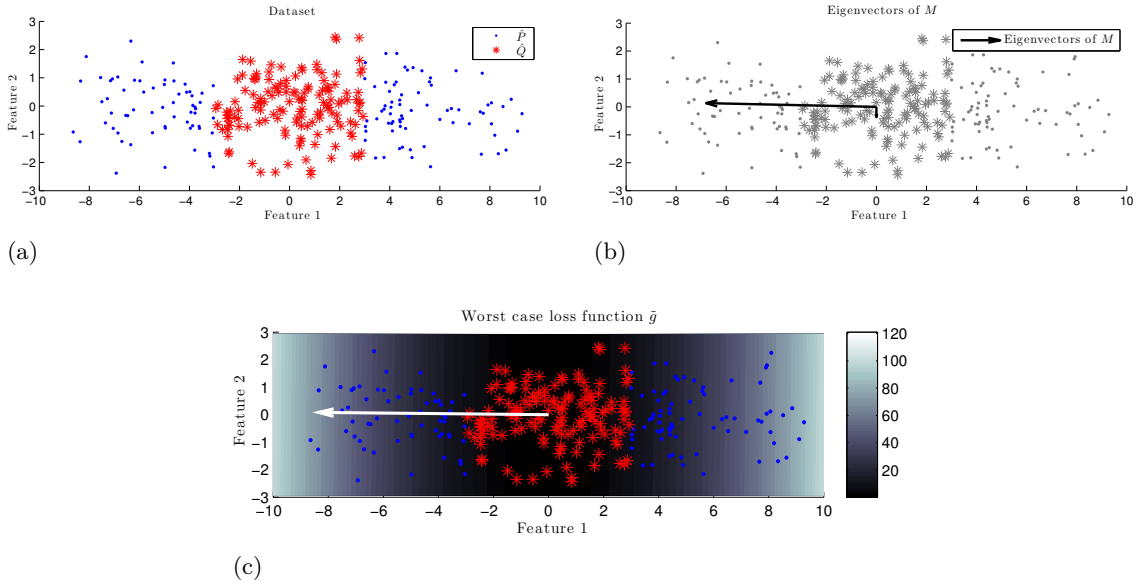


Figure 2.5: a) Description of the dataset. b) The eigenvectors of the matrix M. The length of the eigenvector is proportional to the squared eigenvalue. c) The eigenvector in the x-direction corresponds to the direction of $\tilde{z}$, from which we can compute the quadratic worst-case loss function $g(\tilde{x}) = \tilde{z}(x)^2$. The functional values of $\tilde{g}$ are displayed in this plot using shading.

kernel for reasons we will discuss in the next chapter, but for additional intuition as to what is happening this is an useful additional example.

The covariance matrices of both the set $\hat{P}$ and $\hat{Q}$ are subtracted to obtain M in Equation 2.6. M characterizes in which areas $\hat{Q}$ has a larger or smaller variance compared with $\hat{P}$. The discrepancy can be understood better in terms of the eigenvalue decomposition of M. The eigenvalue decomposition of M is similar to the PCA decomposition of a data matrix, only eigenvalues can be negative. Negative eigenvalues can occur because M is defined as the subtraction of two (positive semi-definite) matrices.

Recall that in active learning the dataset $\hat{Q}$ is the labeled set, and $\hat{P}$ is the unlabeled set. Ideally, we would want to train on $\hat{P}$, since this would correspond to labeling all data in the active learning setting. However, we have to train on $\hat{Q}$. The eigenvalues of M indicate 'what is wrong' with the dataset $\hat{Q}$ compared with the dataset $\hat{P}$. Positive eigenvalues of M indicate that $\hat{Q}$ has too little variance compared with $\hat{P}$ in the direction of the corresponding eigenvector, and negative eigenvalues indicate that $\hat{Q}$ has a too large variance compared with $\hat{P}$ in the direction of the corresponding eigenvector. See Figure 2.5b which indicates the eigenvectors of the matrix M for our artificial example.

We see that in the y-direction the variance of $\hat{P}$ and $\hat{Q}$ is almost equal causing the small eigenvalue for the eigenvector in the y-direction, but in the x-direction the difference is large indicated by the large absolute eigenvalue. We define the vector u as the eigenvector corresponding to the largest absolute eigenvalue of the matrix M. Recall that the discrepancy is determined by the spectral norm of M, or in other words by the largest absolute eigenvalue of M. In this example the eigenvalue that determines the discrepancy is positive, since the variance in the x-direction is larger for $\hat{P}$ than $\hat{Q}$.

The eigenvector u determines the function $\tilde{z}(x)$: $\tilde{z}(x) = u^T x$ of Equation 2.4, see Appendix B.1. $\tilde{z}$ is a linear function in this example since we use the linear kernel. From this function we can compute the worst-case loss function $\tilde{g}(x) = \tilde{z}(x)^2$ considered by the discrepancy, which is displayed in Figure 2.5c by shading. From this we see why this direction was chosen for the function $\tilde{z}$: in this direction the empirical average of $\tilde{g}$ on the set $\hat{P}$ is much larger than the empirical average on the set $\hat{Q}$, and therefore this direction maximizes Equation 2.5.

We now relate this example in the linear kernel to the example in the Gaussian kernel. This

discussion is quite technical, and may be skipped by the reader initially, since it is unnecessary to understand the rest of this thesis. However, this discussion offers additional insights into what the discrepancy is measuring if the Gaussian kernel is used.

In case we compute the discrepancy with a Gaussian kernel, this same process takes place in the RKHS of the Gaussian kernel. Recall that for the Gaussian kernel, the vectors $\psi(x)$ in the RKHS always have norm one. Because of this, positive eigenvalues of M in the Gaussian kernel indicate under sampling of $\hat{Q}$ compared with $\hat{P}$ in the RKHS in the direction of the vector $\tilde{z}$ (since the norm plays no role in the Gaussian kernel, only the sample density plays a role). Negative eigenvalues indicate oversampling of $\hat{Q}$ with respect to $\hat{P}$ in the RKHS in the direction of $\tilde{z}$. This over- or undersampled direction is also indicated by the large values of $\tilde{g}$ for the samples that are over- or undersampled. If we look back at the example in Figure 2.3 we can thus conclude the following. In Figure 2.3a the eigenvalue determining the discrepancy was positive (too little samples of $\hat{Q}$ with respect to the number of samples $\hat{P}$ near the distribution of P indicated by large values of $\tilde{g}$ near the distribution P), in Figure 2.3b it was negative (too many samples of $\hat{Q}$ with respect to the number of samples of $\hat{P}$ near the distribution of Q indicated by large values of $\tilde{g}$ near the distribution of Q), in Figure 2.3c it was positive.

2.2.2. Discrepancy Generalization Bound

In the last subsection we illustrated how the discrepancy measures the similarity between two empirical distributions and how it differs from the MMD measure. In this subsection we discuss the discrepancy generalization bound.

Multiple bounds are given in [4] to motivate the use of the discrepancy. However, we choose to give a generalization bound of [15] which is easier to derive and understand and which we found to be tighter in all our active learning experiments. Furthermore this generalization bound is directly comparable with the generalization bound of the MMD in the previous section. For a complete discussion of why we chose this bound, see Appendix B.3.

The discrepancy bound given in this section upper bounds the same quantity as the MMD for all $h \in H$:

$$|L_{\hat{P}}(h, f) - L_{\hat{Q}}(h, f)| \tag{2.7}$$

Refer to Subsection 2.1.2 for a detailed motivation why it is beneficial for this quantity to be small for all $h \in H$. See also Figure 2.2 on page 16 which illustrated why it is desirable for this quantity to be small.

It is relatively easy to derive the discrepancy bound in the realizable setting where the labeling function f is in the hypothesis set H of our model, which we briefly do here. Before selecting the next sample in active learning we don't know the resulting model h after training, and we don't know the labeling function f. We assume that the labeling function f is in our hypothesis set: $f = h' \in H$. Now we can bound Equation 2.7 by taking the worst-case functions h and h' in H:

$$|L_{\hat{P}}(h, f) - L_{\hat{Q}}(h, f)| \leq \max_{h,h' \in H} |L_{\hat{P}}(h, h') - L_{\hat{Q}}(h, h')| = \text{disc}(\hat{P}, \hat{Q})$$

Here we obtain the definition of the discrepancy. This gives us the discrepancy generalization bound for the realizable setting.

When we assumed $f \in H$, we may have made an approximation error. Taking this into account we obtain the following generalization bound:

Theorem 3 (Discrepancy generalization bound[15]) *Assume that for any $x \in \mathcal{X}$ and $h \in H$ that $L(h(x), f(x)) \leq C$. Then given any hypothesis $h \in H$ and a deterministic labeling function f we have that:*

$$|L_{\hat{P}}(h, f) - L_{\hat{Q}}(h, f)| \leq \mathit{disc}(\hat{P}, \hat{Q}) + 2C\eta_{\mathit{disc}}(\hat{P}, \hat{Q}, f)$$

Where $\eta_{\mathit{disc}}(\hat{P}, \hat{Q}, f)$ is given by:

$$\eta_{\mathit{disc}}(\hat{P}, \hat{Q}, f) = \min_{\tilde{f} \in H} \left(\frac{1}{n_{\hat{P}}} \sum_{x \in \hat{P}} |\tilde{f}(x) - f(x)| + \frac{1}{n_{\hat{Q}}} \sum_{x \in \hat{Q}} |\tilde{f}(x) - f(x)| \right)$$

This bound is slightly tighter than the bound given in [15] but is based on their derivations. We give the proof of this bound in appendix B.2.

The approximation term $\eta_{\text{disc}}(\hat{P}, \hat{Q}, f)$ is now included, since we have relaxed the assumption $f \in H$. The quantity $\eta_{\text{disc}}(\hat{P}, \hat{Q}, f)$ measures the error of approximating f by $\tilde{f} \in H$ on the sets $\hat{Q}$ and $\hat{P}$. In case we have that $f \in H$ the term $\eta_{\text{disc}}(\hat{P}, \hat{Q}, f)$ becomes zero and the bound only depends on the discrepancy measure. In a real world active learning setting $\eta_{\text{disc}}(\hat{P}, \hat{Q}, f)$ is impossible to measure since it depends on the labels of $\hat{P}$. Observe that it is much clearer when this approximation error vanishes, unlike the approximation error of the MMD in the bound of Theorem 2.

Like the MMD bound, we can also rewrite this generalization bound in a more familiar form:

$$L_{\hat{P}}(h, f) \leq L_{\hat{Q}}(h, f) + \text{disc}(\hat{P}, \hat{Q}) + 2C\eta_{\text{disc}}(\hat{P}, \hat{Q}, f)$$

Here we directly relate the performance on $\hat{P}$ to the empirical loss on $\hat{Q}$, the discrepancy measure and the approximation term. The analysis that was given in Section 2.1.2 for the MMD bound also holds for this bound which we will briefly repeat here. Assume $f \in H$ and thus $\eta_{\text{disc}}(\hat{P}, \hat{Q}, f) = 0$. The training algorithm will minimize the average loss on $\hat{Q}$. If the discrepancy is small, this small loss will generalize to the set $\hat{P}$. This equation also illustrates why we want the approximation error $\eta_{\text{disc}}(\hat{P}, \hat{Q}, f)$ to be small: if this approximation error is smaller the bound on the loss on $\hat{P}$ is tighter.

We may wonder why the discrepancy bounds the quantity of Equation 2.7 in absolute value. This allows the analysis of the bound that was given in Figure 2.2 in Subsection 2.1.2, and it allows us to compute the approximation term $\eta_{\text{disc}}(\hat{P}, \hat{Q}, f)$ for the discrepancy.

We can give a generalization bound on the generalization error on the distribution P with the same method discussed at the end of Subsection 2.1.2 which we will not repeat here. In this subsection we also argued why a small error on $\hat{P}$ will likely result in good generalization performance on the distribution P, which is the goal in our considered active learning setting.

2.2.3. Comparison with MMD Generalization Bound

The bound for the discrepancy given in the previous subsection is in the same form as the MMD bound in Section 2.1.2, and therefore these bounds are comparable. To simplify our comparison of the bounds we consider the realizable setting. The comparison between both bounds in this setting is novel and has not been considered yet in works of the discrepancy and the MMD.

In the realizable setting the approximation error of the discrepancy vanishes. In this setting if H' for the MMD is chosen to take into account the hypothesis set and the loss as described in Subsection 2.1.4, we are sure that $g \in H'$ and the approximation error of the MMD bound will be zero as well. In this case the bounds only depend on the discrepancy measure and MMD measure, and the bounds become easily comparable. In this case, we can rewrite the MMD measure as:

$$\text{MMD}(\hat{P}, \hat{Q}) = 4\Lambda^2 ||M||_F = 4\Lambda^2 \sqrt{\sum_i \lambda_i^2} \tag{2.8}$$

See appendix C.4 for the derivation. Here λ_i are the eigenvalues of the matrix M. We have seen in Subsection 2.2.1 that the discrepancy can be calculated using:

$$\text{disc}(\hat{P}, \hat{Q}) = 4\Lambda^2 ||M||_2 = 4\Lambda^2 \max_i |\lambda_i|$$

Comparing both equations above, we can show that the discrepancy quantity is always smaller or equal than the MMD quantity, and thus the discrepancy bound is always tighter than the MMD bound in this setting. The discrepancy bound should be especially favorable if there are many large absolute eigenvalues λ_i, since the MMD will become larger with each eigenvalue, while the discrepancy only depends on the largest eigenvalue. This situation occurs for example if in many directions the variances of $\hat{P}$ and $\hat{Q}$ are different.

Finally, we might wonder if the assumptions of the MMD are less strict than the assumptions of the discrepancy. In appendix C.7 we show that the assumptions of both bounds are equivalent

in this setting. Thus, from a theoretical standpoint, the bound of the discrepancy seems to be truly better in the case of the squared loss in the realizable setting if the MMD takes the hypothesis set and loss into account (which we argued is desirable), since under the same assumptions the discrepancy bound is tighter. Only in the agnostic setting the approximation error of the MMD might be smaller than the discrepancy approximation term, see our discussion in Appendix C.7.

Note that any bound in terms of the discrepancy can directly be generalized to a bound in terms of the MMD since the MMD upper bounds the discrepancy if the kernel K' is chosen to take the hypothesis set into account. Thus the discrepancy bound in Subsection 2.2.2 can directly be extended to a MMD bound. We give this bound below for completeness:

Theorem 4 (MMD Generalization bound that always holds (2)) *Assume that for any $x \in \mathcal{X}$ and $h \in H$ that $L(h(x), f(x)) \leq C$. Then given any hypothesis $h \in H$ and a deterministic labeling function f we have that:*

$$|L_{\hat{P}}(h, f) - L_{\hat{Q}}(h, f)| \leq MMD(\hat{P}, \hat{Q}) + 2C\eta_{disc}(f, \hat{Q}, \hat{P})$$

Where the MMD quantity is computed using the kernel $K'(x, x') = K(x, x')^2$, where K is the kernel of the learning algorithm, and where $\Lambda' = 4\Lambda^2$, and $\eta_{disc}(f, \hat{Q}, \hat{P})$ is given by:

$$\eta_{disc}(f, \hat{Q}, \hat{P}) = \min_{\tilde{f} \in H} \left(\frac{1}{n_{\hat{P}}} \sum_{x \in \hat{P}} |\tilde{f}(x) - f(x)| + \frac{1}{n_{\hat{Q}}} \sum_{x \in \hat{Q}} |\tilde{f}(x) - f(x)| \right)$$

The interpretation of the approximation term η_{disc} in this bound is especially more straightforward than the approximation term in the bound of Theorem 2, as we have discussed before.

2.3. Transductive Experimental Design (TED)

First we describe the objective of TED and review an interpretation of the TED objective that was given in [8] to better understand what the TED objective means. We give multiple examples to illustrate the TED objective. Then we discuss the generalization bound based on the TED objective introduced in [7]. Finally we compare the generalization bound of TED with the bounds of the discrepancy and the MMD and we highlight important differences that indicate that the TED bound may be more informative than the discrepancy and MMD bound.

2.3.1. TED Objective

In this section we discuss the TED objective and show multiple examples to illustrate the meaning of the TED objective.

TED uses a Bayesian linear regression framework and has as goal to minimize the prediction variance of the trained model during active learning. A zero centered Gaussian prior is set on the slope parameter w of the linear model and it is assumed labels are generated by a linear model perturbed with Gaussian noise. From these assumptions a model can be derived that has the same MAP (maximum a posteriori probability) predictions as the predictions of the ridge regression model considered in this work. The variance of the prediction can be computed using the posterior distribution of the model. The idea of TED is to select the examples that reduce the variance of the predictions on the set $\hat{P}$ the most. The criterion of TED can easily be extended to kernelized models.

The objective of TED depends explicitly on the model: it uses the analytical form of the solution of the model. If a ridge regression model with kernel K and regularization parameter λ is used, the objective is given by[8]:

$$\text{TED}(\hat{P}, \hat{Q}) = \text{tr}(K_{\hat{P}\hat{P}} - K_{\hat{P}\hat{Q}}(K_{\hat{Q}\hat{Q}} + \lambda n_{\hat{Q}} I)^{-1} K_{\hat{P}\hat{Q}}^T) \tag{2.9}$$

This objective is minimized by the TED active learner. We can say the following about this objective. The objective $\text{TED}(\hat{P}, \hat{Q})$ is proportional to the variance of the predictions on the set

$\hat{P}$ given that the model is trained on the set $\hat{Q}$ under the probabilistic assumptions listed above. This objective is easily visualized, since it is straightforward to compute it over the entire input space. To compute the objective value of a particular configuration of $\hat{P}$ and $\hat{Q}$, the variances of the predictions at all positions of all samples $\hat{P}$ need to be summed.

See Figure 2.6 on page 26 for an example where we plot the variance of the predictions over the entire input space. To generate this figure a Gaussian kernel was used. The dataset consists of two Gaussian distributed classes. In this figure three situations are shown: in the top two figures we use a small value for σ of the Gaussian kernel, in the middle figures a larger σ is used. We observe that the variance of the predictions is reduced in a larger area around labeled objects if a larger σ is used: this is because a labeled object has a larger area of influence if σ is larger. In the bottom figures a model is trained with the same value of σ as in the middle figures but a larger value of λ is used. We observe that if λ is larger, the magnitude of the model output decreases, and the reduction in prediction variance decreases as well.

Note that the objective value of TED is not simply the absolute value of the predictions: see the middle situation of Figure 2.6. Here between the two Gaussian distributions the decision boundary lies where the model output is zero. Observe that the prediction variance is not zero in the same location.

From Figure 2.6 we can deduce that TED will cover as many samples in 'black' (in the right plots). From this it follows that the TED active learner will aim to choose non-redundant samples that are similar to as many samples in $\hat{P}$. We can recognize this in Equation 2.9. When minimizing the TED objective for active learning, the factor $(K_{\hat{Q}\hat{Q}} + \lambda n_{\hat{Q}})^{-1}$ ensures dissimilarity among labeled samples, and the factors $K_{\hat{P}\hat{Q}}$ ensure similarity between the sets $\hat{Q}$ and $\hat{P}$. Furthermore the objective in this form is difficult to interpret mathematically.

In [8] the TED objective is rewritten as follows, making it easier to interpret:

$$\frac{1}{\lambda n_{\hat{Q}}}\text{TED}(\hat{P},\hat{Q}) = \min_{\alpha} \sum_{i=1,\ldots,n_{\hat{P}}} ||x_i - \alpha_i X_{\hat{Q}}||^2 + \lambda n_{\hat{Q}} ||\alpha_i||^2 \tag{2.10}$$

This assumes a linear kernel K is used[2]. The first term in the objective is an approximation error, and the second term arises due to the regularization of the ridge regression model. Studying the first term, we see that TED tries to approximate each object of the dataset $x_i \in \hat{P}$ with a linear combination of labeled objects $\hat{Q}$. The linear combination is determined by the vector α_i for each object x_i that is to be reconstructed. Thus note that this is a minimization over the matrix α, since $n_{\hat{P}}$ objects need to be reconstructed with $n_{\hat{Q}}$ objects. The error in the approximation is weighed by the l_2 norm. In the second term each vector α_i is regularized by the regularization parameter λ used in the ridge regression model.

This indicates that the TED active learner will prefer samples a large norm, that point in the direction of many samples of $\hat{P}$, so these samples can be reconstructed with small values of α. In Figure 2.7 on page 27 an illustration is given of which samples will be selected by TED in a linear kernel in 2D for samples generated by a single Gaussian distribution that is centered in the origin. Observe that the chosen queries have large norm, and do not point in the same direction: because of this they can be used to reconstruct the data $\hat{P}$ by linear combinations with relative small coefficients.

Note that we will perform most experiments in the Gaussian kernel. In a normalized kernel such as the Gaussian kernel, the norm of each object is the same in the RKHS. Because of this only the directions of the vectors in the RKHS will matter for the TED objective: TED will then select the objects that point in the direction of many unlabeled samples.

[2]If a different kernel is used, this objective is still valid, but then the featurevectors and datamatrices will be in the RKHS of the kernel K used for the learning algorithm.

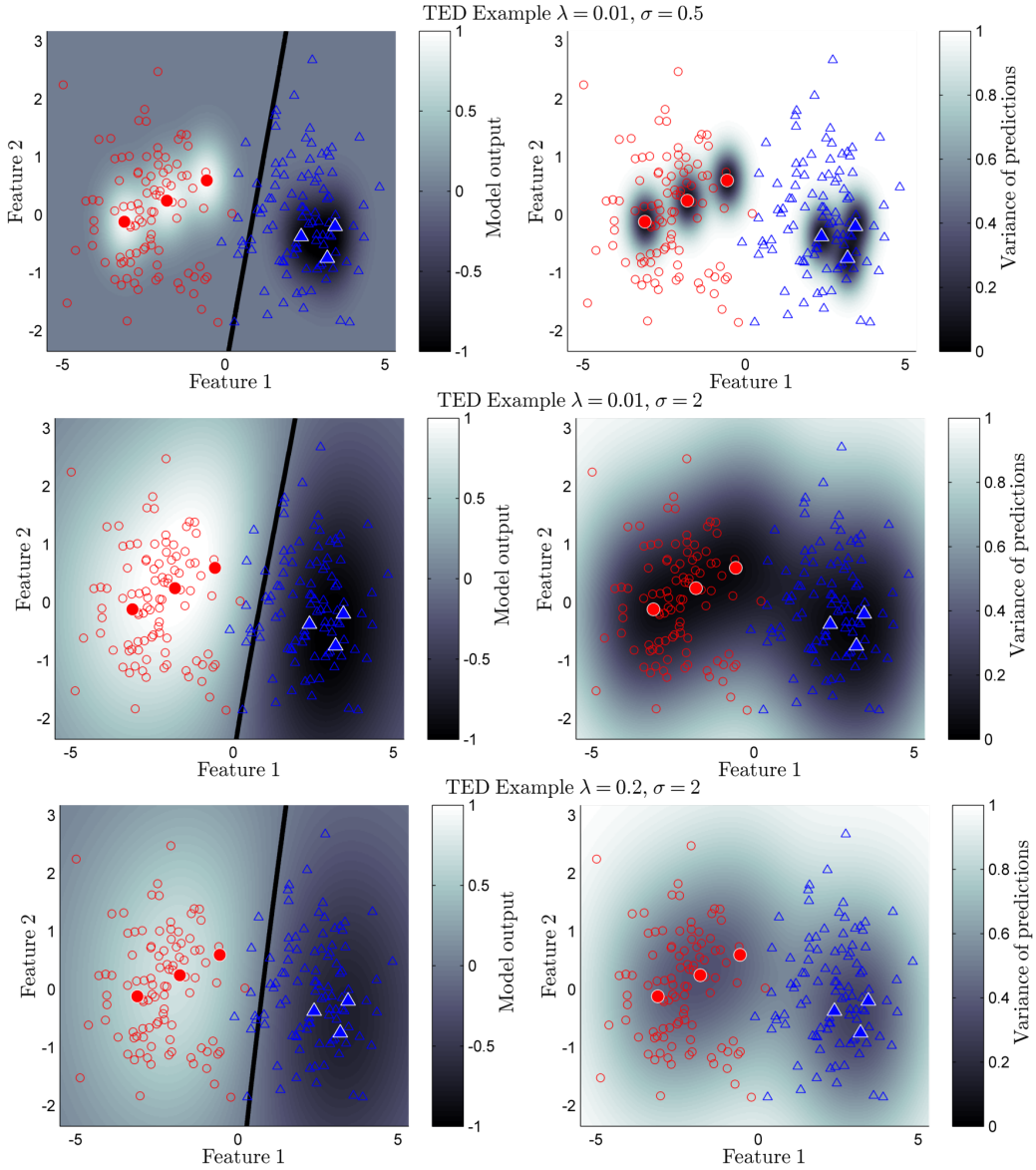


Figure 2.6: This figure illustrates the objective values of TED. The labeled objects are indicated by solid symbols (which corresponds to the set $\hat{Q}$), and unlabeled objects are indicated with outlines ($\hat{P}$). The left plots show the model output, the right plots show the variance of the prediction for each position. In the plot at the top a smaller value of σ is used then in the middle plot. In the lowest plot the same σ is used as in the middle plot, but the regularization parameter λ is increased. The TED active learner will choose samples in such a way the variance of all samples of $\hat{P}$ is as small as possible. In this figure this means as many samples should be 'covered in black' in the plots at the right. From this we can deduce that the TED active learner will avoid similar samples in $\hat{Q}$, and that the TED active learner will select samples in $\hat{Q}$ that are similar to many samples in $\hat{P}$.

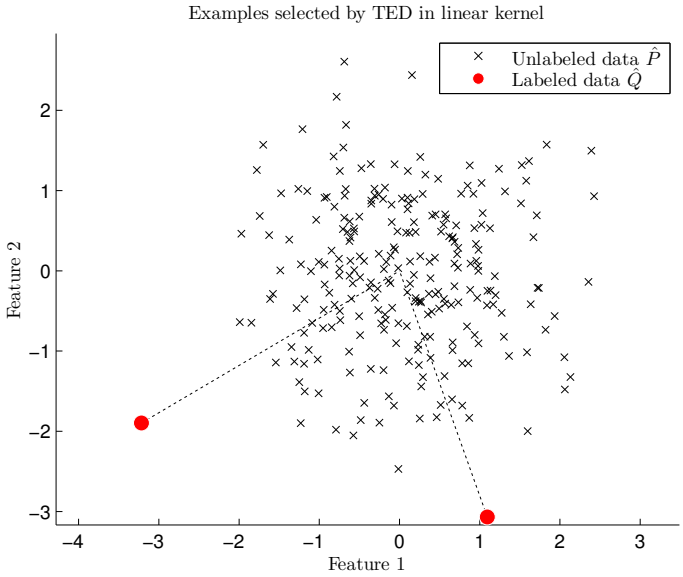


Figure 2.7: Illustration of chosen samples by TED in a linear kernel. These samples have the largest norm and are (somewhat) orthogonal, and thus can be used to reconstruct the data most efficiently using linear combinations according to Equation 2.10.

2.3.2. TED Generalization Bound

The bound in [7] assumes a Gaussian noise model. These assumptions are not appropriate for the classification setting, since in this setting the labels are binary. Therefore in this chapter we give a different novel bound that depends on the TED quantity. This bound is guaranteed to hold in the agnostic setting and does not assume any specific noise model. This bound is more comparable to the discrepancy and MMD bound. Before we give the theorem of this bound, we briefly discuss the intuition behind the generalization bound.

In our active learning setting our goal is to minimize the squared error on the set $\hat{P}$. If the model h is trained on the set $\hat{Q}$, we can write this quantity of interest as:

$$L_{\hat{P}}(f,h) = \frac{1}{n_{\hat{P}}}||f_{\hat{P}} - \bar{H}f_{\hat{Q}}||_2^2 \tag{2.11}$$

Here we use $f_{\hat{P}}$ to indicate the vector of labels on the set $\hat{P}$, and the vector $f_{\hat{Q}}$ as the vector of labels on the set $\hat{Q}$. Using the analytical solution of the ridge regression model, the outputs of the model h on the set $\hat{P}$ are given by the vector $\bar{H}f_{\hat{Q}}$. Here $\bar{H}$ is the so called hat matrix of the ridge regression model, see also Appendix H.

In the active learning setting we however do not know the vector $f_{\hat{P}}$, which makes it impossible to calculate Equation 2.11 in practice during active learning. Therefore, we consider a worst-case scenario over all possible f, and then we obtain the following bound on the mean squared error:

$$L_{\hat{P}}(f,h) \leq \max_f \frac{1}{n_{\hat{P}}}||f_{\hat{P}} - \bar{H}f_{\hat{Q}}||_2^2$$

This quantity is difficult to compute, since this is a maximization over many vectors f. In the binary classification setting, we know that the label of each object is either $+1$ or -1. Then even if we only maximize over $f_{\hat{P}}$ we need to consider 2^n possibilities, where n is the number of objects in $\hat{P}$. Thus the number of candidates of f that we need to maximize over is exponential in the number of objects in the set $\hat{P}$, and therefore this computation is difficult.

To make the computation easier, we may assume that we are in the realizable setting, thus $f \in H$. In this setting we can use the bound given in [8] to show that:

$$L_{\hat{P}}(f,h) \leq \max_{f \in H} \frac{1}{n_{\hat{P}}}||f_{\hat{P}} - \bar{H}f_{\hat{Q}}||_2^2 \leq \frac{\Lambda^2}{n_{\hat{P}}}\text{TED}(\hat{P},\hat{Q})$$

The work of [8] considers a different setting than this work, in Appendix D.1 we show how we have adapted their bound to our setting.

How is it possible to derive this bound, and where does it come from? Recall that since $f \in H$, the labeling function f is smooth. Recall that f generates the labels on the sets $\hat{Q}$ and $\hat{P}$, and therefore if the sets $\hat{Q}$ and $\hat{P}$ are more similar, the labels of $f_{\hat{P}}$ and $f_{\hat{Q}}$ are necessarily more similar as well because of the smoothness conditions on f. Using the analytical solution of the ridge regression model and these smoothness constraints on f we can quantify how well the model performs on the set $\hat{P}$ in a worst-case scenario.

In the agnostic setting we may have that $f \notin H$. Using the techniques of [15] we can relax the TED bound to conform to this setting. This novel generalization bound is given below:

Theorem 5 (Non stochastic TED generalization bound) *Assume that for any $x \in \mathcal{X}$ and $h' \in H$ that $L(h'(x), f(x)) \leq C$, where we assume that f is a deterministic labeling function. We can give the following generalization bound for the ridge regression model h trained on the set $\hat{Q}$ with regularization parameter λ:*

$$L_{\hat{P}}(h, f) \leq \frac{\Lambda^2}{n_{\hat{P}}} TED(\hat{P}, \hat{Q}) + 2C\eta_{TED}(\hat{P}, \hat{Q}, f)$$

Where $\eta_{TED}(f, \hat{P})$ is given by:

$$\eta_{TED}(\hat{P}, \hat{Q}, f) = \min_{\tilde{f} \in H} \left(\frac{1}{n_{\hat{P}}} \sum_{x \in \hat{P}} |f(x) - \tilde{f}(x)| + \frac{1}{n_{\hat{P}}} ||\bar{H}||_1 \sum_{x \in \hat{Q}} |f(x) - \tilde{f}(x)| \right)$$

Where $|\bar{H}|_1$ is the $p = 1$ operator matrix norm of the hat matrix used to compute the predictions on the set $\hat{P}$ of the model h trained on the set $\hat{Q}$ (see Appendix H) given by $||\bar{H}||_1 = \max_{1 \leq j \leq n_{\hat{Q}}} \sum_{k=1}^{n_{\hat{P}}} |\bar{H}_{kj}|$.

This theorem gives an upper bound on the expected squared loss on the set $\hat{P}$ of the ridge regression model h trained on the set $\hat{Q}$ with regularization parameter λ. Here the term $\eta_{\text{TED}}(\hat{P}, \hat{Q}, f)$ measures the approximation error when f is approximated by a hypothesis $\tilde{f} \in H$, which is similar to the approximation term of the discrepancy and MMD bounds. The approximation term of the TED bound only differs by the factor $|\bar{H}|_1$. Therefore we expect that the TED approximation error is comparable in size to the discrepancy approximation error[3]. In the realizable case we have $\eta_{\text{TED}}(\hat{P}, \hat{Q}, f) = 0$ since in that case $f \in H$ and no approximation is required.

Finally, similar to the discrepancy and MMD, if one is more interested in the performance of the model on the distribution P, a Rademacher generalization can be combined with the bound above to derive a bound on the performance on the distribution P. Since the set $\hat{P}$ is an i.i.d. sample from P, and since the set $\hat{P}$ is relatively large, the error on $\hat{P}$ is a good estimate of the generalization error. Therefore, a small error on $\hat{P}$ is desirable, since in that case the model will likely generalize well.

2.3.3. Comparison with Discrepancy and MMD

There are four key differences of the TED bound compared to the discrepancy and MMD bound, we discuss them in this subsection.

The first key difference is the form of the TED bound and its derivation. We have seen that the generalization bounds of the discrepancy and the MMD bound the difference between the average loss on the set $\hat{Q}$ and $\hat{P}$. TED does no such thing: it directly bounds the squared error of the model on the set $\hat{P}$. This indicates a completely different method is used to derive this bound. Note that the MMD and discrepancy bound are straightforward to adapt to models that use different loss functions. For TED this is not straightforward at all. Most models do not admit an analytical

[3]This factor $|\bar{H}|_1$ is likely only large if the model is very complex, since this factor is very large if the model predictions can become very large, see also the definition of this matrix norm in Section 1.2

solution required for bounds of this type. Furthermore the TED bound is especially derived for the ridge regression model, it might not be possible to use the techniques that are required to derive this bound for other models. This illustrates that TED is a less general bound.

The second key difference is that the TED bound is only valid for the model obtained after training on $\hat{Q}$, while the discrepancy and MMD bound are valid for any $h \in H$. Because the TED bound holds only for this model, it might be a tighter bound or more informative, since the discrepancy and MMD seem to be more general since they are valid for any $h \in H$. Furthermore, TED explicitly uses the analytical solution of the model, and therefore the TED criterion depends non-trivially on λ. The discrepancy and MMD bound only contain λ as multiplicative constant. Because of this the discrepancy and MMD bound result in active learners that do not depend on λ, while the active learning strategy of TED does depend on λ (see also Section 2.5). Therefore the TED bound might be more informative for active learning.

Furthermore, the TED bound is not symmetric with respect to $\hat{Q}$ and $\hat{P}$ while the discrepancy and MMD are symmetric with respect to $\hat{Q}$ and $\hat{P}$. Here we argue that the TED bound is therefore more desirable. To illustrate this consider two scenarios. Scenario one: we train on $\hat{Q}$ and evaluate on $\hat{P}$, and $\hat{Q} \in \hat{P}$. In the second scenario we train on $\hat{P}$ and evaluate on $\hat{Q}$, while still $\hat{Q} \in \hat{P}$. Since $\hat{Q} \in \hat{P}$, we would logically expect to perform better in the latter scenario since in that case the evaluation set is in the training set. In these situations the TED bound can be different for both scenarios since $TED(\hat{Q}, \hat{P}) \neq TED(\hat{P}, \hat{Q})$, where the first argument indicates the set trained on and the second argument indicates the set evaluated on. The MMD and discrepancy will for both these scenarios however give the same bound since they are symmetric measures, while TED will likely return a tighter bound when the model is trained on $\hat{P}$ than when is trained on $\hat{Q}$. Since TED can differentiate between these different cases the TED bound may be more informative.

Another important difference is the following. Recall that in deriving the discrepancy and MMD quantity, we essentially maximized over all $h \in H$ and over all possible labeling functions $\tilde{f} \in H$. In particular, in the worst-case scenario it is considered that $h = -\tilde{f}$ by the discrepancy and the MMD. This worst-case scenario considered by the discrepancy and the MMD is very conservative. In fact, in the realizable setting this worst-case scenario is impossible. The TED bound instead only maximizes over $\tilde{f} \in H$. The model considered by TED is obtained by using the analytical solution to train on $\tilde{f}$. Therefore, TED considers that in the worst case $\tilde{f} \approx h$. This worst case is much more probable to occur in the realizable setting than the worse case $h = -\tilde{f}$. Therefore, the worst case considered by TED may be more informative than the worst case considered by the discrepancy and the MMD generalization bounds. Therefore, the active learning strategy of the discrepancy and the MMD might be overly conservative in choosing samples.

Finally, one may wonder if it is possible to relate the TED quantity to the discrepancy or MMD quantity. However, observe that the TED quantity depends on the eigenvalues of the product of the covariance matrix of the set $\hat{P}$ and the inverse covariance matrix of the set $\hat{Q}$ with an additional regularization matrix (see equation D.2). The discrepancy and MMD quantities can be characterized in terms of the eigenvalues of the matrix M, which depends on the difference between the covariance matrices of the sets $\hat{Q}$ and $\hat{P}$. Therefore such a comparison is difficult and we have not been able to show a relation between these quantities.

2.4. Nuclear Discrepancy

In the previous section we have studied the TED quantity and its generalization bound. We have argued that the TED bound might be more informative for active learning, since it considers more realistic worst-case scenarios than the MMD and the discrepancy.

Inspired by the TED generalization bound and our results in Section 4.4, we propose a novel discrepancy measure: the nuclear discrepancy. We show using a probabilistic analysis that this quantity weighs the realistic worst-case scenario considered by TED more than the discrepancy and the MMD. Therefore a bound in terms of the nuclear discrepancy might be more informative for active learning, since it assigns more weight to more likely scenarios. We also give a non-probabilistic generalization bound in terms of the nuclear discrepancy, and show that this bound is actually looser than the MMD and discrepancy bound. This in contrast with our proposed

discrepancy active learner, which was motivated by a tighter generalization bound. Using the nuclear discrepancy we can confirm whether or not tighter generalization bounds will perform better for active learning.

2.4.1. Motivation

First, we derive a generalization bound that holds in expectation. We will argue in this section that such a bound will consider more realistic scenarios than the MMD and the discrepancy. This generalization bound shows us which quantity might be relevant to characterize the performance in average cases. We use this quantity to define the nuclear discrepancy in the next section.

The discrepancy measure is derived by taking the worst-case scenario over the models w and w', where the model w is trained during active learning and the model w' is assumed to generate the labels. This is done by defining a vector $u = w - w'$, and maximizing with respect to u. The MMD uses a similar approach to calculate the worst-case loss function.

With no prior knowledge on w and w', we could assume that u is a random vector. Actually, we know in the realizable case that $w \approx w'$, since w is trained on the labels that are generated by w', however without using the analytical solution of w this is difficult to characterize. Using the closed form solution of w is possible, then we will likely create an active learner that is very similar to TED. We are however more interested in deriving a bound that does not explicitly depends on the analytical solution, since such a bound might be more broadly applicable.

Thus we assume u is a random vector. We can define a probability density over u and compute a probabilistic generalization bound that holds in expectation. We aim to bound the same quantity as the discrepancy and the MMD. Recall that both bound the quantity:

$$|L_{\hat{P}}(w, w') - L_{\hat{Q}}(w, w')|$$

We discussed the motivation behind bounding this quantity in detail in Section 2.1.2. We bound the same quantity in expectation with respect to u:

$$\mathop{\mathbb{E}}_{u}\left[|L_{\hat{P}}(w, w') - L_{\hat{Q}}(w, w')|\right]$$

To compute this we can define a probability distribution over $u = w - w'$ that is for example concentrated around zero (to indicate that $w \approx w'$), such as a centered Gaussian distribution. Instead, we give a bound that holds for any distribution over u, making the bound more general.

However, to make the bound easy to compute we make one simplifying assumption. That is that each component $\bar{u}_i$ is independently and identically distributed, where $\bar{u}_i$ is the projection of u on the normalized eigenvector v_i of M corresponding to the eigenvalue λ_i. Essentially, all $\bar{u}_i$ are the components of u in the basis formed by the orthonormal eigenvectors of M. We indicate the vector u with respect to this basis as $\bar{u}$. Note that $\bar{u}$ is equal to u up to a rotation.

What does the assumption mean? It means that we assume that an error in estimating w' by w in each direction given by the eigenvectors v_i is distributed in the same way and that these errors in these directions are independent. Since in practice $w \approx w'$, we have that u is concentrated around zero. However, without using the analytical solution of w we cannot characterize in which direction u will point. Therefore, we assume instead that all directions are equally likely.

This way we are likely to account for the scenario that $w \approx w'$. Recall that the MMD and the discrepancy instead consider worst-case scenarios for u and the loss function, which may not capture the behavior that $w \approx w'$, since u then will not be concentrated around zero.

This analysis leads to the following novel probabilistic generalization bound:

Theorem 6 (Probabilistic nuclear discrepancy generalization bound) *We assume we are in the realizable setting where all labels are generated by a hypothesis $w' \in \mathcal{H}$. This bound holds for the model w that is trained on the set $\hat{Q}$. We define u as $u = w - w'$, and we define $\bar{u}_i$ as the projection of u on the eigenvector v_i of matrix M corresponding to eigenvalue λ_i. Assuming that all components of $\bar{u}_i$ are independently and identically distributed according to a distribution*

$p(\bar{u}_i)$, the following generalization bound holds in expectation:

$$\mathop{\mathbb{E}}_{\bar{u}}\left[|L_{\hat{P}}(w,w') - L_{\hat{Q}}(w,w')|\right] \leq \bar{C}\sum_i |\lambda_i|$$

Where λ_i are the eigenvalues of the matrix M, and $\bar{C}$ is a constant given by:

$$\bar{C} = \int \bar{u}_1^2 p(\bar{u}_1) d\bar{u}_1$$

The proof of this theorem is given in Appendix E. Our probabilistic generalization bound indicates that we should minimize all absolute eigenvalues of M during active learning if we care about the performance on average and if we want to account for the case $w \approx w'$.

The probabilistic assumptions required to derrive this bound may seem very strong. For example in the realizable setting we know that $w \in H$ and $w' \in H$, thus we know that the norm of u cannot exceed 2Λ. The independence assumption of all components $\bar{u}_i$ is for example not compatible with the previous fact. However, including the bound on the norm of u in our probabilistic generalization bound is difficult, and therefore we have applied this approximation. Our average-case analysis, while a rough simplification, is likely to include the case $w \approx w'$ more often than the worst-case scenario of the discrepancy and the MMD, and therefore may perform better.

Recall that the MMD HS active learner minimizes the quantity (see Equation 2.8):

$$\text{MMD}(\hat{P},\hat{Q}) = 4\Lambda^2\sqrt{\sum_i \lambda_i^2}$$

If we were to derive this quantity using a probabilistic analysis as given above, this would mean that we would assume that u is more likely to point in the direction of eigenvectors corresponding to large eigenvectors of M. We do not see why this would be likely, probabilistically, and therefore we think the average-case analysis considered above where the distribution of u is concentrated around zero is more appropriate. A similar argument illustrating the advantage of this probabilistic generalization bound can be given for the discrepancy. The discrepancy assumes that the vector u *only* points in the direction of the eigenvector that correspond to the largest absolute eigenvalue of M. This also therefore does not assume $w \approx w'$.

2.4.2. Non-Probabilistic Nuclear Discrepancy Generalization Bound

Inspired by our probabilistic analysis, we introduce the nuclear discrepancy quantity in this subsection. We show that this quantity upper bounds the MMD and discrepancy quantities. Afterward we introduce a novel non-probabilistic generalization in terms of the nuclear discrepancy, and we compare this generalization bound with the other generalization bounds given in this work.

Recall that the definition of the discrepancy is given by:

$$\text{disc}(\hat{P},\hat{Q}) = 4\Lambda^2 \max_i(|\lambda_i|)$$

It is straightforward to show that:

$$4\Lambda^2 \max_i(|\lambda_i|) \leq 4\Lambda^2 \sum_i |\lambda_i| \tag{2.12}$$

We define the quantity on the right hand side of Equation 2.12 as the nuclear discrepancy, or $\text{disc}_N(\hat{P},\hat{Q})$:

$$\text{disc}_N(\hat{P},\hat{Q}) = 4\Lambda^2 \sum_i |\lambda_i| \tag{2.13}$$

Given the probabilistic analysis in the previous subsection we expect that minimization of the nuclear discrepancy quantity can lead to better results for active learning.

The name nuclear discrepancy comes from the fact that this quantity can be calculated using the nuclear matrix norm of M:

$$\text{disc}_N(\hat{P}, \hat{Q}) = 4\Lambda^2 ||M||_*$$

Recall that since M and M_K (the kernel version of M) share the same eigenvalues (see Appendix B.1), the nuclear discrepancy can be directly applied to any arbitrary kernel by replacing the eigenvalues λ_i in 2.13 by the eigenvalues of the matrix M_K.

Observe that this quantity upper bounds the discrepancy quantity (see Equation 2.12):

$$\text{disc}(\hat{P}, \hat{Q}) \leq \text{disc}_N(\hat{P}, \hat{Q})$$

This quantity also upper bounds the MMD quantity if the MMD quantity takes the hypothesis set into account, since:

$$\text{disc}_N(\hat{P}, \hat{Q}) = 4\Lambda^2 \sum_i |\lambda_i| \geq 4\Lambda^2 \sqrt{\sum_i \lambda_i^2} = \text{MMD}(\hat{P}, \hat{Q})$$

This follows from the fact we can consider the MMD quantity as the l_2 norm of a vector containing all eigenvalues of M and the nuclear discrepancy quantity as the l_1 norm of a vector containing all eigenvalues of M. Since the l_2 norm for vectors is always smaller than or equal to the l_1 of a vector, the nuclear discrepancy upper bounds the MMD quantity.

Since the nuclear discrepancy upper bounds the discrepancy, we can directly give the following novel generalization bound based on the discrepancy generalization bound in Theorem 3:

Theorem 7 (Nuclear discrepancy generalization bound) *Assume that for any $x \in \mathcal{X}$ and $h \in H$ that $L(h(x), f(x)) \leq C$. Then given any hypothesis $h \in H$ and a deterministic labeling function f we have that:*

$$|L_{\hat{P}}(h, f) - L_{\hat{Q}}(h, f)| \leq disc_N(\hat{P}, \hat{Q}) + 2C\eta_{disc}(\hat{P}, \hat{Q}, f)$$

Where $\eta_{disc}(\hat{P}, \hat{Q}, f)$ is given by:

$$\eta_{disc}(\hat{P}, \hat{Q}, f) = \min_{\tilde{f} \in H} \left(\frac{1}{n_{\hat{P}}} \sum_{x \in \hat{P}} |\tilde{f}(x) - f(x)| + \frac{1}{n_{\hat{Q}}} \sum_{x \in \hat{Q}} |\tilde{f}(x) - f(x)| \right)$$

Recall that $\eta_{\text{disc}}(\hat{P}, \hat{Q}, f)$ is the approximation error which comes from the fact that we approximated the label function f by a function $\tilde{f} \in H$. Observe that this generalization bound is looser than the discrepancy generalization bound (Theorem 3) and MMD HS generalization bound (Theorem 4), since the nuclear discrepancy upper bounds the discrepancy and MMD HS quantity and since the nuclear discrepancy depends on the same approximation error. Just as for the discrepancy bound, if we are in the realizable setting we have that $f \in H$ and thus $\eta_{\text{disc}}(\hat{P}, \hat{Q}, f) = 0$. During active learning we cannot compute $\eta_{\text{disc}}(\hat{P}, \hat{Q}, f)$ since we require all labels of the dataset $\hat{P}$.

A comparison of this generalization bound with TED is not straightforward, because like the discrepancy and MMD HS bounds, the generalization bound given above depends on different quantities than the TED generalization bound. However, considering our probabilistic analysis in the previous subsection, we have argued that the nuclear discrepancy bound is more likely to account for the fact that $w \approx w'$. This is similar to TED, which also assumes that $w \approx w'$. Therefore, we may expect that the nuclear discrepancy active learner will perform more similar to the TED active learner.

In this section, we have introduced a new quantity: the nuclear discrepancy. We have shown using a probabilistic analysis that this quantity will weigh realistic scenarios more than the MMD HS and discrepancy quantity, similar to the TED active learner. Therefore, we argue that minimization of this quantity could be more desirable for active learning. Unlike the TED bound, the

nuclear discrepancy does not dependent on the analytical solution of the model, making it a more general quantity. We have introduced a generalization bound in terms of the nuclear discrepancy, and we have shown that the MMD HS and the discrepancy bound are tighter. However, will the nuclear discrepancy perform worse because its bound is looser, or will it perform better because it considers more realistic average-case scenarios? We investigate this issue in Section 4.5.

2.5. Active Learning Algorithms

In this section we briefly discuss how the bounds in the previous sections can be used to construct active learning algorithms. We compare the performance of five active learning methods based on the following bounds: the MMD, the MMD HS, the discrepancy, the nuclear discrepancy and the TED active learner. See Table 2.1 for a summary of the bounds, their features and assumptions.

Each active learner minimizes an objective of the form $\text{obj}(\hat{P}, \hat{Q})$ sequentially. The discrepancy active learner minimizes the discrepancy, the MMD active learner minimizes the MMD, etc... All active learners ignore the approximation errors in the bounds since these quantities cannot be estimated during active learning, since to compute these terms we require the labels of the set $\hat{P}$.

Sequential minimization of the objectives is performed as follows. The active learner can only select a sample from the unlabeled pool $\hat{U} = \hat{P} - \hat{Q}$. The labeled set $\hat{Q}$ can be empty initially. In each iteration each possible query is added to the set $\hat{Q}$ and the objective is computed. The sample s^* which minimizes the objective is selected for labeling by the active learner:

$$s^* = \arg\min_{s \in \hat{U}} \text{obj}(\hat{P}, \hat{Q} \cup s)$$

Note that for all active learners the constants Λ or Λ' do not need to be known. This is because these appear in front of the objectives as multiplicative constants and therefore in the equation above can be moved in front of the arg min: they do not influence which object is chosen by an active learner.

	Bound / strategy takes these properties of the learning algorithm into account			
Active learner	Hypothesis set (kernel of model)	Loss	Training procedure and regularization parameter	Assumptions
MMD	☐	☐	☐	loss function $g \in H'$
MMD HS*	☑	☐	☐	label function $f \in H$
Discrepancy*	☑	☑	☐	label function $f \in H$
Nuclear Discrepancy*	☑	☑	☐	label function $f \in H$
TED	☑	☑	☑	label function $f \in H$

Table 2.1: Overview of the bounds used to construct active learners. Active learners with a star (*) are novel active learning methods proposed in this work. Note that for the MMD HS, the discrepancy and the nuclear discrepancy the value of λ is taken into account to compute the bounds, however the active learning strategy is independent of λ, while the active learning strategy of TED does depend on λ.

3

Experimental Setup

In this chapter we discuss the experimental setup in detail that is used to perform the experiments in the next chapter.

For an overview of the datasets used in the experiments also refer to Figure 1.2. In our experiments we use a train set (65%) and test set (35%). The train set corresponds to the set $\hat{P}$, the labeled part of the train set is the set $\hat{Q} \in \hat{P}$. The active learner has no knowledge of the test set. The train and test set are drawn randomly, and thus can be assumed to be sampled i.i.d. from P. Unless otherwise specified we consider the case where the labeled set $\hat{Q}$ is initially empty. After each query, the labeled set is updated and the model is trained and evaluated.

We generate learning curves where the performance is plotted against the number of queries. The performance is measured in terms of the mean squared error on the test set. For each active learning experiment new training and test sets are generated. The experiments are repeated 100 times and the resulting learning curve is generated by averaging. The random active learner is ran 1000 times since it is cheap to evaluate. Error bars in learning curves represent the 95% confidence interval of the mean and is computed using the standard error. During one experiment the active learners can be compared since they share the same train and test set.

Often we summarize our results using tables as is done often in active learning. We use the same approach as [10]. In these tables we count the wins, ties and losses when several methods are compared. We use a paired two tailed t-test with $p = 0.05$ to compare the results of active learning methods. This is allowed because the active learners share the same training and test set. If a method significantly performs better than another method, this is counted as a win, if a method performs significantly worse this is counted as a loss, if the methods do not differ significantly this is counted as a tie. We compare the active learners each time after five queries.

In our experiments we actually measure the generalization error on a testset (an unseen set that is not $\hat{P}$) that is sampled from P. The bounds however do not give performance guarantees directly on the distribution P. However as discussed in the previous chapter, standard generalization bounds can be used to relate the performance on $\hat{P}$ to the performance of P. The relation between the performance on $\hat{P}$ and P is known: namely the performance on $\hat{P}$ generalizes to performance on P if the model complexity is chosen appropriately. In case the model is too complex or not complex enough, the model can overfit or underfit. We choose the hyperparameters of the model in such a way to avoid over and underfitting as detailed further on in this chapter.

The active learning methods are evaluated on artificial data and on the real world datasets shown in Table 3.1 on page 36. Most datasets originate from the UCI Machine Learning repository[2], the other benchmark datasets were collected by Gunnar Raetsch and were repackaged by Gavin Cawley and Nicola Talbot[3]. We limit ourselves to binary classification datasets. The datasets

[1] The original database of the Wisconsin Breast Cancer Databases from UCI
[2] Available at `http://archive.ics.uci.edu/ml/`
[3] Available at `http://theoval.cmp.uea.ac.uk/~gcc/matlab/#benchmarks`

Dataset	Objects	Positive class	Negative class	Dimensionality	Pool size
vehicles	435	218	217	18	283
heart	297	137	160	13	193
sonar	208	97	111	60	135
iris	100	50	50	4	65
thyroid	215	65	150	5	140
ringnorm	1000	503	497	20	650
ionosphere	351	126	225	33	228
diabetes	768	500	268	8	499
twonorm	1000	500	500	20	650
banana	1000	439	561	2	650
german	1000	700	300	20	650
splice	1000	541	459	60	650
breast[1]	699	458	241	9	454

Table 3.1: Characteristics of evaluation datasets.

vehicles and iris are actually multiclass datasets. We convert vehicles into a two class dataset by only comparing the classes saab and bus. To convert iris to a binary dataset we took the classes virginica and versicolor, since these classes are not linearly separable and therefore pose a difficult classification problem. We use these multiclass datasets because they are either well known or are used in other active learning works as well [6, 21, 22]. We subsampled the datasets ringnorm, twonorm, banana and splice to contain 1000 objects. All datasets are preprocessed so each feature has zero mean and a standard deviation of one. The active learners can request up to 50 labels maximally, except for the dataset iris since the set $\hat{P}$ is so small, we limit the active learner to 40 samples. In case the pool $\hat{P}$ is too small the differences between active learning methods become harder to distinguish because they have less choice in selecting samples.

In case we use real world data we use the Gaussian kernel for the learning algorithm because for these datasets it is unlikely f is well modeled by a linear function. The bounds in this study are the strongest if the labeling function f is well approximated by the hypothesis set, and therefore we use the Gaussian kernel since it is a universal kernel: it can approximate bounded and continuous labeling functions up to arbitrary precision in the sense of the infinity norm[23]. See also the discussion in Appendix G for more details.

We now describe how we set the hyperparameters. In most works such as [9] the parameters of the model and kernel are chosen by cross validation on the whole dataset. We avoid this because we have observed that if parameters are chosen in this way the model can overfit. This happens because of the following. The number of samples on which the model is trained is much larger in cross validation than in the active learning experiments. For example, when using 10-fold cross validation the model on the dataset ringnorm is trained with 900 samples. However in our active learning experiments we do not select more than 50 samples. Therefore these parameters might not work optimally with much less labeled samples. Therefore we use the following procedure to optimize the hyper parameters instead.

We repeat the following procedure multiple times. We randomly select 25 examples from the dataset and label these. We train a model on these samples and evaluate the mean squared error on all unselected objects.

The hyper parameters that give the best performance after averaging are used in the active learning experiments. We use 25 examples since this corresponds with the mean number of samples selected during an active learning experiment. In accordance with the popular LIBSVM [24] package we choose the regularization parameter from the set $\lambda \in \{10^{-15}, 10^{-14.6}, 10^{-14.2}, \ldots, 10^{5}\}$ and we choose the kernel bandwidth from the set $\sigma \in \{10^{-5}, 10^{-4.625}, 10^{-4.25}, \ldots, 10^{10}\}$. However, according to [25] it is useful to use quartiles of $||x_i - x_j||$ to estimate σ as well. Therefore we also include following sets of quartiles: $\{0.1, 0.2, \ldots, 0.9\}$ of $||x_i - x_j||$ for parameter selection of σ.

The above procedure to select hyperparameters is impossible in practice when using active learning since we start without any labeled data. Choosing the optimal parameters in active learning remains an open problem. Our procedure at least ensures that during active learning experiments no overfitting occurs.

To make the comparison between the active learners more meaningful, we often compare our active learners in the realizable setting. In this setting the bounds do not depend on any approximation terms as was shown in the previous chapter, and therefore these approximation terms do not influence our results. This way the bounds only depend on the quantities that are minimized by the active learners, and because of this the comparison of the active learners based on these quantities becomes more straightforward.

To make the datasets conform to the realizable setting, we fit a model to the whole dataset and use the model output as labels (similar to the approach of [4]). Sometimes we will refer to this model as the 'oracle model'. The parameters of the oracle model were determined using the procedure described above and are thus the same as the parameters used for the model that is trained during the active learning experiments. The model trained during active learning and the oracle model have to share the same parameters, otherwise we may not be in the realizable setting. Since we optimized the parameters of this oracle model, this model will likely have a small mean squared error, and thus the labeling function in the realizable setting remains similar to the original labeling function of the agnostic setting. Effectively this turns the classification datasets into a regression datasets without noise. In the agnostic setting we use the original binary labels of the dataset.

We use (kernel) ridge regression as learning algorithm without intercept term and class priors. We revisit the influence of this model choice in the discussion in Section 5.2. For all details and settings on how to reproduce the experiments in the next chapter, see Appendix L. Especially the experiments using artificial experiments may deviate from the experimental setup described above. However all benchmark experiments on real world data conform to the settings described above.

4

Experiments and Results

In this section we perform multiple experiments to answer the research questions posed in the introduction. Below we give a preview of the experiments we will perform.

In Section 4.1 we illustrate why label independent active learners can perform better than random sampling. We characterize when this is the case and explain the trends observed in the active learning curves.

In Subsection 2.1.4 we showed how the MMD can be adapted to take the hypothesis set into account. We investigate if this is useful for active learning in Section 4.2 using artificial and real world data in both the realizable and agnostic setting.

In Section 4.3 we perform the most important experiments to answer three of our research questions. We compare the performance of the MMD, the discrepancy and the TED active learner in the realizable setting on real world and artificial data. This comparison is the most meaningful since in this setting the approximation errors in all generalization bounds vanish.

First we investigate the differences between the query strategies of discrepancy, MMD and TED using an artificial dataset. We show that the MMD and discrepancy active learners query more redundant samples than TED. This generalizes to our benchmark datasets as well: TED consistently outperforms these other methods in the realizable setting.

In this section we also investigate the question: does a tighter bound guarantee better active learning performance? We compute and compare the bounds of all methods.

Furthermore, in this section we answer the main research question of this work: does the discrepancy active learner improve upon the MMD active learner as suggested by our theoretical analysis? In this setting we find the surprising result that the MMD active learner generally outperforms or matches the performance of the discrepancy active learner, while we have shown that the discrepancy generalization bound is tighter in the realizable setting.

In Section 4.4 we investigate these surprising results concerning the MMD and the discrepancy. We show using two artificial datasets that the discrepancy active learner can have trouble selecting non-redundant samples while the MMD does not suffer from this behavior. To explain this behavior we investigate the worst-case scenarios that are considered by the MMD and the discrepancy. Using this analysis we can explain the surprising results concerning the MMD and the discrepancy. These insights also motivate the nuclear discrepancy active learner.

In Section 4.5 we benchmark the nuclear discrepancy on real world data in the realizable setting. Recall that the nuclear discrepancy has a looser generalization bound. This section can thus once more confirm that tighter generalization bounds do not relate to better results in active learning.

In Section 4.6 we study all active learning strategies in the agnostic setting. We compare the agnostic and the realizable setting and show that in most cases the performance in the realizable setting is indicative for the performance in the agnostic setting.

4.1. Why Do these Methods Work?

Why and in what case do these non-adaptive active learning methods work? This is the question we aim to answer in this section. Furthermore we discuss the trends typically observed in active learning curves and try to explain them.

We answer these questions using a one dimensional dataset, displayed in Figure 4.1a (top). The dataset consists of 300 objects divided over N clusters. We use a Gaussian kernel for the learning algorithm. It is extremely likely that random sampling chooses objects for labeling from the same clusters multiple times, see Figure 4.1a (middle) for an example of queries made by random sampling. For $N = 9$ the chance of random sampling discovering the 9 clusters with 9 samples is $\frac{8}{9} \times \frac{7}{9} \times \ldots \times \frac{1}{9} \approx 1 \times 10^{-3}$, which is approximately a chance of 1 in 1000. The active learners studied in this work try to approximate the empirical labeled distribution $\hat{Q}$ with the empirical distribution of the whole dataset $\hat{P}$ in some sense. For this dataset, this leads these active learners to select one sample from each cluster, see Figure 4.1a (bottom). For this experiment we used the TED active learner, but using the discrepancy and MMD active learner gives similar results.

Another way to look at this result is the following. All active learners discussed in this work assume the labels are generated by a model from the hypothesis class. Because of this, the labeling function is assumed to be smooth. Therefore choosing one example from a cluster gives us a lot of information about the labels of the neighboring objects in that cluster. The active learner therefore chooses to sample all 9 clusters, since this will give the most information about all objects in all clusters. Thus the active learners aim to sample influential samples first. Furthermore, the active learners avoid choosing similar samples if possible: the active learners aim to avoid redundancy in the labeled dataset $\hat{Q}$.

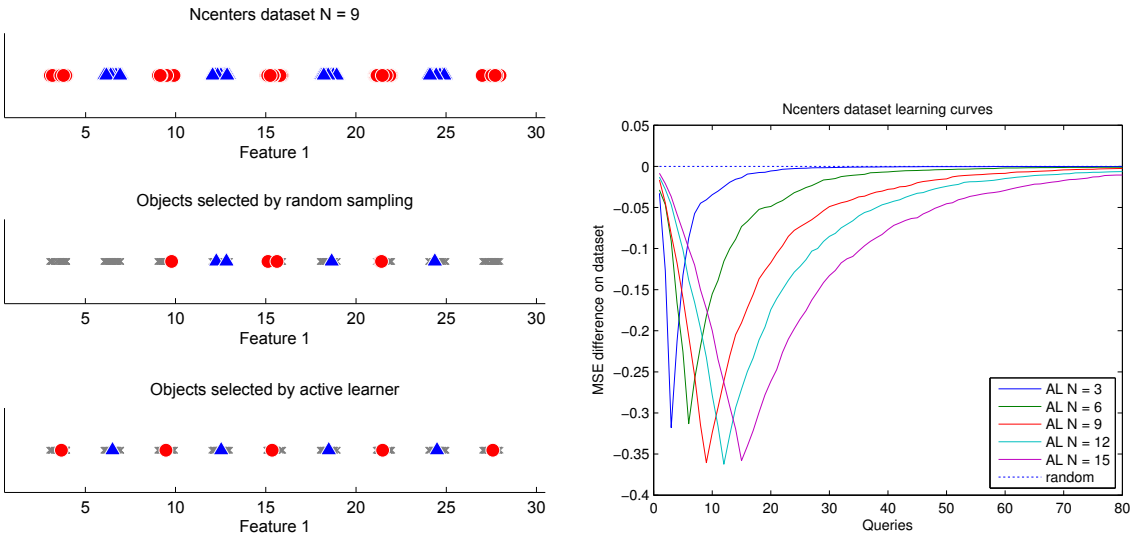
Because of this behavior of the active learners, the performance of the active learners is initially much better than random sampling until a point where all clusters are discovered, see the learning curve in Figure 4.1b. After this point, the performance difference between random and active learning will become smaller since random sampling will over time discover all clusters as well. These trends (the initial dip in the learning curve) are observed for real world datasets as well.

We see that the dip in the curve depends on the value N, the number of clusters. Increasing the number of clusters will benefit the active learner: because there are more clusters, it will take random sampling longer to discover them all. The more clusters there are, the longer the initial phase of the active learner where its performance is increasing with respect to random sampling, because it will need more queries to discover all clusters. This indicates that if the distribution P or its empirical sample has more modalities (in terms of the distance measure σ of the Gaussian kernel) these active learners have a larger advantage over random sampling.

Finally, we illustrate what happens if there is non-influential data as well. We use the same dataset as before, with $N = 9$ centers. However, now we distribute 300 samples over the first 3 clusters, and we distribute 30 samples over the last 6 clusters. The dataset is illustrated with the histogram in Figure 4.2 (top). We examine the first 3 queries. What we observe is that random sampling might sample one of the clusters with little samples, see Figure 4.2 (middle). The chance that this happens is approximately $1-(\frac{300}{330})^3 \approx 75\%$ in this example. Selecting one of these samples is undesirable, because if the sampled cluster has little samples, selecting a sample from one of these clusters will give us little information about the majority of all unlabeled objects. Selecting data from these small clusters therefore likely decreases the generalization error only little.

The active learners in this work sample will sample the most influential objects first, see Figure 4.2 (bottom). The active learners sample these dense clusters first, because these labeled samples give the most information about *all* unlabeled objects.

These artificial experiments explain why active learning using these methods is beneficial. The active learning methods in this work aim to find 'influential' examples and samples these first and avoid selecting redundant samples. The influence of a labeled sample depends on the kernel of the model: for the Gaussian kernel this means if samples are close (as measured by the kernel bandwidth σ) they influence each other's labels more. It however requires that the labeling function of the dataset is well approximated by the hypothesis class of the model.



(a) (b)

Figure 4.1: a) Ncenters dataset, N indicates the amount of clusters (top). Circles and triangles indicate positive and negative class, respectively. Grey crosses indicate unlabeled samples. Random sampling doesn't always directly find all relevant clusters (middle), while the studied active learning methods do (bottom). b) Learning curves of active learner performance compared to random sampling with varying values of N.

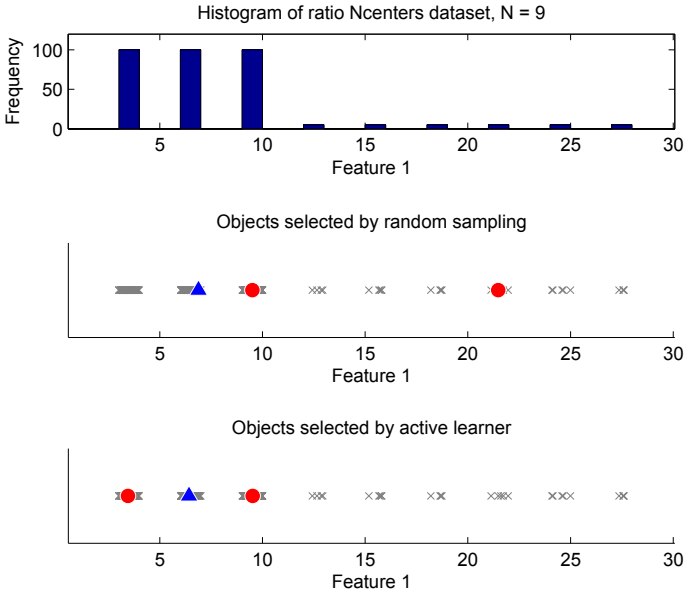


Figure 4.2: In this example we use the same dataset as in Figure 4.1a with $N = 9$, the number of samples in each cluster is displayed in the histogram (top). Random sampling might query objects in clusters with little samples and therefore gains little information about the labels of all unlabeled objects (middle). The studied active learners choose the most influential examples first (bottom).

4.2. Is It Useful to Take into Account the Hypothesis Set and Loss of the Learning Algorithm?

In the last subsection we saw why these active learning methods provide benefits compared to random sampling: in this section we will focus on the MMD active learning method. In Section 2.1.4 we showed how the MMD can be adapted to take the hypothesis set and loss into account: meaning how the kernel for the computation of the MMD should be chosen based on the kernel and loss of the learning algorithm. In this section we answer the question: is it useful to take the hypothesis set and loss into account for active learning using the MMD? We first show using an artificial example what can go wrong if we don't take the hypothesis set and loss of the learning algorithm into account when using MMD for active learning. Finally, we validate our hypothesis that taking the hypothesis set and loss into account is beneficial on real world data in the realizable setting and agnostic setting.

4.2.1. Artificial Dataset

To demonstrate the benefit of taking the hypothesis set and loss of the model into account we use the artificial dataset in Figure 4.3a. We use a linear kernel for our learning algorithm and we assume we are in the realizable setting. The majority of all samples are in the left and right clusters. The selected samples should give our model some information about the labels of these samples. There are 3 other samples in the middle of the dataset placed exactly on the line $x = 0$. These samples are unimportant for learning our model since they all have a zero x-component, thus will provide no information about the labels of the samples in the dense clusters.

We illustrate what can go wrong if the hypothesis set and loss is not taken into account by looking at the first example that is chosen for labeling by all methods. We use two MMD active learners. The first MMD active learner, MMD median, does not take the hypothesis set into account. For MMD median we use a Gaussian kernel where σ is set to the median distance in the dataset as suggested by [14]. For this dataset $\sigma_{\text{median}} \approx 2.2$. The second MMD active learner, MMD HS, uses a quadratic kernel and thus takes the hypothesis set and loss into account according to our analysis in Subsection 2.1.4. We also compare with the discrepancy and TED active learners, which both naturally take the hypothesis set and loss of the model into account.

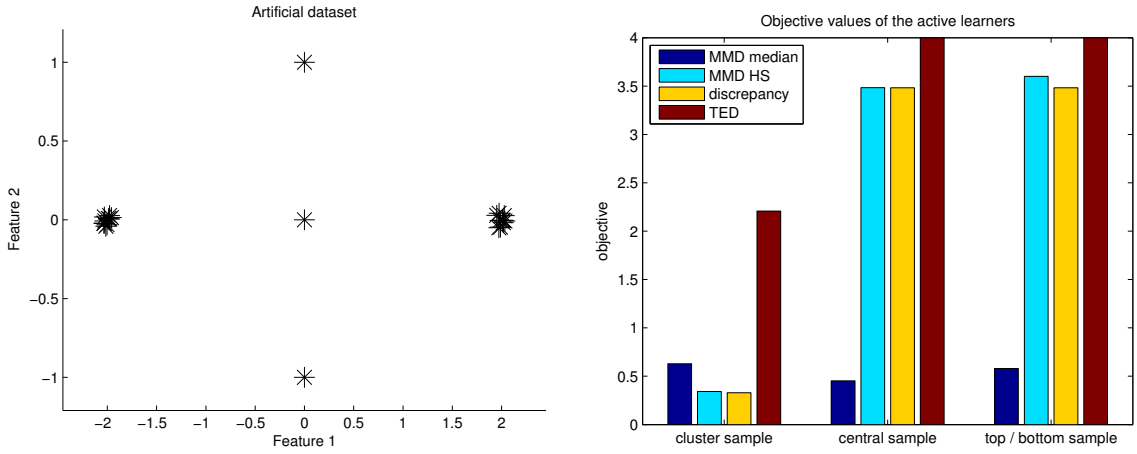
The objective values for all active learners for all samples are shown in Figure 4.3b[1]. The active learners will select the sample with the lowest objective. Because the samples on the line $x = 0$ are relatively close to all samples, these samples are judged to be most informative for MMD median. However, this fails to take into account the informativeness of these samples in the kernel of our learning algorithm. Therefore the sample selected by MMD median in the first iteration is suboptimal. The other methods, MMD HS, discrepancy and TED, that do take into account the hypothesis set and loss of the learning algorithm, do detect that these central samples are not informative.

It is possible to perform a similar experiment with this dataset with a model that uses a Gaussian kernel, but where the MMD uses a different value of σ than $\frac{\sigma}{\sqrt{2}}$ as suggested by our theoretical analysis in Subsection 2.1.4. If for example the MMD is calculated with a too large value of σ (such as the median considered here), it would sample the objects on the y-axis. If our model instead uses a value of σ which is much smaller, this query might not be informative at all since it's too far away from the clusters.

The reverse is also true: for a model with a large σ the objects on the y-axis are informative for both clusters and in this case these would be the best objects to query. If in this case a small σ is used to compute the MMD the MMD will prefer samples from the clusters themselves, while a central example would be more informative for the whole dataset.

In conclusion: we have illustrated using this artificial example what can go wrong if the hypothesis set and loss is not taken into account for the MMD in active learning.

[1] Note that the TED objective values are capped, originally the two longest bars extended to objective values of 80



(a) (b)

Figure 4.3: a) Artificial dataset. The labels are irrelevant for this example. The samples in the center are exactly placed on the line $x = 0$. b) The objective values for the active learners. The active learners all select the object with the smallest objective value. All methods prefer an informative sample from one of the clusters, except MMD median, which does not take the hypothesis set into account, and therefore does not notice the samples in the center are not informative for the model.

4.2.2. Real World Data

We now will evaluate whether taking into account the hypothesis set and loss is useful for active learning on real world data. We consider two settings: the realizable case, where $f \in H$, and the agnostic case, where $f \notin H$. We evaluate on the datasets listed in Table 3.1 and we use the Gaussian kernel for the learning algorithm and the MMD.

Our theoretical analysis in Section 2.1.4 suggests to set $\sigma_{\text{MMD}} = \frac{\sigma_{\text{RR}}}{\sqrt{2}}$ in order to take the hypothesis set and loss into account, where σ_{RR} is the kernel width used by the learning algorithm. We call this active learner MMD HS for hypothesis set. A different obvious choice is to choose $\sigma_{\text{MMD}} = \sigma_{\text{RR}}$ as discussed in Section 2.1.3, we call this active learning method MMD RR, since here the same bandwidth is used for the MMD and the ridge regression model. We also use an MMD active learner where we set σ_{MMD} to the median distance between all samples in the dataset as suggested by [14], we call this active learner MMD median. This active learner does not take the hypothesis set and loss into account at all.

First we compare these three methods in the realizable case where $f \in H$. We use the method described in Chapter 3 to make the datasets conform to this setting. In this case the assumptions of the bound for MMD HS are satisfied. For all datasets we have $\sigma_{\text{RR}} > \frac{\sigma_{\text{RR}}}{\sqrt{2}}$ and $\sigma_{\text{median}} > \frac{\sigma_{\text{RR}}}{\sqrt{2}}$, and therefore the assumptions of the MMD bound may not be satisfied if σ_{RR} or σ_{median} are used to compute the MMD since these bounds assume that the loss function is too smooth. Therefore these methods may perform worse.

Some illustrative learning curves are shown in Figure 4.4 on page 46, for all learning curves see Figure K.1 in the appendix. To summarize the results on all datasets we use a paired two tailed t-test with $p = 0.05$. We count the number of wins, ties, and losses for the method MMD HS compared to MMD median and MMD RR, the results are shown in Table 4.1 on page 45.

In these experiments the active learner MMD HS seems to perform better on most datasets than the other MMD active learners: on `vehicles`, `sonar`, `ringnorm`, `diabetis`, `twonorm`, `banana`, `german` and `breast` it almost always has a lower mean squared error, see Table 4.1. Only on the datasets `thyroid` and `ionosphere` in the beginning MMD HS performs worse, but later in the active learning process it matches the performance of the other methods, see also Figure 4.4. MMD HS seems to have a slightly larger advantage over MMD median than over MMD RR. This is to be expected, since MMD RR has a value of σ which is generally closer to the value of σ used by MMD HS, and therefore MMD RR may make smaller errors when approximating the true loss function.

We find it quite surprising that MMD HS actually can outperform MMD RR significantly, given that the values of σ are so close.

However, it may be expected that MMD HS performs better, since in this experiment its assumptions are exactly satisfied. To verify that taking the hypothesis set and loss into account is also useful on real world data we perform the same experiment in the agnostic case where we use the original binary labels of the datasets. Some resulting learning curves are shown in Figure 4.5 on page 47 and we use the same method to summarize the results on all datasets in Table 4.2. For all learning curves see Figure K.2 in the appendix.

In the agnostic case we see that the advantage of MMD HS is generally smaller, which is to be expected since in this case the assumptions of the MMD bound do not strictly hold anymore. Yet it still clearly outperforms the other methods on the `ringnorm` and `banana` dataset — even when compared to MMD RR which uses a value of σ that is very similar. MMD HS still performs slightly worse on `ionosphere` and `thyroid` in the beginning of the active learning experiments as in the realizable case. On most other datasets MMD HS has largely lost its advantage and now performs similar to the other methods as can be seen in Table 4.2. This can be seen in the learning curves as well. For example on `sonar` MMD HS loses a little of its advantage, and on the dataset `german` it completely loses its edge, to see this compare Figure 4.4 and 4.5.

We can conclude the following. Taking the hypothesis set and loss into account is more important on some datasets and less important for others in a real world setting. The advantage of taking the hypothesis set into account is the largest in the realizable setting. Even in case the same value of σ is used for the model and the MMD, it is more beneficial to choose σ_{MMD} according to our theoretical analysis as $\sigma_{\mathrm{MMD}} = \frac{\sigma_{\mathrm{RR}}}{\sqrt{2}}$.

Dataset	MMD HS vs MMD median	MMD HS vs MMD RR	σ_{RR}	σ_{median}
vehicles	2/8/0	2/8/0	5.3	5.3
heart	5/5/0	10/0/0	5.9	5.0
sonar	10/0/0	9/1/0	7.1	10.2
iris	4/4/0	4/4/0	2.3	2.3
thyroid	0/8/2	0/8/2	1.7	1.7
ringnorm	10/0/0	10/0/0	1.8	6.2
ionosphere	0/6/4	0/6/4	4.7	7.8
diabetes	10/0/0	6/4/0	3.0	3.6
twonorm	10/0/0	8/2/0	5.3	6.2
banana	10/0/0	10/0/0	0.6	1.8
german	10/0/0	9/1/0	4.2	6.0
splice	0/9/1	0/10/0	9.5	10.9
breast	9/1/0	10/0/0	4.2	3.6

Table 4.1: Win / tie / loss counts for MMD HS versus MMD median and MMD RR during the whole active learning process in the **realizable setting** where $f \in H$.

Dataset	MMD HS vs MMD median	MMD HS vs MMD RR	σ_{RR}	σ_{median}
vehicles	1/9/0	1/9/0	5.3	5.3
heart	0/10/0	1/9/0	5.9	5.0
sonar	4/6/0	6/4/0	7.1	10.2
iris	1/6/1	1/6/1	2.3	2.3
thyroid	0/8/2	0/8/2	1.7	1.7
ringnorm	10/0/0	10/0/0	1.8	6.2
ionosphere	0/6/4	0/6/4	4.7	7.8
diabetes	2/8/0	1/9/0	3.0	3.6
twonorm	2/8/0	2/8/0	5.3	6.2
banana	9/1/0	10/0/0	0.6	1.8
german	0/10/0	0/10/0	4.2	6.0
splice	0/9/1	0/9/1	9.5	10.9
breast	0/8/2	6/4/0	4.2	3.6

Table 4.2: Win / tie / loss counts for MMD HS versus MMD median and MMD RR during the whole active learning process in the **agnostic setting** where $f \notin H$.

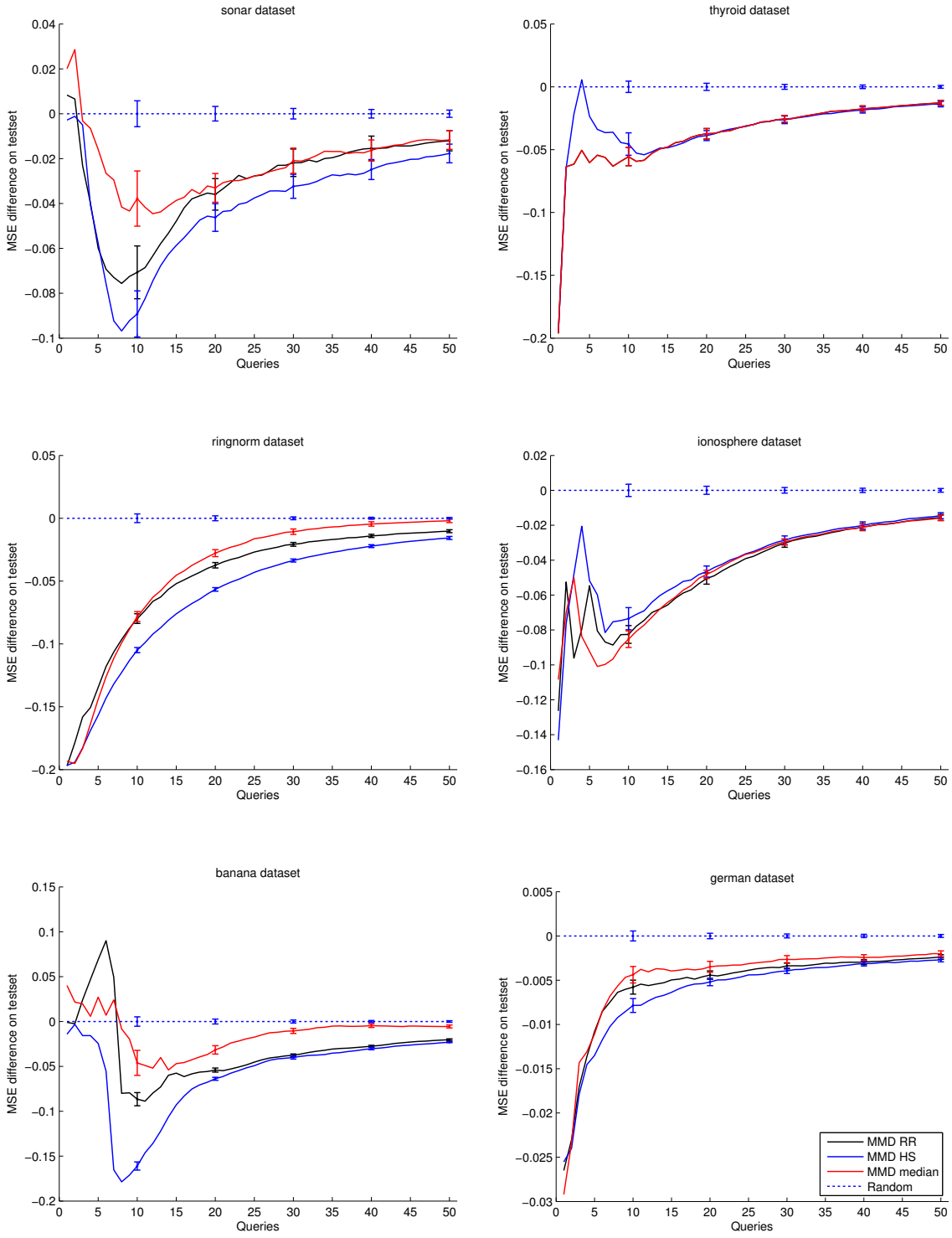


Figure 4.4: Comparison between MMD HS which takes the hypothesis set into account according to our theoretical analysis and MMD median and MMD RR which do not on six benchmark datasets in the **realizable setting** where $f \in H$. For all learning curves see Figure K.1 in the appendix.

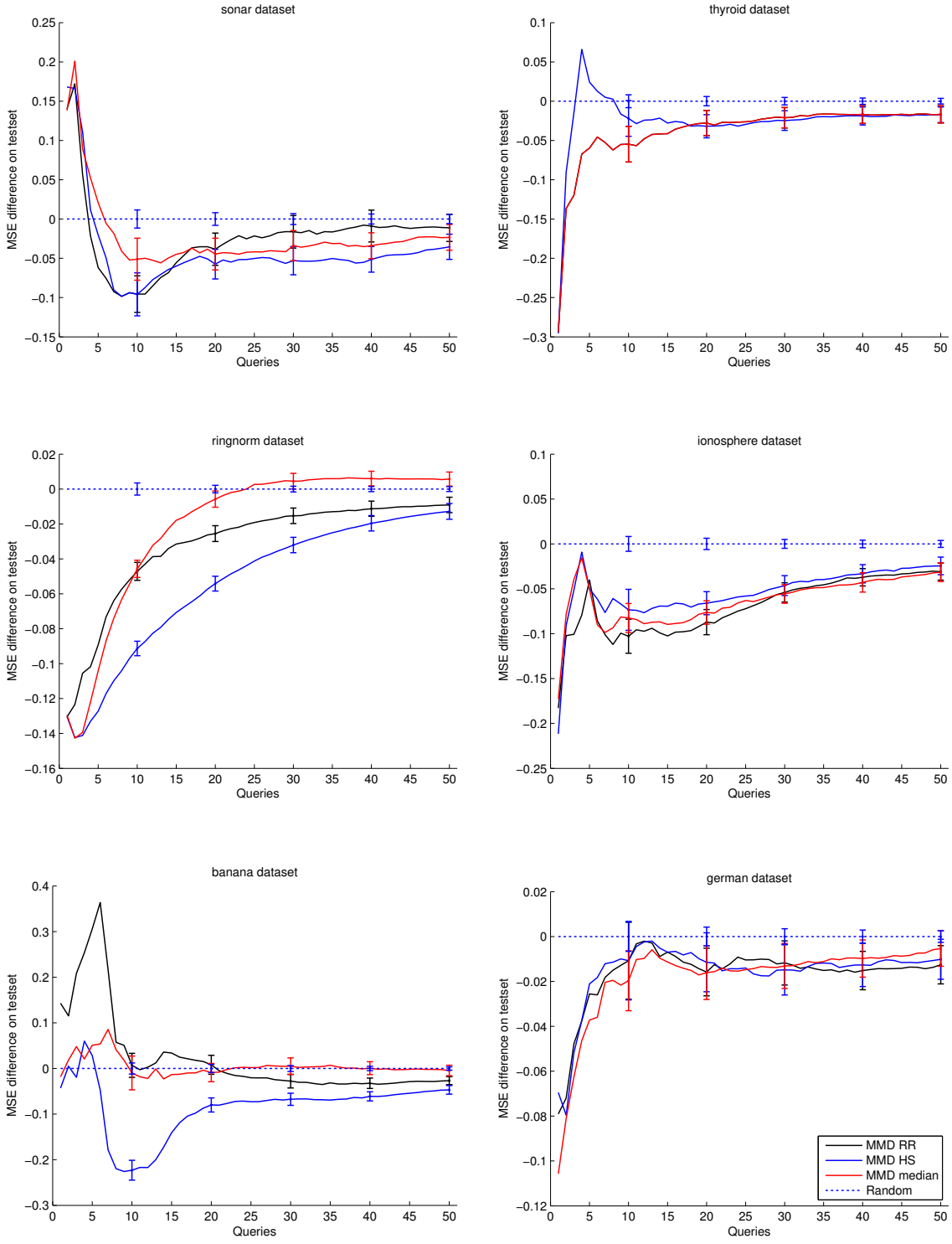


Figure 4.5: Comparison between MMD HS which takes the hypothesis set into account according to our theoretical analysis and MMD median and MMD RR which do not on six benchmark datasets in the **agnostic setting** where $f \notin H$. For all learning curves see Figure K.2 in the appendix.

4.3. Performance Comparison in the Realizable Setting

In the last section we focused on the individual active learning method of the MMD. In this section we move on towards comparing the performance of the MMD, the discrepancy and the TED active learner. We defer the comparison with the nuclear discrepancy to Section 4.5. We perform this comparison in the realizable setting since this is the most meaningful setting to study the behavior of the generalization bounds, since in this setting the approximation errors of all bounds vanish and all behavior is captured by the main term of the bounds which is minimized by the active learners. We answer three of our posed research questions including the main research question of this work. In the next section we introduce the nuclear discrepancy active learner.

We begin our investigation on an artificial dataset, where we illustrate the differences in the sampling strategy of the discrepancy and MMD compared to TED. Afterward we compare their performance on real world data in the realizable setting. We perform these experiments to gain insight in how the MMD and discrepancy compare with the TED active learning strategy. Using this experiment we answer our main research question: will our introduced discrepancy active learning strategy improve upon MMD active learning as suggested by our theoretical analysis? Finally we explicitly compute the values of the generalization bounds and we investigate if a tighter bound guarantees better active learning performance.

4.3.1. Artificial Dataset

To illustrate the differences between the discrepancy and the MMD compared to TED we use an artificial dataset. We first describe the dataset and setting and afterward discuss the results.

In this setting we use a linear kernel. The artificial dataset is eight dimensional and contains eight clusters. The location of cluster one is given by $[1, 0, 0, 0, 0, 0, 0, 0]$, the location of cluster two is given by $[0, 1, 0, 0, 0, 0, 0, 0]$, etc.... All objects within a cluster share the same label and position. The dataset contains 510 objects, and the number of objects in cluster i is 2^i. This dataset is illustrated in Table 4.3. We call this dataset the 'ratio dataset'.

We consider the realizable setting where the labeling function $f \in H$. For this dataset, MMD HS (using a quadratic kernel) and the discrepancy choose exactly the same objects. Therefore in our results we only display the performance of the discrepancy active learner and the TED active learner.

We use a small regularization parameter. To perform optimally on this dataset, with the knowledge that $f \in H$ (as all methods assume), an active learner should sample each cluster once, because then it can reconstruct f (almost[2]) perfectly.

This corresponds exactly to the strategy of TED: in the first eight iterations all eight clusters are sampled, in order of the number of objects in each cluster. Thus the first eight selected objects are in the clusters: `8,7,6,5,4,3,2,1`. Therefore TED performs optimal on this artificial dataset. See also Figure 4.6a on page 50 for the selected samples of TED after eight iterations and the learning curve of TED in Figure 4.6b.

[2]Because of the small regularization term it is impossible to reconstruct f perfectly.

Cluster	Number of objects	Relative frequency
1	2	0.0039
2	4	0.0078
3	8	0.0157
4	16	0.0314
5	32	0.0627
6	64	0.1255
7	128	0.2510
8	256	0.5020

Table 4.3: Description of the ratio dataset.

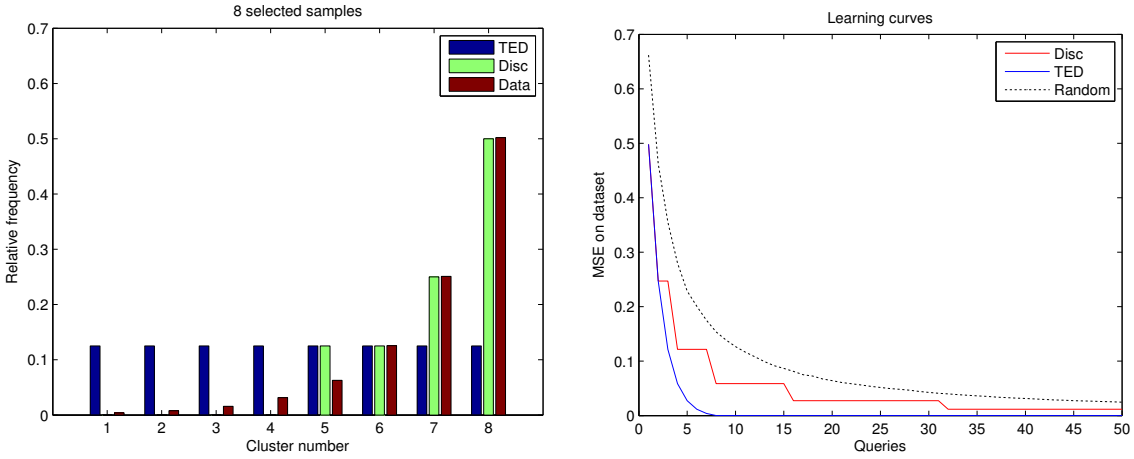
To understand the behavior of TED, we interpret the TED objective using Equation 2.10 given in Subsection 2.3.1. For small values of λ, the approximation error will play the largest role in the TED objective, determining which object will be labeled. Initially when starting with an empty labeled set none of the objects in $\hat{P}$ can be reconstructed. The reconstruction error for objects in cluster eight will be the largest (since these objects are in the majority), so first one of these objects will be sampled. After this query, the objects in cluster eight can be reconstructed perfectly and thus the TED objective is decreased the most. After this, TED will request the label of an object from cluster seven (since these objects are now in the majority and since these objects cannot be approximated), and so on.

In Figure 4.6a and Figure 4.6b we observe that the discrepancy (and thus also the MMD) perform suboptimal, we now explain this behavior. Recall that the discrepancy active learner will try to make the covariance matrices $\frac{1}{n}X^TX$ of the sets $\hat{P}$ and $\hat{Q}$ similar, see Equation 2.6 in Subsection 2.2.1. Recall that the MMD active learner also aims to accomplish this, see Equation 2.8 in Subsection 2.2.3. To make the covariance matrices similar, the discrepancy and MMD active learner will try to select queries so that the ratios of the clusters in $\hat{Q}$ are equal to the ratios of $\hat{P}$ as displayed in Table 4.3. This can be observed in Figure 4.6a and Figure 4.6c by noting that the histogram of the discrepancy resembles the histogram of the dataset. Therefore the discrepancy and the MMD will sample clusters with many objects multiple times before sampling all clusters. In the realizable setting, these samples are redundant since the label is deterministic (as also assumed by the discrepancy and MMD bound). Therefore, the discrepancy and the MMD active learner perform suboptimal on this artificial dataset.

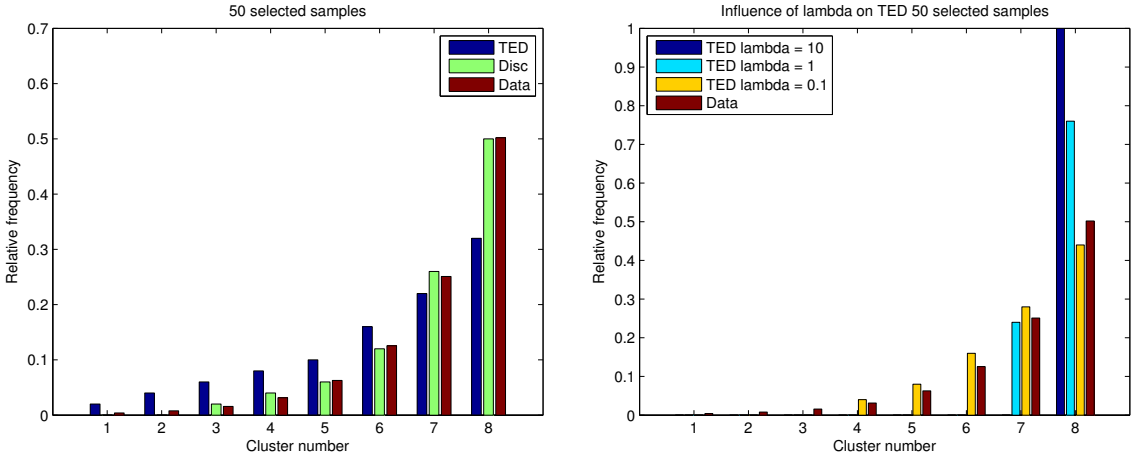
TED thus samples the feature space more aggressively and therefore is likely to perform better in the realizable setting. It is noteworthy that the discrepancy and MMD active learner query redundant samples given that both corresponding generalization bounds assume the labels are deterministic. This experiment suggests that approximating the sampling ratios of $\hat{P}$ may not be important in the realizable setting.

Now we briefly comment on the behavior of TED after eight queries. After selecting eight examples TED will focus on minimizing the regularization term in Equation 2.10, since the approximation term will be small since all objects can be reconstructed perfectly. Because cluster eight is in the majority it has the biggest influence on the regularization term. To reduce the regularization term TED samples another object from cluster eight, since then the objects in cluster eight can be reconstructed using two objects from $\hat{Q}$ and thus the regularization term will be smaller. In the end this will cause TED to also select objects somewhat proportionally to the ratios of the whole dataset just like the discrepancy, as can be seen in Figure 4.6c.

For larger values of λ the ratios in the selected sample $\hat{Q}$ of TED change. The larger the regularization parameter, the more important dense clusters become. This is because the regularization term in the TED objective will play a larger role. This is illustrated in Figure 4.6d. So for larger λ TED queries more redundant samples, and in these cases the discrepancy active or MMD active learner can perform better in the realizable setting. However λ is typically quite small in practice, and therefore it is more likely that the discrepancy active learner will choose more redundant samples than the TED active learner.



(a) (b)



(c) (d)

Figure 4.6: Results on the ratio dataset.

4.3.2. Real World Data

In the artificial example in previous subsection we saw that TED outperforms the discrepancy and the MMD because it samples the feature space more aggressively. In this subsection we benchmark their performance on real world data in the realizable setting to see if this holds for real world data as well. We also compare the MMD and discrepancy and answer our main research question: can the discrepancy improve upon the MMD as suggested by our theoretical analysis? Furthermore, since in this setting the approximation errors vanish and all bounds are guaranteed to hold, we look at the explicit values of the generalization bounds to see which bounds are tighter. Finally we investigate if the tightness of the bounds can tell us something about which active learner performs better.

Since we found that taking the hypothesis set and loss into account is generally favorable, we only compare the MMD HS active learner with the other methods and do not include the other MMD active learners (discussed in Section 4.2) in this comparison.

Some illustrating learning curves are shown in Figure 4.7 on page 52, all learning curves can be found in Figure K.3 in the appendix. We summarize the results using a paired two tailed t-test with $p = 0.05$ in Table 4.4.

Dataset	TED vs Disc	TED vs MMD HS	Disc vs MMD HS
vehicles	9/1/0	9/1/0	0/10/0
heart	5/5/0	4/5/1	0/3/7
sonar	9/1/0	3/7/0	0/1/9
iris	8/0/0	8/0/0	0/8/0
thyroid	10/0/0	10/0/0	0/10/0
ringnorm	10/0/0	10/0/0	0/0/10
ionosphere	10/0/0	10/0/0	0/0/10
diabetes	9/1/0	8/2/0	0/9/1
twonorm	10/0/0	3/4/3	0/1/9
banana	9/1/0	9/0/1	0/3/7
german	10/0/0	10/0/0	0/1/9
splice	9/1/0	9/1/0	1/9/0
breast	10/0/0	10/0/0	1/0/9

Table 4.4: Win / tie / loss counts comparing MMD HS, discrepancy and TED. We see TED performs the best in the majority of all experiments and the discrepancy performs the worst.

Observe that for some datasets the performance with little labeled samples seems very unstable, see for example the active learning curves on the datasets `german` and `ionosphere`. It is unclear what causes this behavior which seems to be consistent, since these results remain after averaging.

We observe that TED outperforms the discrepancy and MMD HS active learner consistently for all datasets: if we look at the learning curves we observe that the TED learning curve is almost without exception lower than the other learning curves. From the table we observe that TED outperforms the discrepancy significantly for all datasets except the dataset `heart`, and TED outperforms the MMD HS significantly for all datasets except `heart` and `twonorm`. TED is almost never outperformed significantly by other methods. We can explain this behavior using the artificial experiment we performed in the previous subsection: TED chooses samples more aggressively, and the discrepancy and the MMD HS query more redundant samples.

Surprisingly we observe that the discrepancy performs significantly worse for 8 out of the 13 datasets compared to the MMD HS active learner. For other datasets it matches the performance of MMD HS, but it almost never obtains better performance. The most notable example is the performance of the discrepancy on the `ringnorm` dataset where it even performs significantly worse than random sampling for queries 10 to 25, while other methods perform significantly better than random sampling. In Section 2.2.3 we showed that the discrepancy always results in a tighter generalization bound. Surprisingly, this does not imply better active learning performance. We revisit these surprising results later in this subsection.

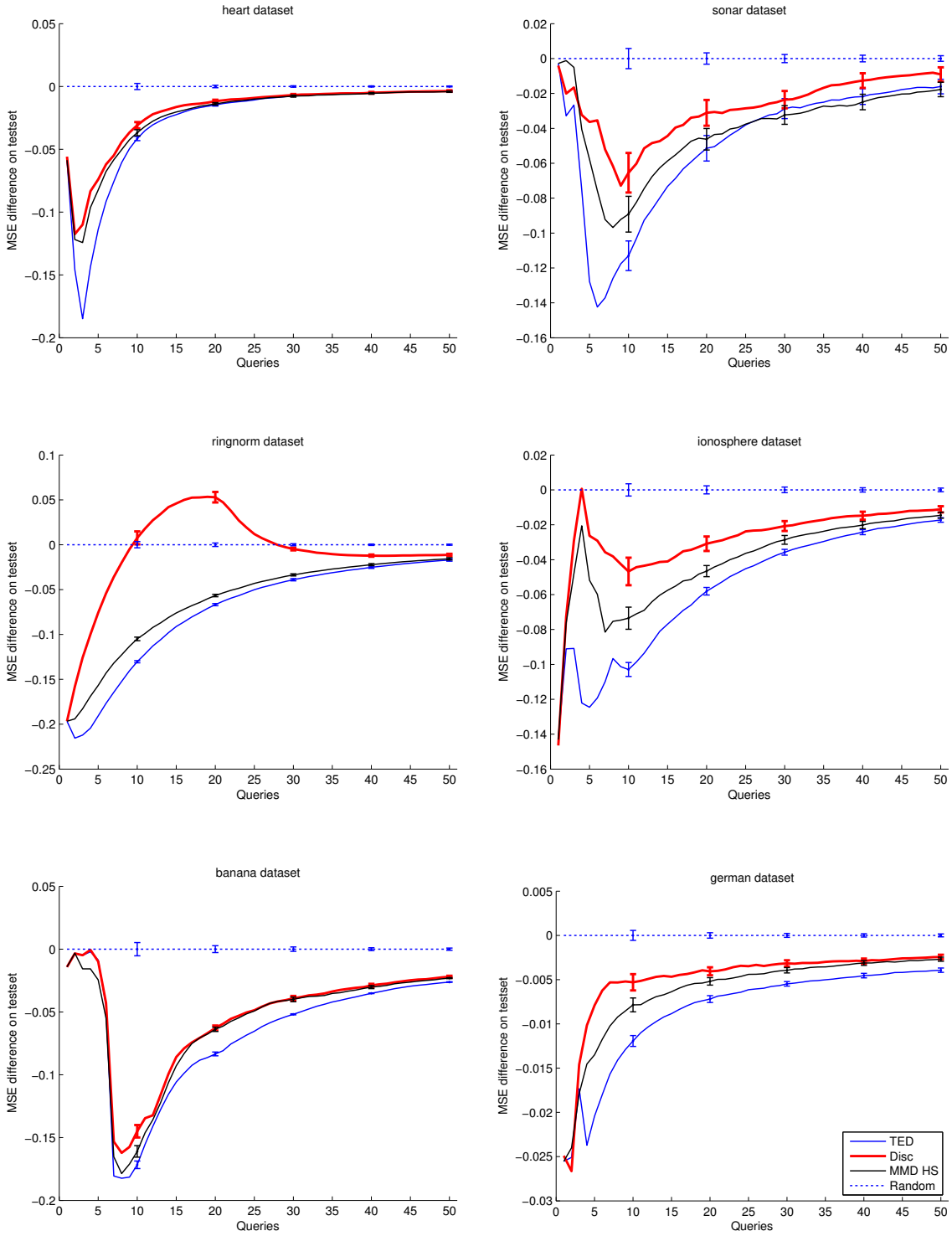


Figure 4.7: Results on six benchmark datasets for the realizable case where $f \in H$. TED often performs the best while the discrepancy often performs the worst. See Figure K.3 in the appendix for all learning curves.

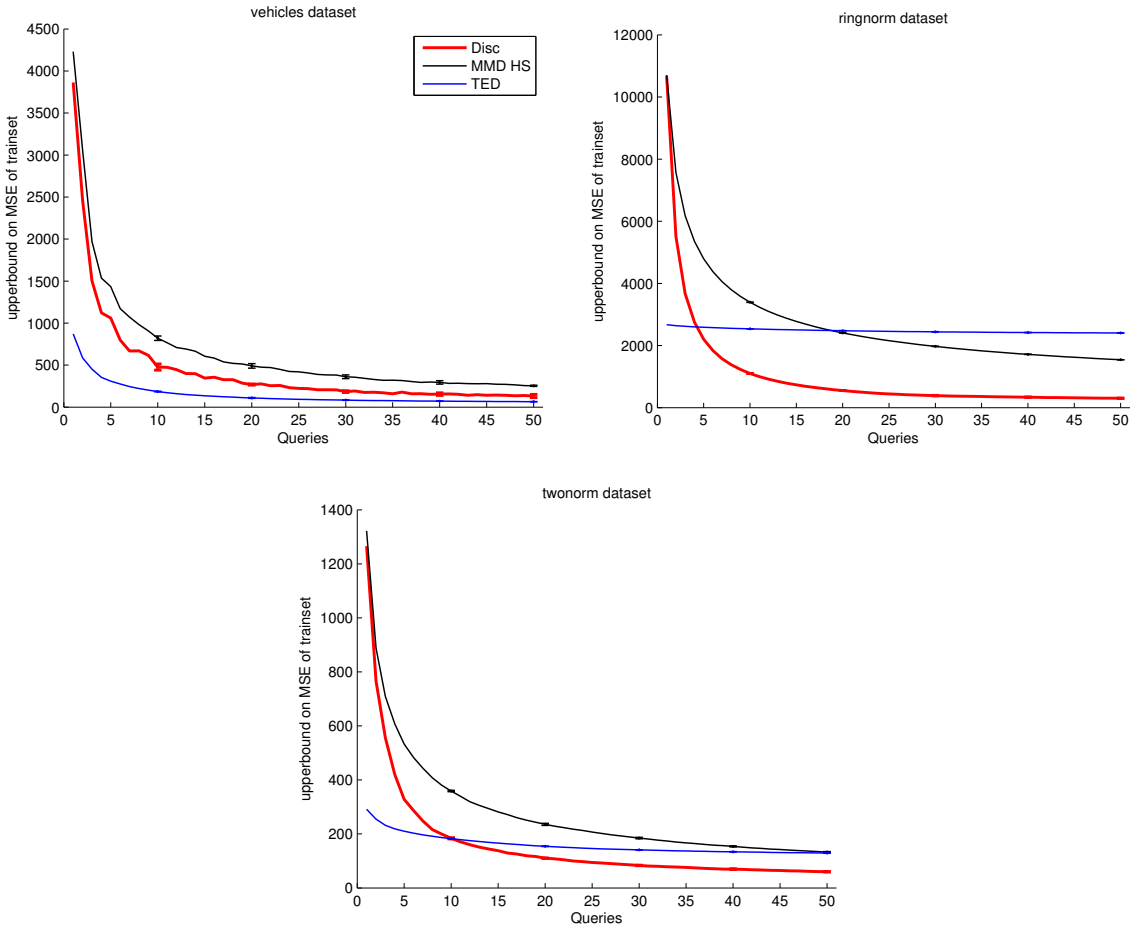


Figure 4.8: The values of the generalization bounds on the performance of the trained model during active learning on the set $\hat{P}$ for the realizable setting where $f \in H$ for three illustrating datasets. In this setting the bounds are guaranteed to hold. See Figure K.4 in the appendix for the results on all datasets.

How about the bounds for TED, MMD HS and the discrepancy? Which bounds are tighter and is this informative? The average value of the bounds for all active learners during some of the active learning experiments are shown in Figure 4.8, for all results see Figure K.4 in the appendix. Observe that the discrepancy bound is tighter than the MMD HS bound as expected. Surprisingly the TED bound is not always tighter than the discrepancy or MMD HS bound. We expected that the TED bound would be tighter, since it is less general bound: it only holds for the trained model, and takes the loss, hypothesis set and regularization parameter into account. TED also samples more aggressively but this is also not reflected in the tightness of the bound. Which bound is tighter seems to be dataset dependent and dependent on the amount of queries. The general trend is that the TED bound is tighter for a small amount of labeled examples, while the discrepancy bound seems to be tighter for a large number of labeled examples. This behavior remains difficult to explain, since the TED and MMD HS bounds are difficult to compare since they depend on different quantities, see also our discussion in Subsection 2.3.3.

This once more confirms that tightness of the bound does not correlate well to performance in active learning. While for many datasets the TED bound is not the tightest, the TED active learner performs best for most datasets.

Observe that the bound on the mean squared error are quite loose. For all datasets the bounds are much larger than 1, while the mean squared error is typically smaller than 1 on the testset for all datasets for all number of queries. The bounds are loose because the value of λ is typically small, and therefore the bounds become quite weak because of their dependence on $\Lambda^2 = \frac{1}{\lambda}$. This

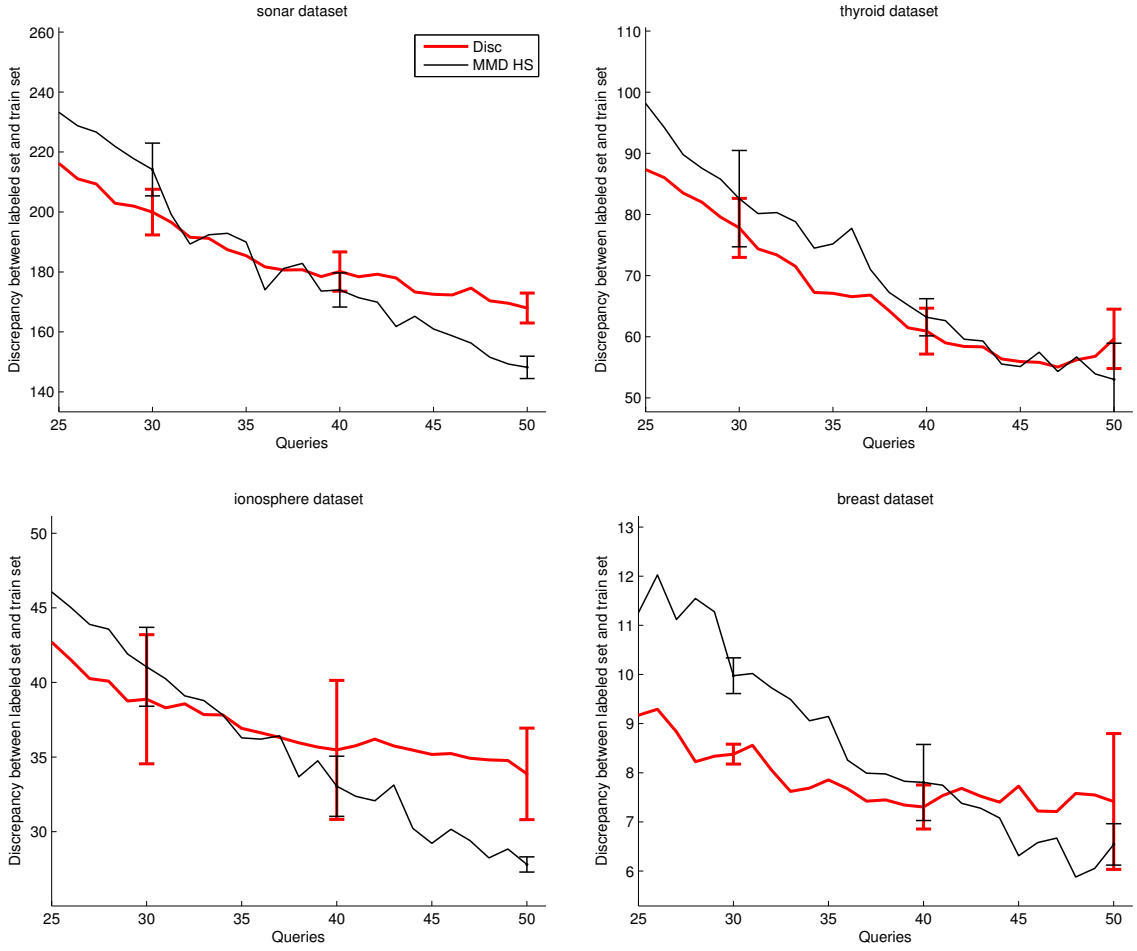


Figure 4.9: The average value of the discrepancy between the sets $\hat{P}$ and $\hat{Q}$ during active learning for the MMD HS and discrepancy active learner. We see that on average it's possible for the MMD HS active learner to obtain a lower discrepancy on these datasets, while MMD HS does not aim to minimize the discrepancy.

may explain why a tighter bound does not guarantee better performance: the bounds are too loose for their relative 'tightness' to matter.

Finally, to gain more insight in our surprising results concerning the MMD HS and the discrepancy active learners, we can compare the value of the discrepancy between the sets $\hat{P}$ and $\hat{Q}$ for both active learners. We would expect that the discrepancy active learner will obtain a minimal discrepancy, since this is explicitly minimized by this active learner. On the datasets `sonar`, `thyroid`, `ionosphere` and `breast` we observe however that the MMD HS active learner actually can attain a lower discrepancy value, see Figure 4.9. In the next section we investigate the differences between the MMD HS and discrepancy active learner in more detail and explain this behavior. Most importantly: we explain why the discrepancy performs worse than MMD HS while the discrepancy generalization bound is tighter.

4.4. Why Does the Discrepancy Perform Worse than the MMD HS Active Learner?

In the last section we found surprising results: the discrepancy performs worse compared to MMD HS in general on real world data in the realizable setting, while our theoretical results indicate the discrepancy generalization bound can estimate the generalization error more accurately since it is a tighter bound. Furthermore we found that the MMD HS can obtain a lower discrepancy

value during active learning, while the discrepancy active learner actually aims to minimize this and MMD HS does not.

In this section we show using two artificial examples that the discrepancy in certain cases can choose non-informative examples. The first artificial example is shown in the linear kernel for simplicity, afterward we show that the same behavior is observed in the Gaussian kernel. In the third subsection we show that this can happen because the discrepancy only considers a worst-case scenario which is extremely unlikely in practice. We show that the MMD HS minimizes a quantity that is related to the generalization error in the worst case but also in the average case. Using these insights we explain why the discrepancy performs worse than MMD HS in the previous subsection.

4.4.1. Artificial Example in the Linear Kernel

First we explain the artificial dataset and the setting. Afterward, we calculate by hand which samples are chosen by the discrepancy and MMD HS active learner, and illustrate by this calculation why the discrepancy can choose suboptimal examples for labeling.

For the linear kernel we choose a four dimensional dataset which contains four clusters. The dataset is illustrated in Table 4.5 on page 56 and is similar to the previous artificial dataset in Section 4.3.1. All objects within a cluster share the same label and the same position. The covariance matrix of this dataset is diagonal, and therefore the discrepancy and MMD HS active learners are easy to analyze, since the matrix M will be diagonal. We assume the regularization parameter is small and that we are in the realizable case, so the assumption of both bounds are satisfied and the approximation errors vanish.

To investigate the differences between MMD HS and the discrepancy, we exploit their difference. The discrepancy only evaluates the largest absolute eigenvalue, while MMD HS takes all eigenvalues into account. We create an artificial situation where the largest absolute eigenvalue is not informative for choosing samples, by starting the active learning process with an initially biased sample. We take an initial biased sample that contains 38 objects from cluster one, and one object from cluster two. Now we examine which sample is selected next by the discrepancy and MMD HS. To perform optimally in this scenario, the active learner should either query an object from cluster three or four, and afterward should query the remaining cluster. Since then we obtain four linearly independent samples, and f can (almost, ignoring regularization) be uniquely determined.

To compute which object is selected, we compute $M = C_{\hat{P}} - C_{\hat{Q}}$, where $C_{\hat{P}}$ and $C_{\hat{Q}}$ are the covariance matrices of $\hat{P}$ and $\hat{Q}$, respectively. The computation is illustrated in Table 4.6 on page 56. Because the matrices are diagonal we only write down the diagonal. We directly compute the absolute values of the diagonal matrix M, since only the absolute values are needed to compute the MMD HS objective and the discrepancy. Recall that the discrepancy is given by the largest absolute eigenvalue of M, or in this case the largest absolute value on the diagonal. The MMD HS objective is given by the Frobenius norm of the matrix M, since M is diagonal this is equal to the euclidean norm of the vector representing the diagonal.

As can be seen, the discrepancy cannot differentiate between querying cluster two, three and four. The MMD HS active learner however can, and will query cluster three or four. Thus the discrepancy active learner might sample cluster two again, while this will not give the learning algorithm new information about the remaining unlabeled clusters. The same happens when multiple queries are selected afterward in this scenario: the discrepancy will rank all clusters except cluster one equally. In this example, MMD HS therefore will perform better, since on average, it will query all three clusters faster than the discrepancy.

This happens because the discrepancy only looks at the largest eigenvalue of M. The largest eigenvalue in this case characterizes the oversampling of cluster one. The discrepancy will only want to decrease this oversampling and ignores the other eigenvalues. MMD HS does take the other eigenvalues into account as well, and therefore can distinguish between sampling from clusters two, three and four. Thus the discrepancy bound is indeed tighter as can be seen in this example, because the discrepancy is lower than the MMD HS value in all cases, but actually the MMD HS bound is more informative for choosing the next query in this example.

Cluster number	Cluster center	Amount of objects
1	$[1, 0, 0, 0]$	250
2	$[0, 1, 0, 0]$	250
3	$[0, 0, 1, 0]$	250
4	$[0, 0, 0, 1]$	250

Table 4.5: Sample bias correction dataset

Query from	cluster 1	cluster 2	cluster 3	cluster 4
Diagonal of $C_{\hat{P}}$	$\frac{1}{40}[10, 10, 10, 10]$	$\frac{1}{40}[10, 10, 10, 10]$	$\frac{1}{40}[10, 10, 10, 10]$	$\frac{1}{40}[10, 10, 10, 10]$
Diagonal of $C_{\hat{Q}}$	$\frac{1}{40}[39, 1, 0, 0]$	$\frac{1}{40}[38, 2, 0, 0]$	$\frac{1}{40}[38, 1, 1, 0]$	$\frac{1}{40}[38, 1, 0, 1]$
Absolute diagonal of M	$\frac{1}{40}[29, 9, 10, 10]$	$\frac{1}{40}[28, 8, 10, 10]$	$\frac{1}{40}[28, 9, 9, 10]$	$\frac{1}{40}[28, 9, 10, 9]$
Discrepancy objective	$\frac{29}{40}$	$\frac{28}{40}$	$\frac{28}{40}$	$\frac{28}{40}$
MMD HS objective	0.8374	0.8093	0.8085	0.8085

Table 4.6: Computation of the discrepancy and MMD for the next sample to query for the sample bias correction dataset where a biased sample is used at the beginning of the active learning experiment: 38 samples from cluster one, and one object from cluster two.

This indicates a fundamental problem with the discrepancy in sequential active learning. Because it only focuses on minimizing the largest absolute eigenvalue in the current situation, other eigenvalues may actually increase. In later iterations these eigenvalues that became larger may start dominating the bound. Thus the discrepancy is in some sense greedy: in later iterations other eigenvalues may dominate the bound, thus it might be wiser to minimize all eigenvalues like MMD HS. Therefore the behavior of MMD HS can be better in the sequential setting: this explains why in the previous subsection we saw that MMD HS can obtain a lower discrepancy value: because it is less greedy it possibly can minimize the discrepancy more effectively. Note that this analysis holds in general (also for non-biased initial samples) for sequential active learning using MMD HS and the discrepancy. However, this greedy behavior of the discrepancy does not completely explain all our surprising results where MMD HS outperforms the discrepancy, since in that case we would expect MMD HS to obtain lower discrepancy values more often.

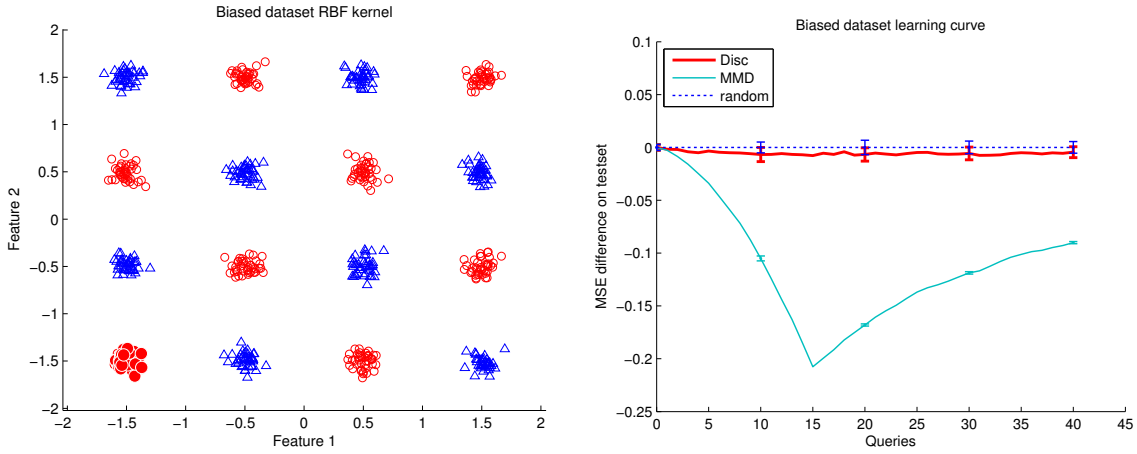
However, observe that the discrepancy only focuses on minimizing the largest absolute eigenvalue. This is because the discrepancy assumes a worst-case scenario. MMD HS however performs better in this artificial example since it minimizes all eigenvalues. This suggests the worst-case analysis of the discrepancy, that is based on the largest absolute eigenvalue, may not always be informative. We investigate this in depth in Subsection 4.4.3.

4.4.2. Artificial Example in the Gaussian Kernel

First, we show in this subsection that the behavior observed in the previous subsection also applies to the Gaussian kernel. We briefly explain the dataset and setting, afterward we show the learning curves that indicate MMD HS outperforms the discrepancy.

We use the dataset displayed in Figure 4.10a. We use a Gaussian kernel with $\sigma = 0.1$ so the labels of one cluster hardly influence the labels in other clusters similar to the linear dataset in the previous subsection. We begin each active learning experiment with an initial labeled set given by 30 objects in the bottom left cluster. The resulting learning curves are shown in Figure 4.10b.

We observe that the MMD HS active learner clearly and significantly outperforms the dis-



(a) (b)

Figure 4.10: a) Biased artificial dataset for the RBF kernel. Circles and squares indicate positive and negative class, respectively. The initially labeled sample is indicated with solid circles and is located in the bottom left corner. b) Learning curves. MMD HS obtains a significantly lower mean squared error than the discrepancy. The point of zero queries corresponds to the initially labeled sample.

crepancy active learner. The discrepancy learner more or less samples randomly from all clusters except the bottom left cluster as expected. As in the linear artificial dataset it does not choose diverse examples. This is because the discrepancy only focuses on minimizing the absolute largest eigenvalue, which indicates oversampling of the bottom left cluster. Thus the discrepancy only aims to minimize this oversampling, and therefore chooses samples randomly from the other clusters.

This illustrates that the conclusions from the previous subsection also generalize to experiments performed using the Gaussian kernel.

4.4.3. The Worst-Case Analysis of the Discrepancy Is Too Unlikely

In the previous subsections we have shown that the discrepancy can choose non-diverse samples in case of an initial biased sample while MMD HS does not suffer from this. This suggests the discrepancy is somehow considering a less informative worst-case scenario than MMD HS. In this subsection we examine the worst-case scenario considered by the discrepancy, and we argue that this worst-case scenario is very unlikely to occur in the agnostic setting and even impossible in the realizable case. We verify this empirically using our experiments in the realizable scenario on the benchmark datasets. We show that the MMD HS minimizes a quantity that is more relevant in practice. Finally, we use these insights to explain the surprising results observed on the benchmark datasets in the realizable setting.

We have already discussed that the worst-case scenario of the discrepancy might be unlikely in Section 2.4. In this section we make this more concrete and we verify this empirically. We also show in this section that the scenarios considered by MMD HS are more likely to occur than the worst scenario of the discrepancy, to explain why the discrepancy does not improve upon the MMD HS active learner.

To this end, we perform a brief theoretical analysis in terms of the matrix M. We perform this analysis in the realizable setting and consider a linear kernel for simplicity. In that case we indicate our learned model by w, and the model generating the labels by w'. Following the derivation of the discrepancy in Section B.1, observe it is straightforward to show that for the squared loss the following holds for any w and $w' \in H$:

$$L_{\hat{P}}(w, w') = L_{\hat{Q}}(w, w') + u^T M u = L_{\hat{Q}}(w, w') + \sum_i \bar{u}_i^2 \lambda_i \tag{4.1}$$

Observe that this equality strictly holds in the realizable setting and is not an approximation. Here $u = w' - w$ and $\bar{u}_i$ is the projection of u on the (normalized) eigenvector v_i of the matrix M

corresponding to the eigenvalue λ_i. Essentially, $\bar{u}_i$ are the components of u in the basis formed by the orthonormal eigenvectors of M. We indicate the vector u with respect to this basis as $\bar{u}$. Note that $\bar{u}$ is equal to u up to a rotation.

As mentioned earlier, equation 4.1 holds for any w and $w' \in H$. However, we are especially interested in the case where w is obtained by training on the set $\hat{Q}$ with labels generated by w', since this is the model we use in practice. Therefore, from here on, w will refer to the model trained on $\hat{Q}$ with labels generated by w'. This will also be reflected in the definition of $u = w' - w$ and $\bar{u} = \bar{w}' - \bar{w}$. Furthermore, we indicate the vectors w and w' with respect to the basis of the orthonormal eigenvectors of M by the vector $\bar{w}$ and $\bar{w}'$.

In the realizable case the loss on the set $\hat{Q}$ will be relatively small after training (assuming a small regularization parameter is used), thus we can approximate Equation 4.1 by:

$$L_{\hat{P}}(w, w') \approx \sum_i \bar{u}_i^2 \lambda_i \tag{4.2}$$

From here on out, we assume the eigenvalues of M are sorted in descending order by absolute value, thus:

$$|\lambda_1| \geq |\lambda_2| \geq |\lambda_3| \geq \dots$$

Note that λ_1 refers to the absolute largest eigenvalue. Recall that the discrepancy is given by:

$$\text{disc}(\hat{P}, \hat{Q}) = \max_{||u|| \leq 2\Lambda} |u^T M u| = \max_{||\bar{u}|| \leq 2\Lambda} |\sum_i \bar{u}_i^2 \lambda_i| = 4\Lambda^2 |\lambda_1|$$

Observe that the discrepancy maximizes over all vectors $\bar{u}$ with norm smaller than $||\bar{u}|| \leq 2\Lambda$. Essentially, the discrepancy maximizes over all w and $w' \in H$. The advantage of this approach is that the discrepancy quantity is independent of the used training procedure and the labels of the set $\hat{Q}$.

The discrepancy thus considers that the vector $\bar{u}$ points in the worst direction possible: in the direction of the eigenvector that has the largest absolute eigenvalue $|\lambda_1|$. Thus the discrepancy assumes $\bar{u} \propto (1, 0, 0, \dots, 0)^T$. In this case, the error on $\hat{P}$ is completely determined by the largest absolute eigenvalue, and the other eigenvalues do not influence the error on $\hat{P}$ at all[3].

However, if we assume we are in the realizable case, than it is extremely unlikely that w' does not point in a similar direction as w. in other words it is likely that $w \approx w'$, which we will make more concrete here using an artificial example. For this example we will use the orthonormal eigenvectors of M as basis. If we have a labeled sample given by $\bar{x} = (1, 0, 0, \dots, 0)^T$, we can determine $\bar{w}_1$ perfectly (ignoring regularization). Thus in this case we can say that $\bar{w}_1 \approx \bar{w}'_1$ or in other words that $\bar{u}_1 \approx 0$ after this sample has been labeled. The discrepancy however, even in this case where $\bar{w}_1$ is (almost, ignoring regularization) known perfectly, still *only* considers the worst case[4] for $\bar{u}_1$. Since $\bar{u}_1 \approx 0$ in this example, it would be more meaningful to instead consider the worst case for other components of $\bar{u}$.

Furthermore, the discrepancy assumes that $||u||$ is maximal, in other words that $w = -w'$. This is impossible in the realizable case if $w' \neq 0$ and extremely unlikely in the agnostic case.

To verify that this worst-case scenario of the discrepancy is very rare in practice, and that the worst case considered by the discrepancy contributes little to the error on $\hat{P}$, we compute the component $\bar{u}_1$. We can do this computation in the realizable setting since here we have access to w and w' during the active learning experiments. The computation of $\bar{u}_1$ is not straightforward in case kernels are used, see Appendix J for details. We show that the component $\bar{u}_1$ is relatively

[3]Note that if $\lambda_1 < 0$, λ_1 actually decreases the error on $\hat{P}$. We will revisit this point later in this subsection. Why is this then a worst-case scenario? Or in other words, why is there an absolute value in the definition of the discrepancy? This is because the discrepancy considers the worst-case scenario where the losses on $\hat{Q}$ and $\hat{P}$ differ the most, see Subsection 2.2.2.

[4]Note that by sampling $\bar{x}$ the matrix M will change, however even if $\bar{x}$ has been sampled the discrepancy may still consider this worst case for $\bar{u}_1$ if λ_1 remains the absolute largest eigenvalue, see also Subsection 4.3.1 or Subsection 4.4.1.

small by comparing it with the other components of $\bar{u}$. To this end we compute the fraction R defined as:

$$R = \frac{|\bar{u}_1|}{\sum_i |\bar{u}_i|}$$

If $R = 1$ we are in the worst-case scenario of the discrepancy. If R is closer to zero the worst case of the discrepancy plays a smaller role in causing the mean squared error on $\hat{P}$. In Figure 4.11 we show a summary of all values of R observed during all experiments on the benchmark datasets of Section 4.3.2. Observe that the maximum value of R observed is 0.25. Only a small fraction of 0.02 of all observed values of R is larger than 0.05. This indicates that the worst-case scenario of the discrepancy indeed is very rare and contributes little to the error on $\hat{P}$. Only if $R = 1$ we truly are in this worst-case scenario which we have never observed.

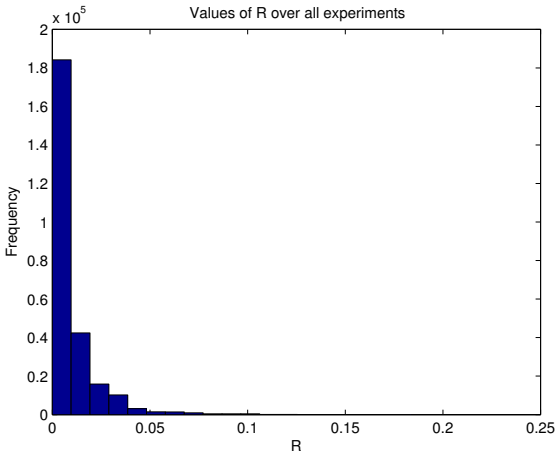


Figure 4.11: Histogram of all observed values of R during all active learning experiments.

Furthermore, observe that the influence of $\bar{u}_1$ and λ_1 on the mean squared on $\hat{P}$ can be measured directly through the quantity:

$$\bar{u}_1^2 \lambda_1$$

See also equation 4.2. To characterize the influence of $\bar{u}_1$ and λ_1 on the mean squared error on $\hat{P}$ we plot $\bar{u}_1^2\lambda_1$ and $u^T M u$ for the discrepancy active learner during some of the active learning experiments in Figure 4.12 on page 60. The results for other active learners and datasets is similar. Observe that the contribution of $\bar{u}_1^2\lambda_1$ is often small and sometimes even negative (in that case, $\bar{u}_1^2\lambda_1$ actually decreases the mean squared error on $\hat{P}$).

Thus we have seen that only little of the mean squared error on $\hat{P}$ is caused by the absolute largest eigenvalue in practice. This indicates that other eigenvalues are more to blame for the error on $\hat{P}$. This suggests other eigenvalues should be minimized as well. This is exactly what MMD HS does, since it minimizes the following objective:

$$4\Lambda^2 \sqrt{\sum_i \lambda_i^2}$$

Thus MMD HS focuses on minimizing all eigenvalues, and therefore likely performs better, since the case considered by MMD HS where all components of $\bar{u}$ play a role in causing the error on $\hat{P}$ is much more likely than the worst-case analysis of the discrepancy. Thus we can say that the MMD HS active learner minimizes a quantity that is more relevant in practice, since this quantity also accounts for non-worst-case scenarios. Observe that MMD HS does prefer minimizing large eigenvalues, since it minimizes the squared sum of the eigenvalues. This suggests that MMD HS weighs worst-case scenarios more than average-case scenarios, which may be a questionable assumption. We address this with our proposed nuclear discrepancy active learner introduced in Section 2.4

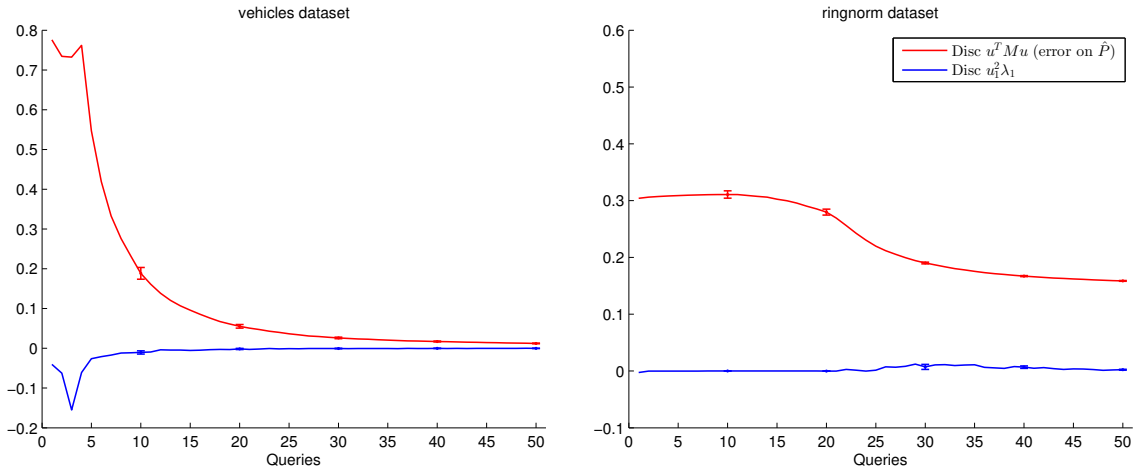


Figure 4.12: We plot the contribution of $\bar{u}_1^2 \lambda_1$, the contribution of $\bar{u}_1$ and λ_1 to the error on $\hat{P}$, along with the complete error on $\hat{P}$ that is given by $u^T M u$.

which we evaluate in the next subsection. However, first we address why the discrepancy learner sometimes does perform well, and in what cases the MMD HS active learner might especially improve upon the discrepancy.

The discrepancy *can* perform well, since minimizing the largest absolute eigenvalue tends to decrease the size of other eigenvalues as well. However in examples where the discrepancy places all of its efforts on minimizing the absolute largest eigenvalue and this has little effect on the absolute size of other eigenvalues, the performance of the discrepancy may suffer. This is because the error on $\hat{P}$ is caused by these other eigenvalues as well which are in that case hardly minimized. This behavior is illustrated by the observed eigenvalues during the active learning experiment of the discrepancy on the dataset `ringnorm` in Figure 4.13. In this case, minimizing the absolute largest eigenvalue hardly influences the other eigenvalues until these eigenvalues become the absolute largest. In Figure 4.13 we see that MMD HS decreases all eigenvalues simultaneously, instead of focusing only on the absolute largest eigenvalue. Because the eigenvalues in this case are quite small the effect seems small, however note that for the `ringnorm` dataset the matrix M has 650 nonzero eigenvalues!

To illustrate this for all datasets more clearly, we compute the sum of absolute eigenvalues, where we do not include the largest absolute eigenvalue. We call this quantity E:

$$E = \sum_{i \neq 1} |\lambda_i|$$

We expect that the MMD HS active learner obtains lower values of E than the discrepancy, since MMD HS focuses on minimizing all eigenvalues and the discrepancy only focuses on minimizing $|\lambda_1|$. In cases where E is much larger for the discrepancy than MMD HS the performance of the discrepancy may degrade compared with the performance of MMD HS because the other eigenvalues also contribute to the squared error on $\hat{P}$.

Plots of the average value of E during some of the active learning experiments of Section 4.3.2 are shown in Figure 4.14 and Figure 4.15 on pages 62 and 63. For results on all benchmark datasets see Figure K.5 in the appendix. Observe that for the datasets `sonar`, `ringnorm`, `ionosphere`, `twonorm`, `slice` and `german` the quantity E is much larger for the discrepancy active learner compared to the value of E obtained by the MMD active learner. For these datasets except the `splice` dataset we observe a performance degradation for the discrepancy, see Table 4.4 and the active learning curves in Figure K.3 in the appendix. The value of E for the discrepancy is sometimes even larger than E for random sampling. This is especially clear for the `ringnorm` dataset, where the discrepancy performs worse than random sampling, see also the learning curves in Figure 4.7 on page 52. We may conclude that the quantity E gives a good indication of the

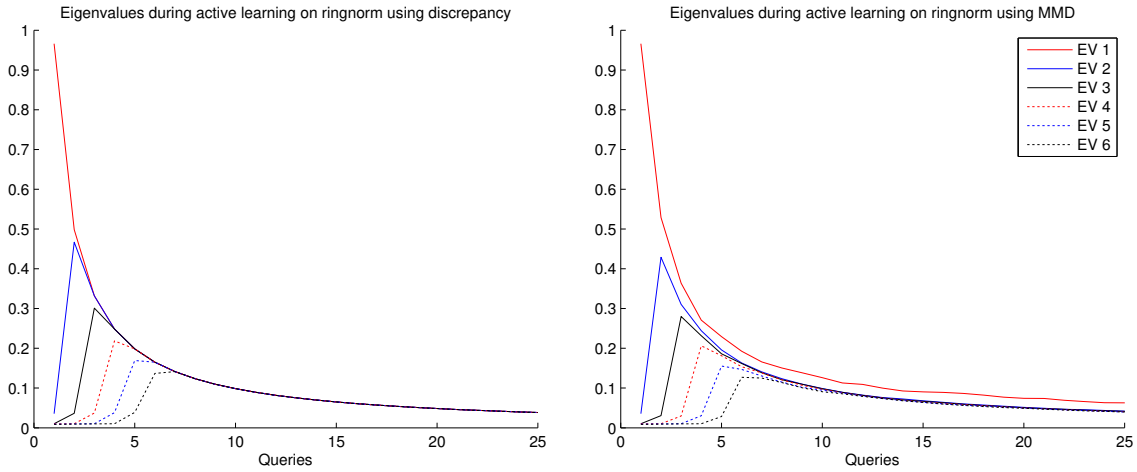


Figure 4.13: The minimization of eigenvalues by the discrepancy versus the MMD. 'EV1' indicates the absolute largest eigenvalue, 'EV2' indicates the second largest absolute eigenvalue, etc... We plot the absolute values of all eigenvalues. Observe that for the `ringnorm` dataset the discrepancy needs to minimize the absolute largest eigenvalue in order to minimize other eigenvalues, and therefore it takes longer before these other eigenvalues are minimized. The MMD HS active learner instead minimizes all eigenvalues simultaneously.

performance of the active learning methods, confirming our analysis that all eigenvalues λ_i are relevant for the performance on the set $\hat{P}$ and not only the absolute largest eigenvalue is the most important.

One may wonder why the discrepancy performs worse than MMD HS, yet both consider a worst-case scenario and use the same assumptions. What happened is the following. Since MMD HS has incomplete knowledge of the loss (since the MMD considers a worst-case where the loss can become negative), it considers a more approximate worst-case scenario. Because the discrepancy takes the loss into account, it can more accurately characterize the worst case given the assumptions. The discrepancy worst case is therefore much more specific. However, because of this specificity, it considers a more unlikely scenario.

This all also has to do with the assumptions of both methods: both assume that it is possible to obtain any model $w \in H$ and any oracle model in $w' \in H$ and consider the worst case for these models. If these assumptions in fact were true, the discrepancy would perform better than the MMD HS in a worst-case scenario (by definition). These assumptions are however too conservative, since in practice it is more likely that $w \approx w'$. MMD HS performs better than the discrepancy since it considers a less specific worst-case scenario. Because of this, MMD HS minimizes a quantity that is more relevant, since this quantity also accounts for non-worst-case scenarios as well which are more likely to occur in practice.

Interestingly, observe in Figure 4.14 and 4.15 that TED may not minimize all eigenvalues. This is likely because TED, since it has knowledge of $w - w'$ due to the analytical solution of w, is able to estimate which $\hat{u}_i$ will be large. Because of this, TED can focus on minimizing the eigenvalues of M that matter most for generalization performance on $\hat{P}$, and thus can obtain good performance without minimizing all eigenvalues.

In this section we have shown that generally all eigenvalues of M contribute towards causing the mean squared error on $\hat{P}$, and we have shown that the absolute largest eigenvalue only contributes little in practice. The discrepancy considers a specific worst-case scenario where the error on $\hat{P}$ is caused only by the largest absolute eigenvalue of M. MMD HS, which considers a more broad worst-case scenario since it has no knowledge of the specific loss used, minimizes all eigenvalues of M. Because of this MMD HS also accounts for non-worst-case scenarios. worst-case scenarios almost never occur due to the fact that the underlying assumptions of both methods are too conservative. Because of this MMD HS performs better in practice. This explains our surprising results observed in the previous section.

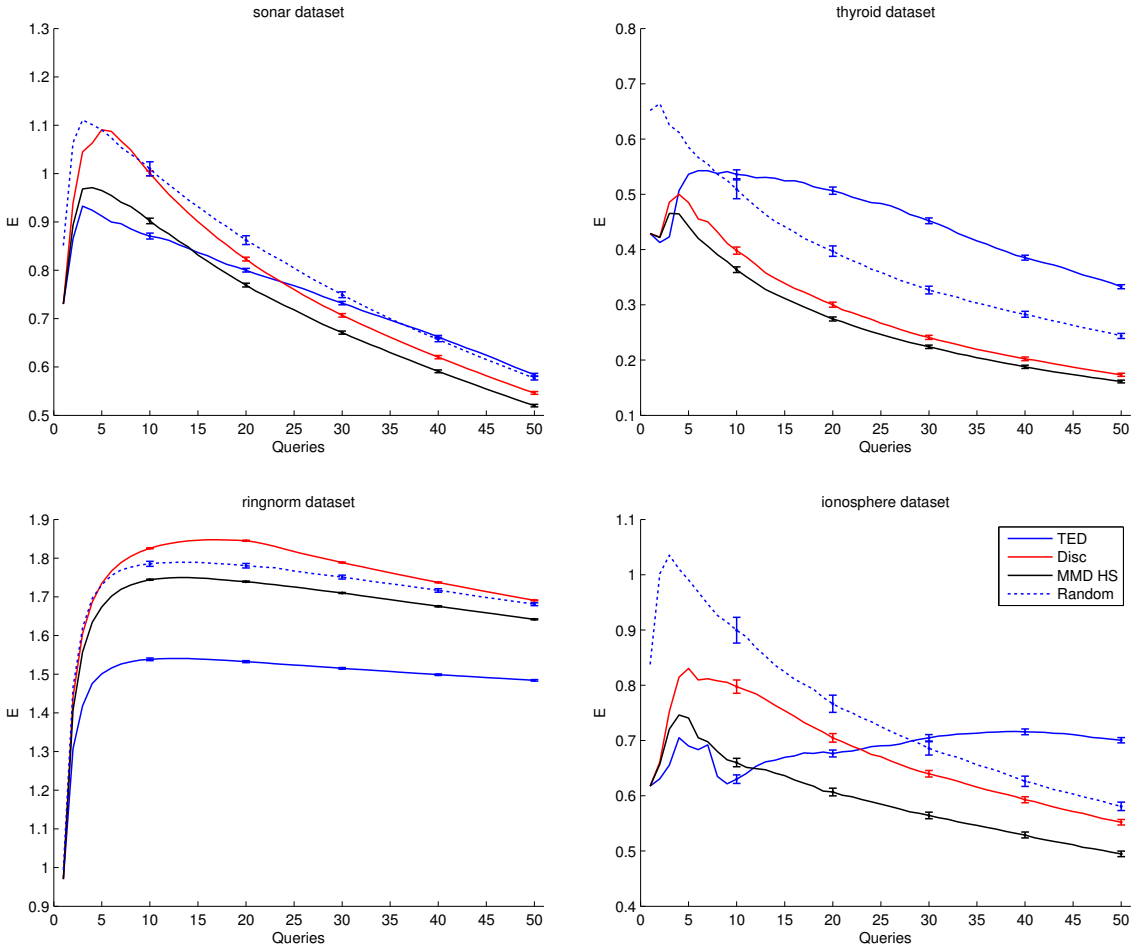


Figure 4.14: We plot the quantity E during the active learning experiments in the realizable setting for four illustrative datasets. Observe that in some cases the value of E is much larger for the discrepancy than the MMD, and sometimes E of the discrepancy becomes even larger than E of random sampling. See Figure K.5 in the appendix for the results on all datasets.

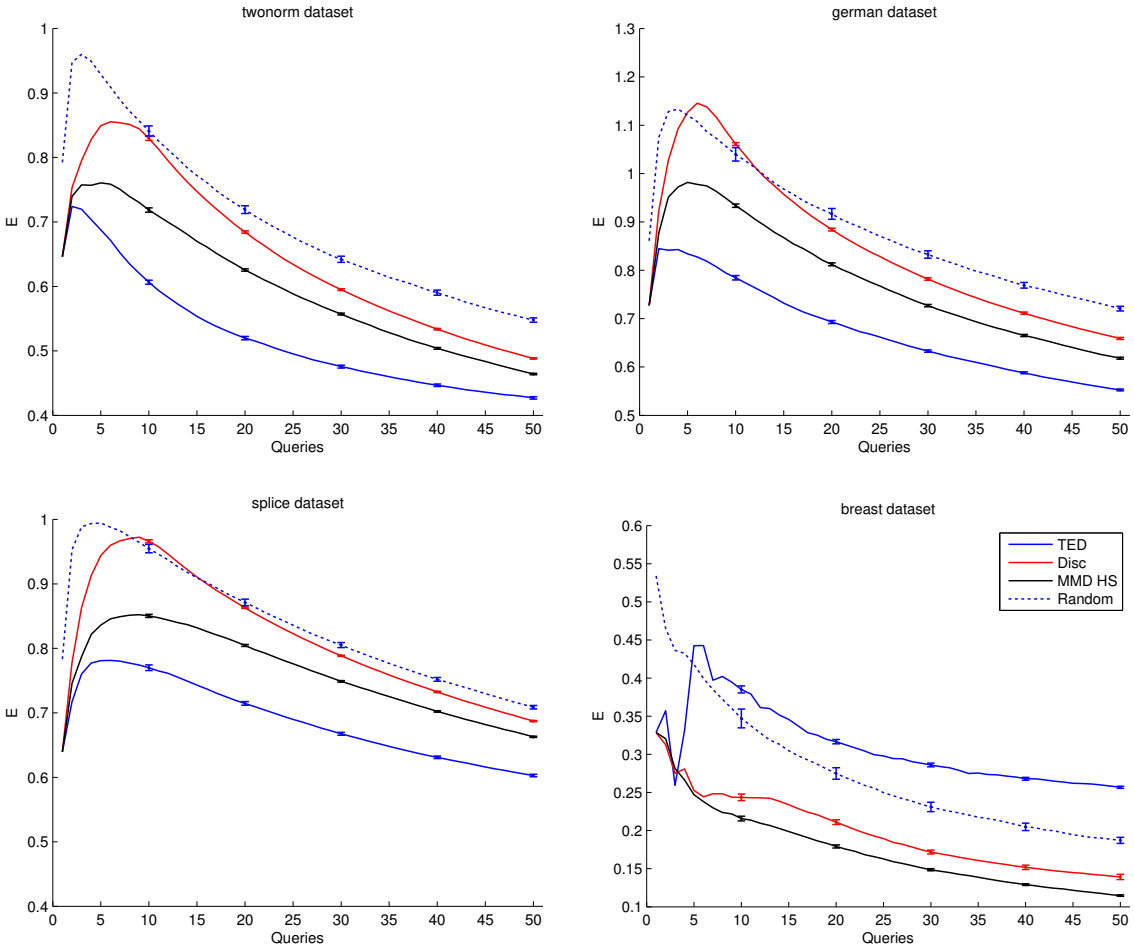


Figure 4.15: We plot the quantity E during the active learning experiments in the realizable setting for four more illustrative datasets. Observe that in some cases the value of E is much larger for the discrepancy than the MMD, and sometimes E of the discrepancy becomes even larger than E of random sampling. See Figure K.5 in the appendix for the results on all datasets.

4.5. Performance Comparison with the Nuclear Discrepancy in the Realizable Setting on Real World Data

The previous section has demonstrated that the worst-case analysis of the discrepancy is very unlikely to occur in practice. This is our motivation for the introduction of the nuclear discrepancy in Section 2.4. In this section we have argued that the nuclear discrepancy will likely perform better in practical scenarios. In view of our results in the previous subsection, where we have seen that worst-case scenarios considered by the discrepancy are unlikely, we therefore expect the nuclear discrepancy to improve upon the discrepancy active learner. In Section 2.4 we have also argued that the nuclear discrepancy weighs realistic scenarios more than the MMD HS active learner, and therefore likely performs better than the MMD HS active learner as well. In this section we validate this hypothesis by repeating the experiments on the benchmark datasets of the realizable setting with the nuclear discrepancy.

The nuclear discrepancy will be able to perform well in the artificial experiments of Subsection 4.4.1 and 4.4.2, since the nuclear discrepancy takes into account all eigenvalues during active learning, unlike the discrepancy. Furthermore, we found that the nuclear discrepancy performs similar to the discrepancy and MMD HS in the artificial experiment of Section 4.3.1. Therefore we do not repeat these experiments here. Instead, we directly evaluate the nuclear discrepancy on the benchmark datasets in the realizable setting.

The results on the benchmark datasets in the realizable setting are shown in Figure 4.16. All learning curves are given in Figure K.6 in the appendix. We summarize all results in Table 4.7 on page 66 using a two-tailed paired t-test with $p = 0.05$. Observe that the nuclear discrepancy for most datasets seems to improve upon the MMD HS and discrepancy active learners significantly. Note that especially it is almost never outperformed by both methods: it either improves upon them or matches their performance. Especially the performance improvement on the `ringnorm` dataset is spectacular, where the nuclear discrepancy performs on par with TED. Furthermore note that for some datasets the learning curves of the nuclear discrepancy seems to resemble the learning curves of TED a lot, indicating that the nuclear discrepancy minimizes a quantity that is more similar to the TED quantity. This is likely because the nuclear discrepancy accounts more often for the scenario where $w \approx w'$ like TED, as argued in the Subsection 2.4.1.

Furthermore we observe that TED does outperform the nuclear discrepancy in most cases. This is however to be expected, since TED can more accurately estimate which eigenvalues of M should be minimized, since TED can use the analytical solution of the model to estimate which eigenvalues contribute the most to the generalization error. This behavior of TED that it only minimizes certain eigenvalues of M was also observed in the previous section. Furthermore, TED has an advantage since it has knowledge about the regularization parameter λ of the model, which the nuclear discrepancy does not use for its active learning strategy.

Finally, we must conclude that tighter generalization bounds truly do not imply better active learning performance, as illustrated by the nuclear discrepancy generalization bound. The nuclear discrepancy generalization bound is looser than the MMD HS and discrepancy generalization bound in the realizable setting, yet performs better in the realizable setting where the approximation errors of all bounds vanish. What instead seems to matter is that the quantity that is being minimized for active learning needs to be relevant to the generalization error in as many scenarios as possible that are likely to occur during active learning experiments.

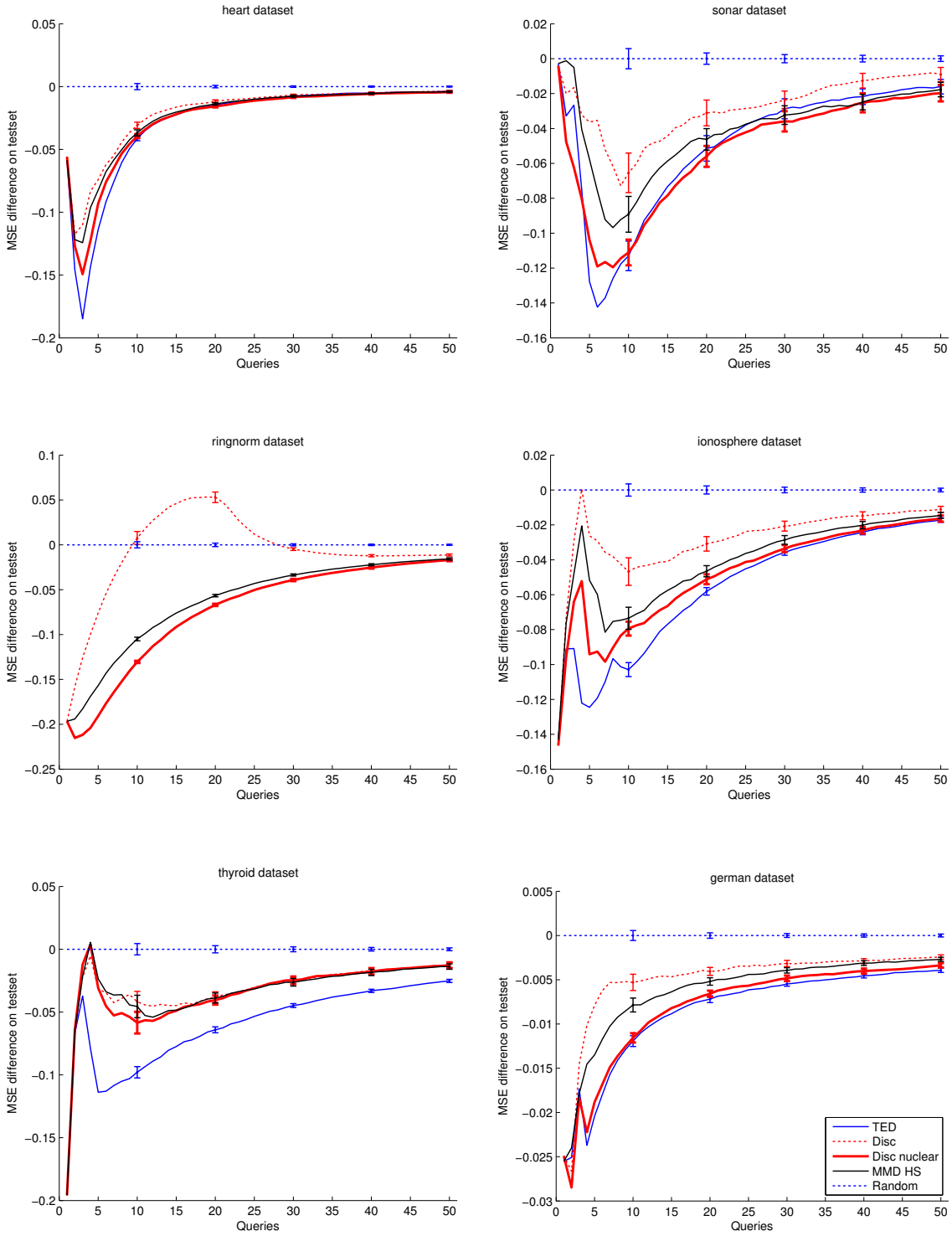


Figure 4.16: Comparison of the nuclear discrepancy active learner with the other active learning methods on some of the benchmark datasets for the realizable case where $f \in H$. Observe that the nuclear discrepancy improves a lot upon the discrepancy and sometimes upon the MMD HS active learner. In particular, on the `ringnorm` dataset the nuclear discrepancy performs much better than the discrepancy and matches the performance of TED. See Figure K.6 for the results on all datasets.

Dataset	ND vs TED	ND vs MMD HS	ND vs Disc
vehicles	0/1/9	0/10/0	0/10/0
heart	6/2/2	7/3/0	10/0/0
sonar	2/7/1	4/6/0	10/0/0
iris	0/0/8	4/4/0	0/7/1
thyroid	0/0/10	1/9/0	1/9/0
ringnorm	2/8/0	10/0/0	10/0/0
ionosphere	0/1/9	10/0/0	10/0/0
diabetes	0/1/9	5/3/2	4/6/0
twonorm	8/0/2	10/0/0	10/0/0
banana	1/0/9	0/10/0	7/3/0
german	0/1/9	10/0/0	10/0/0
splice	0/9/1	8/2/0	9/1/0
breast	0/0/10	10/0/0	10/0/0

Table 4.7: Win / tie / loss counts comparing nuclear discrepancy (ND) with TED, MMD HS and the discrepancy. Observe that the nuclear discrepancy often performs better than the MMD HS and the discrepancy, but only in exceptional cases can improve significantly upon TED, and TED outperforms the nuclear discrepancy for the majority of the datasets.

4.6. Performance Comparison in the Agnostic Setting

In the last sections we have compared the behaviors of all active learners and investigated their differences in the realizable setting. Now we investigate if the behavior observed in the realizable setting generalizes to real world data where the labeling function f might not be in the hypothesis set (the agnostic case). First we illustrate on an artificial example why TED in this case might not be optimal compared to the MMD and the (nuclear) discrepancy.

4.6.1. Artificial Dataset

We use the artificial dataset from Subsection 4.3.1 to illustrate why TED might not be optimal in case $f \notin H$. The dataset is kept the same, only we add gaussian noise to the labels[5]. If there is much noise, it will be better to query clusters multiple times in order to average out the noise, and therefore in this artificial example we use a large standard deviation of $\sigma_{\text{noise}} = 1.5$. In this case the 'redundant' sampling of the (nuclear) discrepancy and the MMD HS can thus be beneficial.

The resulting learning curves are shown in Figure 4.17. For this dataset the nuclear discrepancy, the discrepancy and MMD HS choose exactly the same samples, and therefore for clarity we only show the discrepancy in the results. In the first iteration TED explores all clusters, however, because the labels are so noisy these first labeled objects actually increase the mean squared error. The (nuclear) discrepancy and MMD HS active learner directly sample multiple objects from the most dense cluster, and therefore its performance in these first iterations is better than TED. Because this cluster contains the majority of the samples of the dataset, by choosing multiple samples of this cluster, the noise on this most important cluster is averaged out as fast as possible. In later iterations TED will also sample the most important clusters multiple times to reduce its regularization term, and thus will catch up to the discrepancy active learner in terms of the mean squared error. Note that in this case the performance of the discrepancy active learner is similar to the performance of the random active learner, and that TED performs worse than random sampling at the beginning of the active learning experiment.

In conclusion, if $f \notin H$, the (nuclear) discrepancy active learner and the MMD HS active learner can perform better than TED due to their sampling of 'redundant' samples, for example in cases where there is much label noise.

[5]For binary classification Gaussian noise is actually inappropriate, however we only use this example to illustrate what can happen in the agnostic setting in general.

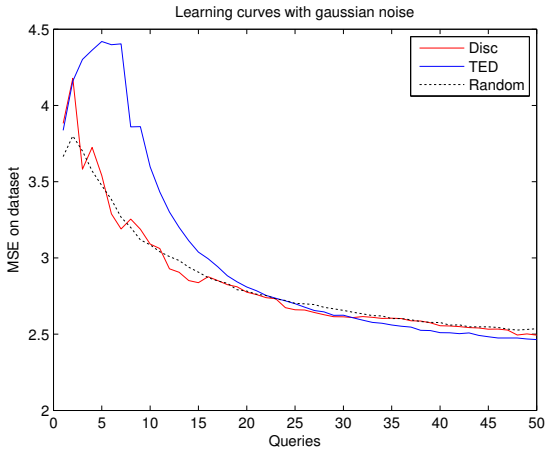


Figure 4.17: Results on ratio dataset where the labels were perturbed with Gaussian noise with $\sigma = 1.5$.

4.6.2. Real World Data

In the last subsection we saw an example that illustrates TED might not always be better than the MMD HS or the (nuclear) discrepancy active learner in the agnostic case. In this section we compare the performance of the active learners on the benchmark datasets using the original binary labels — thus note that this comparison is not artificial unlike our previous experiments. This corresponds to the agnostic case where $f \notin H$. We compare the results in this setting with the results of the realizable setting.

We evaluate the active learners on the datasets shown in Table 3.1. Some illustrating learning curves are shown in Figure 4.18 on page 69, and these results are summarized in Table 4.8 and Table 4.9 on page 68 using two tailed paired t-tests with $p = 0.05$. See Figure K.7 in the appendix for all learning curves.

We make three general observations in comparison with the realizable case. First, the learning curves seem to be less smooth. This is likely the case because the label function f is less smooth and more noisy in this scenario. The standard deviation also seem to increase, making it harder to distinguish which active learning methods perform the best. The dataset `german` is an extreme example of this, the error bars become an order of magnitude larger. Correspondingly this dataset has the largest mean squared error if a model is fitted to the whole dataset compared with other datasets which indicates f is not very well approximated by the hypothesis set.

Second, we observe smaller performance advantages of the active learners over random sampling than in the realizable setting. This may be because the approximation errors in all bounds in this setting are not zero. Because the active learners do not account for these approximation errors (these terms are ignored altogether by all active learners), the advantages of the active learners compared to random sampling become smaller.

Third, we see that the realizable case is a fair indicator of performance for performance in the agnostic setting, but not a perfect indicator. TED in most cases still performs the best, however it has lost its clear advantage observed in the realizable setting where its active learning curve was consistently lower. This might be due to the effects described above. For the `heart` dataset and `ringnorm` dataset (for large number of queries) TED actually seems to perform significantly worse than the MMD HS and discrepancy active learner for some number of queries, illustrating that indeed in the agnostic case it can happen that TED performs worse, as we also saw on the artificial dataset.

The discrepancy and the MMD HS already performed similar in the realizable case. In this setting the discrepancy and MMD HS active learners become even harder to compare. According to the t-tests in the Table 4.8 both methods tie much more than in the realizable case. The MMD HS only still shows a significant advantage for the datasets `sonar`, `ringnorm` and `ionosphere`. In the realizable case we observed that the discrepancy performs significantly worse than random

Dataset	TED vs Disc	TED vs MMD HS	Disc vs MMD HS
vehicles	0/10/0	1/9/0	3/7/0
heart	1/6/3	0/3/7	0/10/0
sonar	4/6/0	2/8/0	0/3/7
iris	8/0/0	8/0/0	0/8/0
thyroid	10/0/0	10/0/0	0/10/0
ringnorm	7/0/3	1/1/8	0/1/9
ionosphere	7/3/0	8/2/0	1/3/6
diabetes	0/8/2	0/10/0	1/9/0
twonorm	3/7/0	2/8/0	0/7/3
banana	5/4/1	7/2/1	2/8/0
german	1/9/0	3/7/0	1/9/0
splice	10/0/0	9/1/0	0/8/2
breast	10/0/0	6/4/0	0/6/4

Table 4.8: Win / tie / loss counts comparing the MMD HS, the discrepancy and the TED active learner on the benchmark datasets for the agnostic setting. We see TED performs the best in the majority of all experiments and the discrepancy performs the worst. In particular, just like in the realizable setting, MMD HS often matches or outperforms the discrepancy.

Dataset	ND vs TED	ND vs MMD HS	ND vs Disc
vehicles	0/9/1	1/9/0	0/9/1
heart	3/7/0	0/10/0	0/10/0
sonar	0/10/0	3/7/0	8/2/0
iris	0/0/8	1/7/0	0/8/0
thyroid	0/0/10	2/8/0	2/8/0
ringnorm	0/10/0	1/1/8	7/0/3
ionosphere	0/0/10	0/7/3	0/8/2
diabetes	0/9/1	0/8/2	0/7/3
twonorm	3/6/1	7/3/0	5/5/0
banana	1/6/3	6/4/0	2/8/0
german	0/10/0	2/8/0	1/9/0
splice	0/6/4	3/7/0	6/4/0
breast	0/0/10	1/9/0	2/8/0

Table 4.9: Win / tie / loss counts comparing the nuclear discrepancy (ND) with TED, the MMD HS and the discrepancy on the benchmark datasets for the agnostic setting. We see that the nuclear discrepancy often still outperforms or matches the performance of MMD HS and the discrepancy. TED performs better than the nuclear discrepancy in most cases, and the nuclear discrepancy almost never outperforms TED as in the realizable setting.

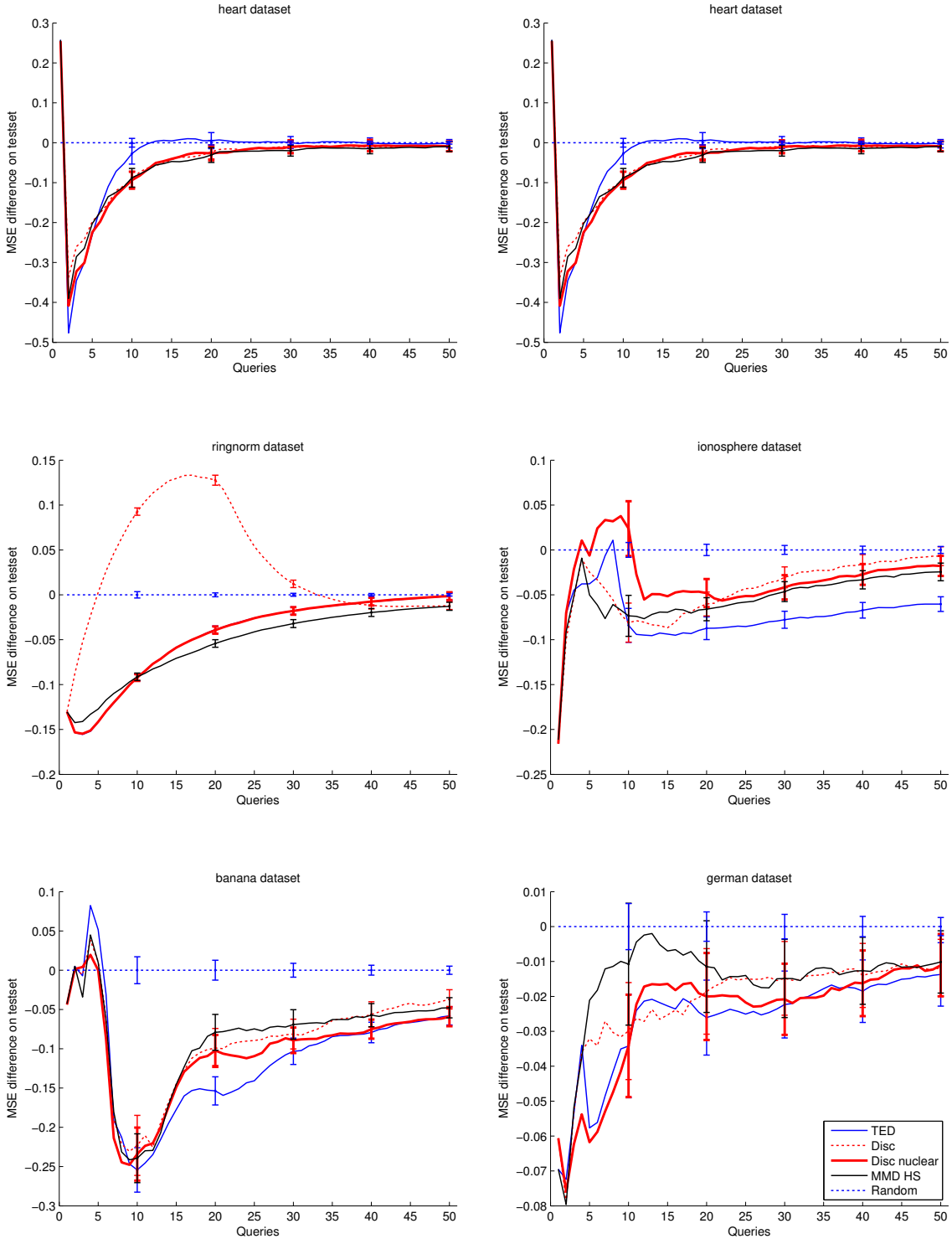


Figure 4.18: Some illustrating learning curves of the active learners on real world data in the agnostic setting where $f \notin H$. Observe that the learning curves of TED and the nuclear discrepancy overlap for the `ringnorm` dataset. See Figure K.7 for the results on all datasets.

sampling on the `ringnorm` dataset, we observed this also for some number of queries in this setting — however note that at the end of the active learning phase the discrepancy actually performs better than TED. In some cases the performance of the discrepancy has increased but in many cases it doesn't outperform other methods significantly.

Finally, the nuclear discrepancy still seems to outperform the MMD HS and discrepancy on most datasets, or at least matches their performance. The performance advantage of the nuclear discrepancy however has become smaller as well. The nuclear discrepancy performs similar to TED for some datasets as observed in the realizable setting. For example see the learning curve on the `ringnorm` dataset in Figure 4.18, where the learning curves of TED and the nuclear discrepancy overlap. Finally, the comparison of the nuclear discrepancy and TED in the agnostic setting is similar to the realizable setting. TED often seems to improve upon the nuclear discrepancy, and is almost never outperformed by the nuclear discrepancy. However also for the agnostic setting the advantage of TED over the nuclear discrepancy is smaller than in the realizable setting.

In the previous subsection we found that TED performed worse since there was a lot of label noise everywhere. There are also two other options that can explain differences between the agnostic and realizable setting, and we discuss them below. It is possible that because the approximation errors for the different active learners are different, some active learners may have smaller approximation than others, and therefore some active learners may work better than others in the agnostic case. If the approximation error is smaller, the main quantity in the bound more accurately describes the behavior in the agnostic setting. Therefore, this quantity is then more informative for active learning. Another possibility is that there are different subgroups in the dataset with different amounts of label noise. Some active learners could query consistently more samples from these noisy regions (for example because this region is deemed informative by this active learner) than others, and because of this these active learners may perform consistently worse.

We conclude the realizable setting gives a fair indication of the performance in the realizable setting. However, for some datasets the active learners perform quite different than in the realizable setting. This could be caused by label noise or because the approximation errors start playing a role in deciding which methods perform better. An in depth study of the approximation errors is required to understand the behavior in the agnostic setting.

5

Discussion

In this section we first summarize the results found in the previous chapter by revisiting our research questions. Afterward we discuss some choices we have made in our experiments and the possible impact on our results. We also discuss interesting directions for future work, and the implications of our theoretical results for other fields. Then we discuss our results in a broader context in the active learning research field. Finally we note the obstacles that have to be overcome before these methods are applicable to real world active learning settings.

5.1. Main Results

In this section we reflect on the main results of the experimental section by answering the research questions posed in the introduction.

Q1. Why and in what case can non-adaptive active learning methods improve upon random sampling?

A1. These methods exploit structure in the data together with the assumed smoothness conditions on the labeling function. Influential examples are chosen first that largely determine the labels of many other objects in the dataset and the selection of redundant samples is avoided. The more structure there is in the data that corresponds to the model assumptions, the more active learners can improve upon random sampling.

Q2. Is it beneficial to take the hypothesis set and the loss into account for active learning based on the MMD?

A2. Yes. We show using an artificial example what can go wrong if we do not take the hypothesis set and loss into account when using the MMD for active learning. We validated this claim on real world data. We show that if the bandwidth σ of the Gaussian kernel is chosen as described in our theoretical analysis in Section 2.1.4 this improves performance. This can even improve performance compared to the situation where the same σ is taken for the MMD and the learning algorithm, a similar choice which is often used in practice. Taking the hypothesis set and loss into account for the MMD has the most advantages in the realizable setting but also shows advantages on some real world datasets.

Q3. Will our introduced discrepancy active learning strategy improve upon the MMD active learning strategy as suggested by our theoretical analysis? (main question)

A3. Surprisingly, this is not the case. Initially our hypothesis was that since the discrepancy active learner has a tighter generalization bound, it would perform better than the MMD active learner. The discrepancy active learner can estimate the generalization error more accurately, and therefore we would expect that minimizing this bound would more accurately minimize

the generalization error. However, the discrepancy in many cases performs worse than the MMD, especially in the realizable case, a favorable setting for these methods. Using two artificial examples we show that the MMD can make more informed choices compared to the discrepancy in case an initially biased sample is used. This led us to believe the worst-case scenario considered by the discrepancy is less informative than the worst case considered by the MMD.

The discrepancy takes a very specific worst-case scenario into account which is extremely unlikely to occur in practice. The discrepancy uses all of its resources to avoid bad generalization performance in this scenario that almost never occurs. Because of this it sometimes performs poorly, since these queries do not always improve performance in practical scenarios. The fault of the discrepancy is that it only weighs this extremely unlikely worst-case scenario, and completely ignores other scenarios.

We show that the MMD takes a less specific worst-case scenario into account and because of this weighs average-case scenarios more than the discrepancy. These average-case scenarios occur more in practice. Therefore the MMD minimizes a quantity that is more relevant to the generalization error in practice. That is why the discrepancy can perform worse than the MMD in practice, yet the discrepancy generalization bound is tighter.

Our analysis suggests that worst-case scenarios assumed by the MMD and the discrepancy are too conservative. This is our motivation for the introduction of a new discrepancy measure, the nuclear discrepancy. This quantity weighs all scenarios equally (worst, average and best cases) unlike the MMD which weighs worst cases more than average cases. Our analysis suggests that the nuclear discrepancy weighs scenarios that are likely to occur in practice more than the MMD and the discrepancy. The proposed nuclear discrepancy active learner improves significantly upon both the discrepancy and the MMD active learner. The generalization bound of the nuclear discrepancy is looser than the discrepancy and MMD bound, underscoring that the tightness of a generalization bound is not relevant for active learning performance.

In summary, the discrepancy is too conservative: the worst case considered by the discrepancy is extremely unlikely to occur in practice. Because the MMD considers more likely scenarios it performs better in practice. Therefore the discrepancy does not improve upon the MMD, even though the discrepancy generalization bound is tighter. Finally, the nuclear discrepancy, which weighs all cases equally, outperforms both the MMD and the discrepancy. This is because the nuclear discrepancy weighs scenarios that are more likely to occur in practice more than the MMD and the discrepancy. Yet the nuclear discrepancy bound is looser than both the MMD and discrepancy bound.

Q4. How does the model-dependent TED active learner and its generalization bound compare with the discrepancy and MMD active learners and bounds?

A4. TED performs consistently better than the discrepancy and MMD active learner in the realizable case on real world data in terms of the squared loss. Using an artificial example we show that the MMD and (nuclear) discrepancy are more likely to choose redundant samples. Specifically the MMD and (nuclear) discrepancy try to match the sampling ratio's of the empirical distribution of the complete unlabeled dataset. Therefore the MMD and the (nuclear) discrepancy sample dense regions more often, which is unnecessary in the realizable case where there is no noise. TED does not show this behavior because using the analytical solution of the model it can consider more realistic worst-case scenarios. The performance advantage of TED is smaller in the agnostic case. In this setting TED is sometimes outperformed by other methods, however overall it performs the best. In most cases the TED bound is tightest when little labeled examples are available. Which bound is tighter depends on the dataset.

Q5. Is the tightness of a generalization bound related to performance in active learning?

A5. TED consistently outperforms all other methods in the realizable case, however its bound is not tightest in most cases. The same holds for the MMD and the discrepancy. The discrepancy bound is almost always tighter, yet the MMD active learner almost consistently outperforms the discrepancy active learner. Furthermore, the nuclear discrepancy generalization bound is looser than both the MMD and discrepancy bound, yet the nuclear discrepancy both outperforms the discrepancy and the MMD active learners. Thus the tightness of a generalization bound is not be indicative for active learning performance. Instead it is important that the scenarios taken into account by the generalization bounds should be realistic.

The reader might be wondering: is a worst-case analysis the correct way to address active learning? This depends on the goal: is one interested in playing it safe so active learning is beneficial even in a worst-case scenario, or is the goal to perform good on average? Given that active learning is used in case labeled data is very costly to acquire, a worst-case analysis is likely desirable in a real world setting, since if a 'greedy' active learning method fails to deliver good active learning performance in the real world, it is very costly to repeat the active learning procedure.

Furthermore, the inferior performance of the discrepancy is not a dent in the coffin of worst-case analysis for active learning. The problem is that the discrepancy only considers a very unlikely worst-case scenario which is either extremely unlikely or impossible. TED also uses a worst-case analysis and is very successful, this is because the worst-case considered by TED is much more likely to occur in practice.

5.2. Influence of Model Choices

In this section we briefly discuss our model choice and the possible impact on the results. We use kernel ridge regression without intercept and class priors. We choose this model since for this model all bounds are straightforward to compute. We leave the incorporation of priors and intercept into these generalization bounds for future work.

The most important comparison of the active learners takes place in the realizable case where no intercept is required, since the model that generates the labels uses no intercept term. Thus in these most important experiments the missing intercept is no issue.

Furthermore, we believe that the intercept will likely influence the active learning strategies little. The intercept can be incorporated by a feature that is the same for each sample in the RKHS of K. Since the active learners minimize the difference between the empirical distributions $\hat{P}$ and $\hat{Q}$ in some sense, we expect that such a constant feature will influence this comparison between $\hat{P}$ and $\hat{Q}$ little. Therefore we expect the active learning strategies will change little with the addition of an intercept term.

The intercept can improve performance of the model when unbalanced datasets are used in the agnostic case, because the average model output over the entire input space can be nonzero when an intercept term is added. However, since all active learners use the same model all methods have the same model mismatch. Therefore this comparison can be considered 'fair', and thus we expect if the experiments were repeated with intercept the results will be similar.

Class priors can be integrated in multiple ways in the ridge regression model. Since we measure performance in terms of the mean squared error, and this is exactly what is minimized by our learning algorithm, we do not expect class priors will change our results much. Improvement can only be made if our active learners somehow do not retain the original sampling ratio of both classes, but since our active learners aim to minimize sampling bias this is unlikely. Furthermore, all active learners suffer from this thus again the comparison can be considered 'fair'.

Finally, may the non-standard objective of the ridge regression model influence our results? We think that our results will change little if the standard ridge regression model is used. For details see Appendix H for the (non-)standard ridge regression model formulation. We choose this non-standard training procedure so the hypothesis set remains constant during active learning. This is important for our artificial experiments, otherwise for some training set sizes we could be in the realizable setting and not for other training set sizes. In appendix H it is shown that our non-standard training procedure can be incorporated by multiplying the regularization parameter of the

standard ridge regression number by the amount of samples in the training set. Since the strategy of the (nuclear) discrepancy and MMD active learners does not depend on the regularization parameter, these active learners will likely perform similar for different regularization parameters and thus also for the regular ridge regression model formulation. In the next section we discuss the influence of the regularization parameter on our results in the realizable setting. In particular, we show that a small change in the regularization parameter influences our results generally little. Therefore we expect that a different training procedure will influence our results little as well. However the theoretical analysis of such a different training procedure would be needlessly more complicated.

5.3. Influence of the Regularization Parameter on the Performance of TED

One of the advantages of TED is that it can take the regularization parameter of the learning algorithm into account in its active learning strategy. In our experiments in the realizable setting we set the regularization parameter of the model trained during active learning to the same value as the regularization parameter of the oracle model generating the labels. Thus TED has knowledge about the regularization parameter of the oracle model that generates the labels. Therefore we could wonder if TED has an unfair advantage: can it also outperform the other methods in the realizable setting if it does not have perfect information about this regularization parameter? Using the TED active learner we also investigate if taking the regularization parameter into account for active learning is beneficial in general.

First we introduce some notation and we clarify some things about the TED active learner. The discussion in this section involves three parameters. We have the regularization parameter of the oracle model that generates the labels, λ_{oracle}, the regularization parameter of the model that is trained during active learning, λ_{model}, and the regularization parameter used in the TED objective, λ_{TED}. In the experiments of the realizable setting in Section 4.3 we use $\lambda_{\text{oracle}} = \lambda_{\text{model}} = \lambda_{\text{TED}}$. This to ensure we are in the realizable setting and that the TED bound holds. In this section we study the case when there is model mismatch with respect to the regularization parameter, so $\lambda_{\text{oracle}} \neq \lambda_{\text{model}}$. In this case the TED bound still holds, however the TED bounds requires us to set $\lambda_{\text{TED}} = \lambda_{\text{model}}$, otherwise the bound does not hold. The actual value of λ_{oracle}, if it were known during active learning, does not influence the active learning strategy of TED through the TED bound[1].

We repeat the experiments of Section 4.3. We change the regularization parameter of the model trained during active learning, we set this parameter to $\lambda_{\text{model}} = a\lambda_{\text{oracle}}$. We repeat the experiments with four values of a: $a \in \{100, 10, 0.1, 0.01\}$. We rerun all experiments with *two* TED active learners. The first TED active learner uses $\lambda_{\text{TED}} = \lambda_{\text{model}}$, which is the proper choice of the regularization parameter for the TED objective. The second TED active learner uses $\lambda_{\text{TED}} = \lambda_{\text{oracle}}$. This is an improper choice of λ_{TED} for this setting as argued above since in this case the TED bound may not hold. Observe that this active learner selects the same samples as the TED active learner used in Section 4.3, since these methods share the same regularization parameter.

Why do we include this 'improper' TED active learner? If we compare these two active learners, we can see if setting the correct value of λ_{TED} influences the active learning strategy. For example, if these two active learners perform similar, we may conclude that the TED active learning strategy does not change much if λ_{TED} is changed. This would suggest that taking the regularization parameter into account for active learning is unnecessary. However, if these two active learners perform different, we may conclude that TED really uses information of the parameter λ_{TED} to change (and perhaps improve) its active learning strategy. This can thus shed light upon whether or not it is useful to construct active learning algorithms that take the regularization parameter into account.

[1] This is because the TED bound depends on λ_{oracle} through the parameter Λ, which does not influence the TED active learning strategy as discussed in Section 2.5

We must make the following remark. In case $\lambda_{\text{model}} < \lambda_{\text{oracle}}$ we are in the realizable scenario, since the hypothesis set of the model becomes larger than the hypothesis set of the oracle model, and thus must contain the oracle model. However, in case we use $\lambda_{\text{model}} > \lambda_{\text{oracle}}$, we may not be in the realizable setting. This is because the hypothesis set used by the model in this case is smaller than the hypothesis set used by the oracle model. In this case the approximation error in the bounds may not be zero. Furthermore, we consider the setting where the labels are generated by an oracle model, and therefore in this setting no regularization might be necessary, since there is no noise. So various factors are in play that complicate the interpretation of the results in this section. However, this comparison can still give us insight into whether or not regularization parameter (in)dependent active learners are desirable in case of model mismatch with respect to the regularization parameter.

First we discuss the results for the case $\lambda_{\text{model}} < \lambda_{\text{oracle}}$. These results are very similar to the results in the realizable setting, and therefore we defer these results to the appendix. The learning curves can be found in Figure K.8 and Figure K.9 on page 137 and page 138. Both TED active learners perform very similar. The TED active learner that uses the proper regularization parameter (λ_{model}) outperforms or matches the performance of the TED active learner with the improper regularization parameter (λ_{oracle}). The observations of the realizable setting still hold: the TED active learner in almost all cases outperforms all other methods. Furthermore, the nuclear discrepancy in general performs the second best and obtains similar learning curves as TED. The MMD also outperforms the discrepancy or matches its performance. We conclude the relative performances of the methods changes little when making λ_{model} smaller than λ_{oracle} in this setting[2].

For $\lambda_{\text{model}} > \lambda_{\text{oracle}}$ we obtain more interesting results. We first review the results for $\lambda_{\text{model}} = 10\lambda_{\text{oracle}}$. Some illustrating learning curves are given in Figure 5.1 on page 76, all learning curves can be found in the appendix in Figure K.10. TED with the proper regularization parameter (λ_{model}) clearly changes its strategy in this case, and because of this it performs significantly better in almost all cases than TED with the improper regularization parameter (λ_{oracle}). See for example the learning curves on the datasets `heart`, `ionosphere`, `diabetis`, `twonorm`, `breast` and `german`. The active learning strategy of TED which uses the improper regularization parameter performs significantly worse than other methods on some datasets, see for example the learning curves on `sonar`, `twonorm`, `ionosphere`, `diabetis`, `german` and `diabetis`. Sometimes TED with the improper regularization parameter can even perform worse than random sampling, see the learning curves on the datasets `ionosphere`, `diabetis` and `breast`. As in the realizable setting, the TED active learner with the proper regularization parameter outperforms the other active learning methods almost consistently or matches their performance. Furthermore, the same trends as in the realizable setting are observed for the other active learners. These results suggest that it is beneficial to take the regularization parameter into account when designing an active learning strategy.

Now we review the results for $\lambda_{\text{model}} = 100\lambda_{\text{oracle}}$. Some illustrating learning curves are given in Figure 5.2 on page 77, all learning curves can be found in the appendix in Figure K.11. In this case TED with the proper regularization parameter (λ_{model}) sometimes improves upon the performance of the TED active learner with an improper regularization parameter (λ_{oracle}) and the other active learning methods, see the learning curves on the datasets `thyroid`, `ionosphere`, `german`, `banana`, `breast` and `diabetes`. However TED with the proper regularization parameter (λ_{model}) can perform significantly worse as well, see the learning curves on the datasets `splice`, `twonorm`, `thyroid` and `iris` where it performs worse than the other active learning methods and in some cases worse than random sampling.

This behavior could have two causes. Since $\lambda_{\text{TED}} = \lambda_{\text{model}}$ is so large, TED might sample too many redundant samples as was observed in Sections 2.3.1 and 4.3.1. Also, the TED quantity becomes larger[3] if a larger value of λ_{model} is used, and thus the TED bound becomes weaker. This in contrast with the MMD and (nuclear) discrepancy bounds which become stronger if a larger value of λ_{model} is used.

[2]This also indicates that in this scenario no overfitting occurs since noise is absent as noted before, otherwise we would have seen this reflected in the (absolute) learning curves.

[3]For this argument we ignore the parameter Λ for TED, since this parameter corresponds to the oracle model.

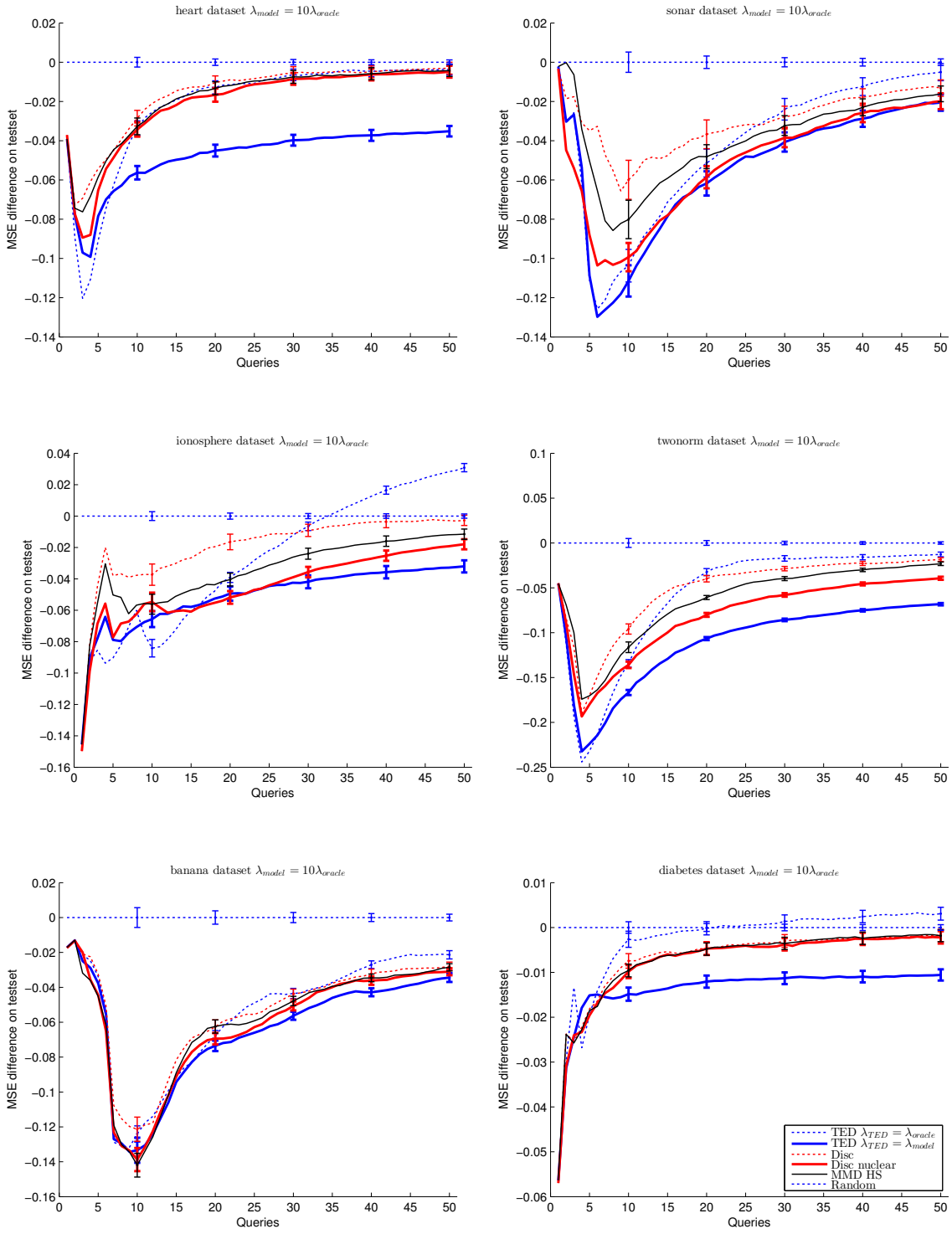


Figure 5.1: Some illustrative learning curves for the setting where there is model mismatch with respect to the regularization parameter. Here we set $\lambda_{\text{model}} = 10\lambda_{\text{oracle}}$. In this case we are not in the realizable setting. Observe that TED with the proper regularization parameter (λ_{model}) generally outperforms all other active learning methods as in the realizable setting. Furthermore, observe that TED with the proper regularization parameter (λ_{model}) performs significantly better than TED with an improper regularization parameter (λ_{oracle}). TED with an improper regularization parameter can even perform worse than random sampling. This shows that TED adapts its active learning strategy in a meaningful way based on the regularization parameter, and suggests that taking the regularization parameter of the model into account is beneficial for active learning. All learning curves are given in Figure K.10 in the appendix.

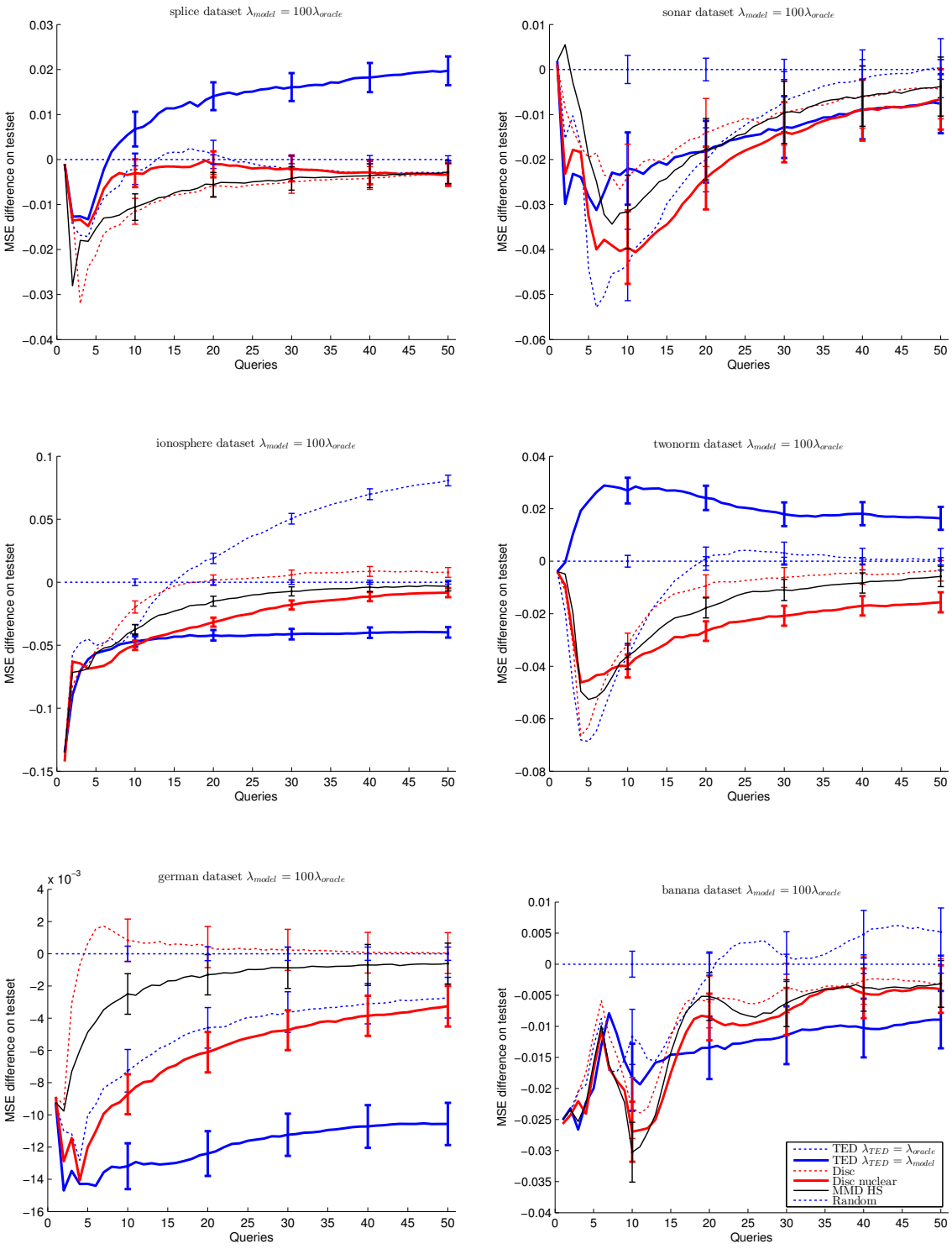


Figure 5.2: Some illustrative learning curves for the setting where there is model mismatch with respect to the regularization parameter. Here we set $\lambda_{\text{model}} = 100\lambda_{\text{oracle}}$. In this case we are not in the realizable setting. Observe that TED with the proper regularization parameter (λ_{model}) sometimes performs better than other methods, but in some cases also performs much worse, even worse than random sampling. This shows that TED cannot deal with a too large model misspecification or large regularization parameter. Observe that the MMD HS and nuclear discrepancy active learners are relatively robust to model misspecification and in all cases perform better than random sampling. For all learning curves see Figure K.11

Another explanation is that the model mismatch has become so large that the approximation error of TED might have increased significantly compared with the approximation errors of the other active learner methods. Because of this the TED generalization bound can become weaker. To investigate this an in depth analysis of the approximation errors of the active learners has to be done which is outside the scope of this work.

Observe that in some cases it is more desirable to choose the improper regularization parameter (λ_{oracle}), see for example the learning curves on `splice` and `twonorm` in Figure K.11. This is quite counterintuitive, since the TED bound in this case should not hold at all.

The performance of the active learning methods that are regularization parameter independent is quite consistent across different values of λ_{model}. Just like in the realizable setting the nuclear discrepancy generally outperforms or matches the performance of the MMD HS active learner, except for the `splice` dataset. In particular, both the MMD HS and nuclear discrepancy consistently perform better than random sampling even if there is model mismatch with respect to the regularization parameter. These methods may be more robust because they simply do not depend on the regularization parameter. Therefore these active learners are especially preferable if the optimal regularization parameter is unknown, as is the case in a real world active learning setting.

Now we come back to the main question of this section: does TED outperform the other methods because it has knowledge of the oracle regularization parameter? Likely, yes. If the regularization parameter somewhat matches the regularization parameter of the oracle model, TED outperforms the other methods and adjusts its strategy in a meaningful way. However if the regularization parameter is chosen more than an order too large compared to the oracle model, TED can perform worse than the other methods and even worse than random sampling. This might be the consequence of a large approximation error in the TED bound, or might be an indication that TED only works well with a small regularization parameter. If the regularization parameter is chosen too small, this affects the performance of the TED active learner little in our setting.

Our results indicate that the MMD HS and the nuclear discrepancy active learners are more robust to model misspecification with respect to the regularization parameter: both consistently perform better than random sampling unlike TED. This indicates that in a real world active learning setting, where we do not know the optimal regularization parameter, the MMD HS and nuclear discrepancy might be preferred over the TED active learner. Especially the nuclear discrepancy is preferable since in most cases it performs better or matches the performance of MMD HS. However, if we know the optimal regularization parameter approximately, TED is more preferable since it can use this information to improve its active learning strategy[4].

5.4. Sample Reweighting During Active Learning

In this section we consider if sample reweighting can improve performance of the discrepancy and the MMD during active learning. This question naturally follows from the fact that algorithms exist from the field of sample bias correction to reweight training data to minimize the MMD and discrepancy.

In case sample reweighting is used, for each sample different weights can be chosen: because of this the definition of the MMD and the discrepancy will change. We propose to reweight samples to minimize the MMD and discrepancy in each iteration of the active learning process, and to reweight samples in each set that is constructed to evaluate queries as described in Section 2.5. The ridge regression model has to be adapted to take the weights into account. See [4] and [3] for details.

In Section 4.3.1 it was shown that TED can perform better than the (nuclear) discrepancy and the MMD in the realizable setting because it avoids redundant samples. The MMD and discrepancy will only need one sample per cluster in this artificial example if sample reweighting is used, since they will use sample reweighting to exactly match the ratios of the unlabeled dataset. Thus sample reweighting can make the MMD and (nuclear) discrepancy more aggressively sample the feature space like TED.

[4] Unless perhaps the regularization parameter of the oracle model is quite large, in which case TED might select too much redundant samples. However, we did not evaluate this and defer this to future work.

However, we believe that even if the discrepancy and MMD are used in combination with sample reweighting, TED will likely still perform better. Because TED has more knowledge of $w - w'$, it can consider more likely worst-case scenarios as argued in Section 4.4.3. Furthermore, since the active learners already aim to minimize the MMD and discrepancy by choosing samples, reweighting in practice will likely have little effect: since the samples are already chosen to minimize sampling bias, reweighting to correct sample bias will likely be unnecessary in most cases.

Sample reweighting will likely make the discrepancy less greedy compared to the MMD. For example if sample reweighting had been used in the artificial example of Section 4.4 the discrepancy would reweight the oversampled cluster. Because of this the discrepancy would be able to focus on minimizing more relevant eigenvalues of the matrix M as well, and will likely match the performance of the MMD in this artificial example.

Even if both the MMD and discrepancy active learners are combined with sample reweighting, the MMD still likely will perform better in the active learning scenario. This is because in that case our analysis in Section 4.4.3 still holds: the discrepancy will only consider an unlikely worst-case scenario while the MMD considers a more likely average-case scenario as well. Therefore the MMD will likely perform better even if reweighting is used.

Finally, reweighting data in combination with active learning is computationally expensive. We believe reweighting is only justified if active learning is combined with transfer learning, sample bias correction or domain adaptation. In the regular active learning setting reweighting likely only provides marginal benefits with great computational cost.

5.5. Extension to Non-Adaptive Strategy

In this section we discuss how the active learners in this work can be extended to take the labels of $\hat{Q}$ into account to make label dependent active learners. We believe adaptive extensions of these active learners could be a very promising direction for future work.

An adaptive active learning strategy is proposed for the MMD in [9]. The discrepancy active learner can be adapted in a straightforward way to take into account labeled data using the same approach. However, this requires a parameter β that balances minimizing the empirical loss and the discrepancy (or in other words: to balance exploitation and exploration). It is unclear how to set β without performing multiple active learning experiments on the same dataset. We believe that the active learning strategies in this work can be adapted in a more meaningful way, and we discuss these proposed approaches below for each active learner.

First we discuss how to adapt the discrepancy. Recall that the discrepancy maximizes over $u = w' - w$, where w is the model trained during active learning and w' is the model generating the labels of the dataset. The discrepancy generalization bound can be adapted to maximize only over w', since w is known (or can be approximated) during the active learning process if we have knowledge about the labels. In that case, it is desirable to constrain w' so that $w \approx w'$, so the discrepancy active learner becomes less conservative and considers more likely worst-case scenarios. For example, the function set $\{\forall w' | L_{\hat{Q}}(w', f) \leq r^2 \wedge ||w'||_K \leq \Lambda^2\}$ could be used as suggested by [15]. The parameter r in this case controls the conservativeness of the active learning algorithm. The involved optimization problem for choosing a sample is then however NP-hard even in a sequential setting. However, [15] proposes a sampling-based approximation that turns the corresponding domain adaptation algorithm into a simple quadratic program. Perhaps such a strategy is applicable to the proposed active learning strategy as well.

We believe this analysis could be a very promising direction for future work, since it addresses the weakness of the discrepancy active learner. If r is chosen small, this active learner assumes w and w' are more similar, and the active learner becomes less conservative. The only difficulty in this case is to tune the parameter r for active learning. In [15] cross validation is used to tune this parameter, which is impossible in the active learning setting, since this would require multiple active learning experiments in practice.

This strategy to improve the discrepancy can also be applied to the TED active learner. TED, like the discrepancy, maximizes over all possible $w' \in H$, see Equation D.3. Instead of maximizing over all $w' \in H$, TED can be similarly adapted to maximize over the function set

$\{\forall w' | L_{\hat{Q}}(w', f) \leq r^2 \wedge ||w'||_K \leq \Lambda^2\}$. However, the corresponding optimization problem is NP-hard for TED as well. Possibly a sampling based approach is also possible to apply here.

A similar strategy is also directly applicable to the MMD. The MMD approximates the true loss function g by a loss function $\tilde{g} \in H$. After the training procedure we know the true loss function on the set $\hat{Q}$. Using linear constraints we can force $\tilde{g}$ to be similar to the true loss function g on the set $\hat{Q}$. Because of this, the MMD will consider more realistic scenarios. These linear constraints will necessarily depend on a parameter r that needs to be tuned. The resulting maximization over $\tilde{g} \in H$ has a linear objective function for the MMD, and only convex quadratic and linear constraints, and is thus convex. Perhaps therefore an initial investigation using the MMD should be preferred, since it would be easy to implement.

Finally, such an analysis for the nuclear discrepancy does not seem straightforward. This is because the nuclear discrepancy does not have an interpretation where the labeling function or loss function is approximated by a function of the hypothesis set.

5.6. Implications of our Results for Other Fields

In this section we briefly describe the implications of our results for other fields.

We have shown the benefits of our theoretical analysis that describes how to adapt the MMD bound to take the hypothesis set and loss into account. In the experiments of [4] where the discrepancy and the MMD are compared for sample bias correction, the authors used $\sigma_{\text{MMD}} = \sigma_{\text{RR}}$. We have shown that in active learning that this choice can lead to suboptimal results. A comparison of the MMD and the discrepancy when the MMD takes the hypothesis set and loss into account in sample bias correction would therefore be very interesting, since in that case the MMD might outperform the discrepancy as in active learning.

Furthermore any comparison between MMD and discrepancy becomes more insightful when the MMD takes the hypothesis set and loss into account, because the assumptions of both bounds are then comparable. Our comparison of the MMD bound and discrepancy bound in terms of the eigenvalues of the matrix M can possibly be used to understand the differences between the performance of both methods in other fields.

Finally, we believe the nuclear discrepancy may perform better than the discrepancy or the MMD in domain adaptation, since it also performs better in the active learning scenario. In general our results describe desirable properties of non-probabilistic generalization bounds for active learning: in particular that the bounds should account for likely scenarios. These properties might be beneficial to consider for other fields which use generalization bounds as well.

5.7. TED for Domain Adaptation

Considering that we found that TED generally performs the best in active learning, we might wonder if TED can also be extended to domain adaptation, sample bias correction or transfer learning.Especially if one is interested in the squared loss TED is appropriate.

Possibly TED can improve upon the MMD and the discrepancy in these fields. Especially since in these fields it is possible to optimize the hyper parameters of the model using cross validation since labeled data is available (of the source distribution and perhaps even the target distribution). Then TED can use its information of the optimal regularization parameter to gain an advantage over the MMD and discrepancy which ignore the regularization parameter.

We must make an important observation: TED assumes that all labels of all samples are generated by $f \in H$. However, in other fields, it is not uncommon to assume that only samples near the target distribution are generated by a hypothesis $f \in H$, and for other samples it may be that $f \notin H$. We illustrate what can go wrong if TED is used in this case for domain adaptation using the artificial example in Figure 5.3 from [3]. Here the target function is linear, while the source function is a polynomial. Since the target function is linear the linear kernel is used for this example. In the linear kernel the TED objective favors objects with large norm when used for active learning. Similarly, TED will likely assign the objects with the largest norm the largest weights in domain adaptation. In this example TED will thus weight the blue objects on the right

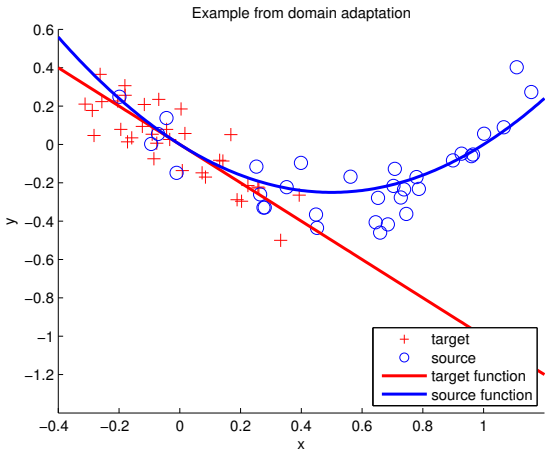


Figure 5.3: Example from domain adaptation where $f \in H$ for the target sample, but where $f \notin H$ for the source sample. We expect if TED is used on this artificial example it will reweight the objects with the largest norm the most, and thus fails to recover the target function.

the most. Using these reweighted samples the linear target function cannot be learned well. This might be less problematic in the Gaussian kernel where the norm of each sample is constant. Through artificial examples it is shown that the MMD and the discrepancy can take the situation into account where $f \notin H$ for the source distribution [3, 15].

In [3] it is suggested that reweighting should be done according to the ratio $\frac{P}{Q}$, which is known as importance weighting. However, as we have seen in Section 4.3.1, TED does not select samples in active learning to obtain similar ratio's between the datasets $\hat{P}$ and $\hat{Q}$. Therefore, TED will likely not perform reweighting according to the ratio $\frac{P}{Q}$ in domain adaptation. Because of this reason, it is especially interesting to know if TED performs well in domain adaptation. This could show the benefits of not reweighting according to the ratio $\frac{P}{Q}$ as also suggested by [15] and [19]. We wonder if TED can match the performance of the complicated domain adaptation algorithm proposed by [15], since both methods assume $w \approx w'$ and both optimize the weights for the specific hypothesis under consideration. However, the optimization of the TED objective combined with reweighting could prove to be a difficult optimization problem.

We conclude TED possibly needs to be adapted to accommodate the assumption that $f \notin H$ for the source distribution if it were to be successfully applied in other fields such as domain adaptation, sample bias correction or transfer learning. Since TED can use knowledge of λ it can possibly outperform the MMD and discrepancy in these fields as was also shown in our active learning experiments. Especially it is interesting to see if TED can outperform the generalized discrepancy domain adaptation algorithm of [15], since both rely on similar assumptions.

5.8. Active Learning for Different Models

In this section we briefly discuss active learning for different models, or in other words, for different loss functions.

Our results suggest that model dependent bounds such as the TED bound can be desirable for active learning. Possibly such a model dependent bound can be more informative since it can depend on the regularization parameter of the model and it can take the training procedure into account. However, often these bounds are hard to derive for more complex models. In these cases the MMD bound can be used to easily create a non-adaptive active learning algorithm. Our results suggest it would be best to take the hypothesis set and loss into account when using the MMD for active learning. Therefore it would be interesting to investigate which kernel should be used to take the hypothesis set and loss into account for other models, such as SVM's or logistic regression.

For the squared loss the discrepancy performs worse than the MMD for active learning, but has a tighter bound. It remains unclear whether or not for other losses an active learning strategy

based on the discrepancy can improve upon the MMD. Likely our analysis that the discrepancy considers a too unlikely worst-case scenario holds for other losses as well. However, computing the discrepancy for other losses seems to be computationally challenging, since these optimization problems seem to be non-convex. Therefore, an analysis in terms of the MMD which can also take the hypothesis set and loss into account could be more promising for future work, also in view of our negative results for the discrepancy.

Finally, unlike the discrepancy the nuclear discrepancy is not straightforward to generalize to other losses. However, perhaps the following quantity:

$$\mathop{\mathbb{E}}_{u}\left[\left|L_{\hat{P}}(w, w') - L_{\hat{Q}}(w, w')\right|\right]$$

in combination with proper probabilistic assumptions on $u = w - w'$ can be used to derive similar (probabilistic or non-probabilistic) generalization bounds for other losses as well. Likely such an analysis could lead to active learning algorithms that improve upon the MMD and discrepancy for other models as well.

5.9. Obstacles to Real World Application

In this section we discuss which obstacles need to be tackled in order to make the methods in this work applicable to real world scenarios. We discuss the agnostic setting, how to evaluate active learners, setting hyper parameters, the zero-one loss, and the performance of active learning methods compared with random sampling.

TED and the nuclear discrepancy are superior in the realizable case compared to the MMD and the discrepancy. Yet sometimes both were outperformed in the agnostic case. It remains unclear on what this is dependent. Possibly redundant samples chosen by the discrepancy and the MMD average out noise as suggested in Section 4.6.1. How can we adapt the active learning methods to account for this? We believe an in depth study of the approximation error of the generalization bounds is necessary. A good place to start would be a scenario where the labels are generated by a model of a different hypothesis class. For example what happens if a different kernel bandwidth is used for the oracle model that generates the labels? Perhaps it is possible to predict or bound the approximation error in this case. Studying the approximation error of all methods can be used to determine which bounds are appropriate in which scenarios. Possibly an extension to probabilistic labeling functions could be useful to this end as well.

We have looked at learning curves mostly qualitatively. However, for real world application it might be desirable that the active learning method performs well after x queries. For example, if the labeling budget is 50 samples, it might not matter how the active learner performs after 10 queries. We however looked at the performance of all active learners for all number of queries. This suggests that for some settings a different method of measuring performance is necessary. One could characterize if active learning methods are better for large or small number of queries. Furthermore, the size of the labeling budget might be important for determining the optimal active learning strategy. We have ignored these aspects altogether, but these aspects are important for the applicability and comparison of active learning methods.

It remains an open problem how to set the hyper parameters in a real world active learning setting. This is closely related to the analysis of the approximation errors of the bounds. There is a trade-off: the smaller λ, the larger the hypothesis class. A larger hypothesis class results in a smaller approximation error in the generalization bounds, but a larger chance of overfitting the model. Furthermore, for larger hypothesis classes the larger the (nuclear) discrepancy quantity, the MMD quantity and the TED objective and thus the larger the main term in these generalization bounds. The bandwidth of the Gaussian kernel has a similar trade off: the larger the bandwidth, the smaller the (nuclear) discrepancy quantity, the MMD quantity and the TED objective. However the larger kernel bandwidth the larger the approximation error: the model predictions might be too smooth and underfitting can occur. Possibly these generalization bounds (including the estimation of approximation error) can be used to gain more insight in how to tune the hyper parameters of the model during active learning.

We have investigated the active learning methods in this work in terms of the surrogate loss of the model. While the squared loss is appropriate for regression, for classification we are generally more interested in the misclassification or zero-one loss. A bound in terms of zero-one loss would be most desirable for classification. It is possible to plug the zero-one loss directly in the definition of the discrepancy since the discrepancy is compatible with any loss function. Or it can be studied which kernel corresponds to the loss function for the zero-one loss to use the MMD HS active learning strategy. However the involved maximization will likely be difficult to solve. A different approach is to study the relation between surrogate losses and the zero-one loss. This is an important open question for machine learning in general. Observe however that if one is only interested in the surrogate loss (for example if one is interested in regression with the squared loss) all our methods are directly applicable.

Finally, one piece of the puzzle is missing. We have characterized qualitatively in which cases the methods studied in this work can improve upon random sampling. However, it remains unclear if the active learners will be able to improve upon random sampling and by how much, even in the realizable setting. A probabilistic interpretation is likely required to analyze this. Perhaps the generalization bounds in this work can be adapted to generalization bounds that hold with a certain probability. Such an analysis may be able to provide a probability quantifying how likely it is an active learning method can improve over random sampling. This would be extremely desirable if these methods are to be applied in practice.

6

Conclusion

We compared four active learning strategies that are based on generalization bound minimization: the state-of-the-art TED and state-of-the-art MMD active learner and our introduced discrepancy and nuclear discrepancy active learners. We have shown why these label independent active learning techniques are generally useful for active learning and we explained the trends observed in the learning curves. We gave a qualitative characterization of the conditions in which these active learners can improve upon random sampling.

An in depth theoretical analysis of the generalization bounds was given. Our first theoretical result describes how to take the hypothesis set and loss into account for the MMD measure. Empirical evidence on real world data suggests taking into account the hypothesis set and loss for the MMD active learner is beneficial. These results have implications for other fields where the MMD is used as well.

We compared the MMD and discrepancy generalization bound for the squared loss. We have shown the relation between these bounds. Our second novel theoretical result is that the discrepancy generalization bound is always tighter than the MMD bound under the same assumptions.

Our theoretical results indicate the discrepancy bound is a more accurate estimator of the generalization error since the bound is tighter. Surprisingly, our introduced discrepancy active learner performs worse than the MMD in most experiments and rarely performs better. This is a counter intuitive result that suggests tighter generalization bounds do not guarantee better active learning performance.

We found that the worst-case scenario considered by the discrepancy is extremely unlikely to occur in practice. The MMD instead considers average-case scenarios as well which are much more likely to occur in practice. Because of this the MMD performs better even though its bound is looser.

However, the MMD weighs worst-case scenarios more than average and best-case scenarios. Instead, we have shown that in practice all scenarios contribute equally to the generalization error. We underscored this issue by introducing a novel active learning algorithm based on the nuclear discrepancy, a novel quantity to estimate the differences between empirical distributions. The nuclear discrepancy weighs all scenarios equally. We have shown the nuclear discrepancy active learner outperforms both the MMD and the discrepancy active learner significantly in the realizable setting. Yet the bound of the nuclear discrepancy is looser than the discrepancy and MMD bound, confirming that the tightness of a generalization bound is not of importance for active learning performance. Instead it is important that the generalization bound considers realistic scenarios that are likely to occur in practice, instead of focusing too much on unlikely scenarios.

Furthermore we have shown that the model dependent TED bound leads to an active learning algorithm that almost consistently outperforms the MMD and (nuclear) discrepancy active learners in the realizable setting. This illustrates the power of model dependent generalization bounds for active learning. The performance advantage of TED can be explained by the fact it has knowledge of the training procedure and of the regularization parameter. Therefore TED can consider more realistic scenarios.

Even though TED performs the best, its bound is not always the tightest, once more confirming that the tightness of the generalization bound is not important for active learning performance. Since TED performs the best in our active learning setting, we conclude that the TED criterion can possibly be successful in other fields as well when the squared loss is the loss of interest.

We performed most experiments in the realizable setting where it is most meaningful to compare the bounds. This setting is meaningful since in this case the approximation errors of the generalization bounds which cannot be minimized vanish. We have demonstrated that the results of the realizable setting do not always generalize to the agnostic setting but do give a good indication of performance.

We found preliminary results suggesting that the MMD and nuclear discrepancy active learners might be more robust to large model misspecification in terms of the regularization parameter than the TED strategy; in all of our experiments with model misspecification these methods consistently performed better than random sampling unlike TED. Since in real world active learning applications one does not know the optimal regularization parameter, the nuclear discrepancy strategy which performs generally the best is more desirable in such a setting.

A more in depth study is required to diagnose why TED does not perform well in case of model misspecification with respect to the regularization parameter. More research is necessary to understand the agnostic case in general which is important for real world applications of these active learning methods. Perhaps such an investigation can lead to improved active learners or can shed light on how to tune hyperparameters during active learning.

It will be interesting to apply the analysis of [15] to the active learners in this work. This way, label dependent active learning strategies can be constructed that consider more realistic scenarios. Because we have shown that active learners that consider more relevant scenarios perform better, we expect that this will be a fruitful direction for future work.

In conclusion, we found that the discrepancy generalization bound when used for the squared loss, while tighter, does not improve active learning performance compared to the MMD active learning strategy. Instead we found that it is important that active learners take realistic scenarios into account. The more likely the scenarios considered by the active learners the better they perform. Our described generalization bound based on our novel quantity called the nuclear discrepancy confirms these results. The nuclear discrepancy bound is looser than the MMD and discrepancy bound, yet the active learner performs better in the realizable setting. In particular we found preliminary results that the nuclear discrepancy generally performs the best under large model misspecification with respect to the regularization parameter, even better than the state-of-the-art TED and MMD active learners. Finally, our qualitative description of desirable properties for generalization bounds can be very useful for designing new active learning algorithms based on generalization bound minimization, and may also apply to other fields that use generalization bound minimization.

Bibliography

[1] B. Settles, *Active Learning,* Synthesis Lectures on Artificial Intelligence and Machine Learning **6**, 1 (2012).

[2] A. Liu, L. Reyzin, and B. D. Ziebart, *Shift-pessimistic Active Learning Using Robust Bias-aware Prediction,* in *Proceedings of the 29th AAAI Conference on Artificial Intelligence* (2015) pp. 2764–2770.

[3] J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, and B. Schölkopf, *Correcting sample selection bias by unlabeled data,* in *Proceedings of the 19th Conference on Advances in Neural Information Processing Systems (NIPS)* (2007) pp. 601–608.

[4] C. Cortes and M. Mohri, *Domain adaptation and sample bias correction theory and algorithm for regression,* Theoretical Computer Science **519**, 103 (2014).

[5] R. Rifkin, G. Yeo, and T. Poggio, *Regularized least-squares classification,* Advances in Learning Theory: Methods, Model, and Applications **190**, 131 (2003).

[6] R. Chattopadhyay, Z. Wang, W. Fan, I. Davidson, S. Panchanathan, and J. Ye, *Batch Mode Active Sampling Based on Marginal Probability Distribution Matching,* in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)* (2012) pp. 741–749.

[7] Q. Gu, T. Zhang, J. Han, and C. H. Ding, *Selective Labeling via Error Bound Minimization,* in *Proceedings of the 25th Conference on Advances in Neural Information Processing Systems (NIPS)* (2012) pp. 323–331.

[8] K. Yu, J. Bi, and V. Tresp, *Active Learning via Transductive Experimental Design,* in *Proceedings of the 23rd International Conference on Machine Learning (ICML)* (2006) pp. 1081–1088.

[9] Z. Wang and J. Ye, *Querying Discriminative and Representative Samples for Batch Mode Active Learning,* in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)* (2013) pp. 158–166.

[10] Q. Gu, T. Zhang, and J. Han, *Batch-Mode Active Learning via Error Bound Minimization,* in *Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence (UAI)* (2014).

[11] L. Bottou, *Large-Scale Machine Learning with Stochastic Gradient Descent,* in *Proceedings of the 19th International Conference on Computational Statistics (AISTATS)* (2010) pp. 177–186.

[12] Y. Mansour, M. Mohri, and A. Rostamizadeh, *Domain Adaptation: Learning Bounds and Algorithms,* in *Proceedings of the 22nd Annual Conference on Learning Theory (COLT)* (2009).

[13] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning* (MIT press, Cambridge, Massachusetts, 2012).

[14] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, *A Kernel Two-sample Test,* Machine Learning Research **13**, 723 (2012).

[15] C. Cortes, M. Mohri, and A. M. Medina, *Adaptation Based on Generalized Discrepancy,* Machine Learning Research (forthcoming).

[16] K. Yu, S. Zhu, W. Xu, and Y. Gong, *Non-greedy Active Learning for Text Categorization Using Convex Transductive Experimental Design,* in *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (2008) pp. 635–642.

[17] Q. Gu and J. Han, *Towards Active Learning on Graphs: An Error Bound Minimization Approach,* in *Proceedings of the 12th IEEE International Conference on Data Mining (ICDM)* (2012) pp. 882–887.

[18] R. El-Yaniv and D. Pechyony, *Transductive Rademacher Complexity and Its Applications,* in *Proceedings of the 20th Conference on Learning Theory (COLT)* (2007) pp. 157–171.

[19] C. Cortes, Y. Mansour, and M. Mohri, *Learning Bounds for Importance Weighting,* in *Proceedings of the 23rd Conference on Advances in Neural Information Processing Systems (NIPS)* (2010) pp. 442–450.

[20] R. Ganti and A. Gray, *UPAL: Unbiased Pool Based Active Learning,* in *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS)* (2012) pp. 422–431.

[21] Z. Wang and J. Ye, *Querying discriminative and representative samples for batch mode active learning,* in *ACM Transactions on Knowledge Discovery from Data* (2013) p. 158.

[22] Q. Gu, T. Zhang, C. Ding, and J. Han, *Selective Labeling via Error Bound Minimization,* in *Advances in Neural Information Processing Systems 25* (2012) pp. 332–340.

[23] C. A. Micchelli, Y. Xu, and H. Zhang, *Universal Kernels,* Machine Learning Research **7**, 2651 (2006).

[24] C.-C. Chang and C.-J. Lin, *LIBSVM: A library for support vector machines,* ACM Transactions on Intelligent Systems and Technology **2**, 27:1 (2011).

[25] B. Caputo, K. Sim, F. Furesjo, and A. Smola, *Appearance-based Object Recognition using SVMs: Which Kernel Should I Use?* in *Proceedings of the Advances in Neural Information Processing Systems (NIPS) workshop on Statistical methods for computational experiments in visual processing and computer vision* (2002).

[26] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis* (Cambridge University Press, Cambridge, UK, 2004).

[27] A. Gretton, *A Kernel Two-Sample Test,* Journal of Machine Learning Research **13**, 723 (2012).

A

MMD

A.1. Derivation of the MMD Generalization Bound

In this section we derive a MMD bound that depends on the assumption that $g \in H'$. In appendix A.3 we give the derivation of the MMD bound where this assumption is relaxed. This bound does not assume i.i.d. samples and thus can be applied to the active learning scenario. At the end of this section we show how to compute the MMD quantity.

The MMD bound can be derived as follows:

$$
\begin{aligned}
L_{\hat{P}}(h,f) - L_{\hat{Q}}(h,f) =& \frac{1}{n_{\hat{P}}} \sum_{x \in \hat{P}} L(h(x), f(x)) - \frac{1}{n_{\hat{Q}}} \sum_{x \in \hat{Q}} L(h(x), f(x)) \\
=& \frac{1}{n_{\hat{P}}} \sum_{x \in \hat{P}} g(x) - \frac{1}{n_{\hat{Q}}} \sum_{x \in \hat{Q}} g(x) && \text{(A.1)} \\
\leq& \max_{\tilde{g} \in H'} \frac{1}{n_{\hat{P}}} \sum_{x \in \hat{P}} \tilde{g}(x) - \frac{1}{n_{\hat{Q}}} \sum_{x \in \hat{Q}} \tilde{g}(x) && \text{(A.2)} \\
=& \max_{\tilde{g} \in H'} \frac{1}{n_{\hat{P}}} \sum_{x \in \hat{P}} \langle \tilde{g}, \psi(x) \rangle_{K'} - \frac{1}{n_{\hat{Q}}} \sum_{x \in \hat{Q}} \langle \tilde{g}, \psi(x) \rangle_{K'} && \text{(A.3)} \\
=& \max_{\tilde{g} \in H'} \langle \tilde{g}, \frac{1}{n_{\hat{P}}} \sum_{x \in \hat{P}} \psi(x) \rangle_{K'} - \langle \tilde{g}, \sum_{x \in \hat{Q}} \frac{1}{n_{\hat{Q}}} \psi(x) \rangle_{K'} \\
=& \max_{\tilde{g} \in H'} \langle \tilde{g}, \mu_{\hat{P}} - \mu_{\hat{Q}} \rangle_{K'} && \text{(A.4)} \\
=& \Lambda' ||\mu_{\hat{P}} - \mu_{\hat{Q}}||_{K'} && \text{(A.5)}
\end{aligned}
$$

In Equation A.1 we took $g(x) = L(h(x), f(x)) = (h(x) - f(x))^2$. In Equation A.2 we upper bound the expression of the unknown function g by the worst-case function $\tilde{g} \in H'$ — for this step to hold we have to assume $g \in H'$, otherwise the maximization over $\tilde{g}$ may not contain g. Note that $\tilde{g}(x) = \langle \tilde{g}, \psi(x) \rangle_{K'}$ due to the reproducing property [13, p. 96]. Equations A.3 to A.4 follow from the linearity of the inner product. In Equation A.4 we defined $\mu_{\hat{P}} = \frac{1}{n_{\hat{P}}} \sum_{x \in \hat{P}} \psi(x)$ and similarly for $\mu_{\hat{Q}}$, note that these are vectors in the RKHS of K'. The last step follows from the fact that the vector in H' maximizing the term in Equation A.4 is:

$$
\tilde{g} = \frac{\mu_{\hat{P}} - \mu_{\hat{Q}}}{||\mu_{\hat{P}} - \mu_{\hat{Q}}||_{K'}} \Lambda' \qquad \text{(A.6)}
$$

This follows from the fact that the inner product between two vectors is maximum if the vectors point in the same direction.

Because of the symmetry of $||\mu_{\hat{P}} - \mu_{\hat{Q}}||_{K'}$ with respect to $\hat{P}$ and $\hat{Q}$, it's straightforward to show that this derivation also holds if we switch $\hat{P}$ and $\hat{Q}$:

$$L_{\hat{Q}}(h, f) - L_{\hat{P}}(h, f) \leq \Lambda' ||\mu_{\hat{P}} - \mu_{\hat{Q}}||_{K'} \tag{A.7}$$

Combining Equation A.5 and A.7 we obtain:

$$|L_{\hat{P}}(h, f) - L_{\hat{Q}}(h, f)| \leq \Lambda' ||\mu_{\hat{P}} - \mu_{\hat{Q}}||_{K'}$$

This bound is in exactly the same form as the discrepancy bound in Section 2.2.2 and therefore these bounds can be directly compared. This comparison is given in Appendix C.

A.2. MMD Computation

We can compute the MMD quantity by working out the norm with kernel products:

$$\begin{aligned}\Lambda' ||\mu_{\hat{Q}} - \mu_{\hat{P}}|| =& \Lambda' \sqrt{\langle \mu_{\hat{Q}}, \mu_{\hat{Q}} \rangle_{K'} - 2\langle \mu_{\hat{P}}, \mu_{\hat{Q}} \rangle_{K'} + \langle \mu_{\hat{Q}}, \mu_{\hat{P}} \rangle_{K'}} \\ =& \Lambda' \sqrt{\frac{1}{n_{\hat{Q}}^2} \sum_{x,x' \in \hat{Q}} K'(x, x') - 2\frac{1}{n_{\hat{P}} n_{\hat{Q}}} \sum_{x \in \hat{P}} \sum_{x' \in \hat{Q}} K'(x, x') + \frac{1}{n_{\hat{P}}^2} \sum_{x,x' \in \hat{P}} K'(x, x')}\end{aligned} \tag{A.8}$$

We only compute $\tilde{g}$ for illustration purposes. To compute $\tilde{g}$ we simplify Equation A.6:

$$\tilde{g} = \frac{\mu_{\hat{P}} - \mu_{\hat{Q}}}{||\mu_{\hat{P}} - \mu_{\hat{Q}}||_{H'}} \Lambda \propto \mu_{\hat{P}} - \mu_{\hat{Q}}$$

Now we can compute $\tilde{g}(x)$ by working out the inner products using kernel matrices:

$$\tilde{g}(x) = \langle \tilde{g}, \psi(x) \rangle \propto \langle \mu_{\hat{P}} - \mu_{\hat{Q}}, \psi(x) \rangle = \frac{1}{n_{\hat{P}}} \sum_{x' \in \hat{P}} K(x, x') - \frac{1}{n_{\hat{Q}}} \sum_{x' \in \hat{Q}} K(x, x')$$

A.3. Agnostic MMD Generalization Bound

In this section we show how the MMD bound can be extended to the agnostic scenario, where $\tilde{g} \notin H'$. This bound will be similar to the discrepancy generalization bound, only here we approximate the true loss function g with the loss function $\tilde{g}$.

We aim to bound the quantity:

$$|L_{\hat{P}}(h, f) - L_{\hat{Q}}(h, f)|$$

In terms of $g(x)$ this is equal to:

$$|L_{\hat{P}}(h, f) - L_{\hat{Q}}(h, f)| = |\frac{1}{n_{\hat{P}}} \sum_{x \in \hat{P}} g(x) - \frac{1}{n_{\hat{Q}}} \sum_{x \in \hat{P}} g(x)|$$

To ease notation, we introduce the quantity $g_{\hat{Q}}$ as the empirical average of g on the set $\hat{Q}$:

$$g_{\hat{Q}} = \frac{1}{n_{\hat{Q}}} \sum_{x \in \hat{P}} g(x)$$

This notation can also be used to indicate the empirical average of g on the set $\hat{P}$, as $g_{\hat{P}}$. Furthermore, we also use this notation for the empirical average of $\tilde{g}$ on the sets $\hat{P}$ and $\hat{Q}$. In this notation, we aim to bound:

$$|L_{\hat{P}}(h, f) - L_{\hat{Q}}(h, f)| = |g_{\hat{P}} - g_{\hat{Q}}|$$

Observe that:

$$|g_{\hat{P}} - g_{\hat{Q}}| = |g_{\hat{P}} - g_{\hat{Q}} + \tilde{g}_{\hat{P}} - \tilde{g}_{\hat{P}} + \tilde{g}_{\hat{Q}} - \tilde{g}_{\hat{Q}}|$$

Where $\tilde{g}$ can be any function in H'. By reordering the terms and applying the triangle inequality we can show that:

$$|g_{\hat{P}} - g_{\hat{Q}}| \leq |\tilde{g}_{\hat{P}} - \tilde{g}_{\hat{Q}}| + |g_{\hat{P}} - g_{\hat{Q}} - \tilde{g}_{\hat{P}} + \tilde{g}_{\hat{Q}}|$$

Now we can bound the first term using the MMD, since we are sure that the loss function that is to be bounded in this case is in the set H' (see also the bound in Appendix A.1):

$$|\tilde{g}_{\hat{P}} - \tilde{g}_{\hat{Q}}| \leq \max_{\tilde{g} \in H'} |\tilde{g}_{\hat{P}} - \tilde{g}_{\hat{Q}}| = \text{MMD}(\hat{P}, \hat{Q})$$

Thus we obtain:

$$|g_{\hat{P}} - g_{\hat{Q}}| \leq \text{MMD}(\hat{P}, \hat{Q}) + |g_{\hat{P}} - g_{\hat{Q}} - \tilde{g}_{\hat{P}} + \tilde{g}_{\hat{Q}}|$$

Now this equation holds for any $\tilde{g} \in H'$. To make the bound tight, we can minimize the right hand side over all $\tilde{g} \in H'$:

$$|g_{\hat{P}} - g_{\hat{Q}}| \leq \text{MMD}(\hat{P}, \hat{Q}) + \min_{\tilde{g} \in H'} |g_{\hat{P}} - g_{\hat{Q}} - \tilde{g}_{\hat{P}} + \tilde{g}_{\hat{Q}}|$$

The term on the right hand side now plays the role of the approximation error of approximating g by $\tilde{g}$. Note that this term is easy to compute, since this can be rewritten as a convex minimization problem (if the labels of $\hat{Q}$ and $\hat{P}$ are known), unlike the discrepancy approximation term which involves a non-convex optimization problem. To make this term more comparable with the discrepancy approximation term, we can rewrite this term using the triangle inequality as:

$$|g_{\hat{P}} - g_{\hat{Q}} - \tilde{g}_{\hat{P}} + \tilde{g}_{\hat{Q}}| \leq |g_{\hat{P}} - \tilde{g}_{\hat{P}}| + |g_{\hat{Q}} - \tilde{g}_{\hat{Q}}|$$

Plugging in the definitions of $g_{\hat{P}}$, $g_{\hat{Q}}$, etc... we have that:

$$|g_{\hat{P}} - g_{\hat{Q}} - \tilde{g}_{\hat{P}} + \tilde{g}_{\hat{Q}}| \leq |\frac{1}{n_{\hat{P}}} \sum_{x \in \hat{P}} g(x) - \frac{1}{n_{\hat{P}}} \sum_{x \in \hat{P}} \tilde{g}(x)| + |\frac{1}{n_{\hat{Q}}} \sum_{x \in \hat{Q}} g(x) - \frac{1}{n_{\hat{Q}}} \sum_{x \in \hat{Q}} \tilde{g}(x)|$$

Using the triangle inequality we can show that:

$$|g_{\hat{P}} - g_{\hat{Q}} - \tilde{g}_{\hat{P}} + \tilde{g}_{\hat{Q}}| \leq \frac{1}{n_{\hat{P}}} \sum_{x \in \hat{P}} |g(x) - \tilde{g}(x)| + \frac{1}{n_{\hat{Q}}} \sum_{x \in \hat{Q}} |g(x) - \tilde{g}(x)|$$

Thus we find the generalization bound:

$$|L_{\hat{P}}(h, f) - L_{\hat{Q}}(h, f)| \leq \text{MMD}(\hat{P}, \hat{Q}) + \min_{\tilde{g} \in H'} \left(\frac{1}{n_{\hat{P}}} \sum_{x \in \hat{P}} |g(x) - \tilde{g}(x)| + \frac{1}{n_{\hat{Q}}} \sum_{x \in \hat{Q}} |g(x) - \tilde{g}(x)| \right)$$

B

Discrepancy

B.1. Computation of the Discrepancy

In this section we calculate the discrepancy analytically for the L_2 loss in the linear kernel (following the derivation of [12] page 8). At the end of this section we extend the computation to any arbitrary kernel (following the derivation of [4] Section 5.2). Note that all vectors are column vectors, and X is a data matrix where each row vector is an object and is thus of size $n \times d$. We consider the setting where $\hat{Q} \in \hat{P}$ and where all samples have equal weights, therefore this derivation is slightly adapted from [12] and [4]. We first rewrite the discrepancy for the linear kernel:

$$\begin{aligned}
\text{disc}(\hat{P}, \hat{Q}) &= \max_{h,h' \in H} |L_{\hat{P}}(h', h) - L_{\hat{Q}}(h', h)| \\
&= \max_{h,h' \in H} |\frac{1}{n_P} \sum_{x \in \hat{P}} (h'(x) - h(x))^2 - \frac{1}{n_{\hat{Q}}} \sum_{x \in \hat{Q}} (h'(x) - h(x))^2| \\
&= \max_{||w|| \leq \Lambda, ||w'|| \leq \Lambda} |\frac{1}{n_{\hat{P}}} \sum_{x \in \hat{P}} ((w' - w)^T x)^2 - \frac{1}{n_{\hat{Q}}} \sum_{x \in \hat{Q}} ((w' - w)^T x)^2| \\
&= \max_{||u|| \leq 2\Lambda} |\frac{1}{n_{\hat{P}}} \sum_{x \in \hat{P}} (u^T x)^2 - \frac{1}{n_{\hat{Q}}} \sum_{x \in \hat{Q}} (u^T x)^2| \qquad \text{(B.1)}
\end{aligned}$$

So note we can compute the discrepancy for the squared loss using the following formula:

$$\text{disc}(\hat{P}, \hat{Q}) = \max_{\tilde{z} \in \mathcal{H}, ||\tilde{z}||_K \leq 2\Lambda} |\frac{1}{n_{\hat{P}}} \sum_{x \in \hat{P}} \tilde{z}(x)^2 - \frac{1}{n_{\hat{Q}}} \sum_{x \in \hat{Q}} \tilde{z}(x)^2|$$

Where we took $\tilde{z} = u^T x$. This generalization to any arbitrary kernel holds since we could do this derivation in the RKHS of K, and use $\tilde{z} = \langle \psi(x), u \rangle^2$. We can call the worst-case loss function $\tilde{g}(x) = \tilde{z}(x)^2$. Then we can write the discrepancy in the same form as the MMD quantity in Equation A.2:

$$\text{disc}(\hat{P}, \hat{Q}) = |\frac{1}{n_{\hat{P}}} \sum_{x \in \hat{P}} \tilde{g}(x) - \frac{1}{n_{\hat{Q}}} \sum_{x \in \hat{Q}} \tilde{g}(x)|$$

Only for the discrepancy we have the constraint that $\tilde{g}(x) = \tilde{z}(x)^2$, and the maximization is over $\tilde{z}$ and not over $\tilde{g}$. See also Equation A.2.

We rewrite Equation B.1 further to compute the discrepancy:

$$\begin{aligned}\text{disc}(\hat{P},\hat{Q}) &= \max_{||u||\leq 2\Lambda} |\frac{1}{n_{\hat{P}}}\sum_{x\in\hat{P}} u^T x x^T u - \frac{1}{n_{\hat{Q}}}\sum_{x\in\hat{Q}} u^T x x^T u| \\ &= \max_{||u||\leq 2\Lambda} |\frac{1}{n_{\hat{P}}} u^T X_{\hat{P}}^T X_{\hat{P}} u - \frac{1}{n_{\hat{Q}}} u^T X_{\hat{Q}}^T X_{\hat{Q}} u| \\ &= \max_{||u||\leq 2\Lambda} |u^T M u|\end{aligned}$$

Where we defined the $d \times d$ matrix M as:

$$M = \frac{1}{n_{\hat{P}}} X_{\hat{P}}^T X_{\hat{P}} - \frac{1}{n_{\hat{Q}}} X_{\hat{Q}}^T X_{\hat{Q}}$$

The absolute value results in two terms we need to maximize over:

$$\text{disc}(\hat{P},\hat{Q}) = \max\left(\max_{||u||\leq 2\Lambda} u^T M u, \max_{||u||\leq 2\Lambda} u^T(-M)u\right) \tag{B.2}$$

We first compute the first term. Since M is a real symmetric matrix, M is a normal matrix and admits an orthonormal eigendecomposition with real eigenvalues. We can solve this maximization by using this eigendecomposition of M:

$$\max_{||u||\leq 2\Lambda} u^T M u = \max_{||u||\leq 2\Lambda} u^T \sum_i e_i \lambda_i e_i^T u \tag{B.3}$$

Here e_i are the eigenvectors of M and λ_i are the corresponding eigenvalues. Since M is a normal matrix its orthonormal eigenvectors span the entire space $\mathbb{R}^d$, and thus form an orthonormal basis for $\mathbb{R}^d$. Because of this we can express the vector u in terms of the eigenvectors of M:

$$u = \sum_i^d u_i e_i$$

We can use this to solve the maximization:

$$\begin{aligned}\max_{||u||\leq 2\Lambda} u^T M u &= \max_{||u||\leq 2\Lambda} \sum_i^d u_i e_i^T \sum_i e_i \lambda_i e_i^T \sum_i^d u_i e_i \\ &= \max_{||u||\leq 2\Lambda} \sum_i^d u_i e_i^T e_i \lambda_i e_i^T u_i e_i \\ &= \max_{||u||\leq 2\Lambda} \sum_i^d u_i^2 \lambda_i\end{aligned}$$

Where we used orthonormality of the eigenvectors of M. Observe that the components of u each weight an eigenvalue of M. To maximize this quantity, we thus need to weight the maximal eigenvalue maximally. Thus the vector u that maximizes this is given by a multiple of the eigenvector $e_{\max}$ corresponding to the maximum eigenvalue $\lambda_{\max}$:

$$u = e_{max} 2\Lambda$$

Note $||u|| = 2\Lambda$ (since the eigenvector e_{max} is orthonormal) to maximize the quantity in Equation B.3. Substituting the solution of u and using that the eigendecomposition is orthogonal we obtain:

$$\begin{aligned}\max_{||u||\leq 2\Lambda} u^T M u &= 4\Lambda^2 e_{\max}^T \sum_i e_i \lambda_i e_i^T e_{\max} \\ &= 4\Lambda^2 e_{\max}^T e_{\max} \lambda_{\max} e_{\max}^T e_{\max} \\ &= 4\Lambda^2 \lambda_{\max}\end{aligned}$$

Now for the maximization of the second term of Equation B.2, we can use that the matrix $(-M)$ has the same eigenvalues as M only with the sign reversed, so we obtain the same solution except with the smallest eigenvalue $\lambda_{\min}$ and a change of sign. Thus we find that the discrepancy is given by:

$$\begin{aligned}\text{disc}(\hat{P},\hat{Q}) &= 4\Lambda^2 \max(\lambda_{\max}, -\lambda_{\min}) \\ &= 4\Lambda^2 \max_i |\lambda_i| = 4\Lambda^2\lambda^* \\ &= 4\Lambda^2 ||M||_2\end{aligned}$$

Where $||M||_2$ is also known as the spectral norm of the matrix M, which is given by the largest absolute eigenvalue λ^*.

Now we can compute the discrepancy in a linear kernel. In an arbitrary kernel we cannot easily compute the covariance matrices of the sets $\hat{P}$ and $\hat{Q}$, since the RKHS of K may be very large or infinite. In the following we rewrite the spectral norm of M in terms of kernel innerproducts, so the discrepancy can be computed in any arbitrary kernel.

First we introduce the set $\hat{U} = \hat{P} - \hat{Q}$. We assume in the following that the matrix $X_{\hat{P}}$ is structured as:

$$X_{\hat{P}} = \begin{bmatrix} X_{\hat{Q}} \\ X_{\hat{U}} \end{bmatrix}$$

It can be shown that M can be rewritten as[4]:

$$M = X_{\hat{P}}^T D X_{\hat{P}}$$

Where D is an $n_{\hat{P}} \times n_{\hat{P}}$ diagonal matrix. The matrix D reweights all objects and is given by:

$$D = \begin{bmatrix} (\frac{1}{n_{\hat{P}}} - \frac{1}{n_{\hat{Q}}})I & 0 \\ 0 & \frac{1}{n_{\hat{P}}} I \end{bmatrix}$$

Where $(\frac{1}{n_{\hat{P}}} - \frac{1}{n_{\hat{Q}}})I$ is a diagonal matrix of size $n_{\hat{Q}} \times n_{\hat{Q}}$, and $\frac{1}{n_{\hat{P}}} I$ is a diagonal matrix of size $n_{\hat{U}} \times n_{\hat{U}}$.

Since the matrix product AB and BA have the same eigenvalues[4], and since $||M||_2$ only depends on the eigenvalues, we can permute the matrices in M to obtain a new matrix M_K while $||M||_2 = ||M_K||_2$:

$$\begin{aligned} M &= (X_{\hat{P}}^T D) X_{\hat{P}} \\ M_K &= X_{\hat{P}} (X_{\hat{P}}^T D) = K_{\hat{P}\hat{P}} D \end{aligned} \tag{B.4}$$

Now M_K only depends on the kernel matrix of $\hat{P}$. Note that the kernel matrix should be ordered the same as $X_{\hat{P}}$, thus the kernel matrix is given by:

$$K_{\hat{P}} = \begin{bmatrix} K_{\hat{Q}\hat{Q}} & K_{\hat{Q}\hat{U}} \\ K_{\hat{U}\hat{Q}} & K_{\hat{U}\hat{U}} \end{bmatrix}$$

Now the discrepancy can be computed in any arbitrary kernel using:

$$\text{disc}(\hat{P},\hat{Q}) = 4\Lambda^2 ||M_K||_2$$

Now we discuss how to compute $\tilde{g}$. The computation of the function $\tilde{g}$ is straightforward in the linear kernel: take the eigenvector $\tilde{u}$ that maximizes Equation B.2. Then by definition $\tilde{g}(x) = (\tilde{u}^T x)^2$.

The works concerning the discrepancy do not provide a way to compute $\tilde{g}$ for an arbitrary kernel, here we describe a method to do so. To obtain $\tilde{g}(x)$ we require the the values of $\tilde{z}(x)$. From here on, we write $\tilde{z}$ as a vector indicating the outputs of $\tilde{z}$ on all objects in $\hat{P}$. In the linear kernel we have that:

$$\tilde{z} = X_{\hat{P}} \tilde{u}$$

Note that $\tilde{u}$ is the eigenvector corresponding to the largest absolute eigenvalue λ^* of M. Now we show that the vector $\tilde{z}$ is the eigenvector of M_K corresponding to the largest absolute eigenvalue λ^* of M_K:

$$M_K\tilde{z} = M_K X_{\hat{P}}\tilde{u} = X_{\hat{P}} X_{\hat{P}}^T D X_{\hat{P}}\tilde{u} = X_{\hat{P}} M\tilde{u} = \lambda^* X_{\hat{P}}\tilde{u} = \lambda^*\tilde{z}$$

This shows that we can compute $\tilde{z}$ by computing the eigenvector corresponding to the largest absolute eigenvalue λ^* of M_K. Since M_K can be computed in any arbitrary kernel, we can now compute $\tilde{z}$ in any arbitrary kernel. Then $\tilde{g}(x)$ is simply given by $\tilde{g}(x) = \tilde{z}(x)^2$. However observe that unlike the MMD where it is straightforward to compute $\tilde{g}(x)$ for any $x \in \mathcal{X}$, for the discrepancy we can only compute $\tilde{g}(x)$ for objects $x \in \hat{P}$.

B.2. Proof of the Agnostic Discrepancy Bound

In this section we show how to derive the agnostic discrepancy generalization bound:

$$|L_{\hat{P}}(h,f) - L_{\hat{Q}}(h,f)| \leq \text{disc}(\hat{P},\hat{Q}) + 2C \min_{\tilde{f}\in H}\left(\frac{1}{n_{\hat{P}}}\sum_{x\in\hat{P}}|f(x)-\tilde{f}(x)| + \frac{1}{n_{\hat{Q}}}\sum_{x\in\hat{Q}}|f(x)-\tilde{f}(x)|\right)$$

Which holds for any deterministic labeling function $f(x)$ and any $h \in H$. We assume that for any h and f that: $L(h(x), f(x)) \leq C$. This proof is slightly adapted from the proof in [15]. Recall that the discrepancy term already gives a bound on this quantity in the realizable case $f \in H$:

$$|L_{\hat{P}}(h,f) - L_{\hat{Q}}(h,f)| \leq \max_{h,f\in H}|L_{\hat{P}}(h,f) - L_{\hat{Q}}(h,f)| = \text{disc}(\hat{P},\hat{Q})$$

To extend this bound to the agnostic case this will require us to approximate the agnostic function f by a function $\tilde{f} \in H$. We give the proof below.

Observe that the following equation holds for all $\tilde{f} \in H$:

$$|L_{\hat{P}}(h,f) - L_{\hat{Q}}(h,f)| = |L_{\hat{P}}(h,f) - L_{\hat{Q}}(h,f) + L_{\hat{P}}(h,\tilde{f}) - L_{\hat{P}}(h,\tilde{f}) - L_{\hat{Q}}(h,\tilde{f}) + L_{\hat{Q}}(h,\tilde{f})|$$

If we rearrange the terms on the right hand side and apply the triangle inequality, we can show that:

$$|L_{\hat{P}}(h,f) - L_{\hat{Q}}(h,f)| \leq |L_{\hat{P}}(h,\tilde{f}) - L_{\hat{Q}}(h,\tilde{f})| + |L_{\hat{P}}(h,f) - L_{\hat{P}}(h,\tilde{f}) + L_{\hat{Q}}(h,\tilde{f}) - L_{\hat{Q}}(h,f)|$$

Now we can bound the first term on the right hand side by maximizing over all h and $\tilde{f} \in H$:

$$|L_{\hat{P}}(h,\tilde{f}) - L_{\hat{Q}}(h,\tilde{f})| \leq \max_{h,\tilde{f}\in H}|L_{\hat{P}}(h,\tilde{f}) - L_{\hat{Q}}(h,\tilde{f})| = \text{disc}(\hat{P},\hat{Q})$$

Then we obtain the following generalization bound which holds for all $\tilde{f} \in H$:

$$|L_{\hat{P}}(h,f) - L_{\hat{Q}}(h,f)| \leq \text{disc}(\hat{P},\hat{Q}) + |L_{\hat{P}}(h,f) - L_{\hat{P}}(h,\tilde{f}) + L_{\hat{Q}}(h,\tilde{f}) - L_{\hat{Q}}(h,f)|$$

Since this bound holds for all $\tilde{f} \in H$, so we can choose to minimize with respect to $\tilde{f}$ to get a bound that is as tight as possible:

$$|L_{\hat{P}}(h,f) - L_{\hat{Q}}(h,f)| \leq \text{disc}(\hat{P},\hat{Q}) + \min_{\tilde{f}\in H}|L_{\hat{P}}(h,f) - L_{\hat{P}}(h,\tilde{f}) + L_{\hat{Q}}(h,\tilde{f}) - L_{\hat{Q}}(h,f)| \quad \text{(B.5)}$$

Observe that the second term on the right hand side plays the role of an approximation error: here the function $\tilde{f}$ approximates the function f on the sets $\hat{Q}$ and $\hat{P}$. The better $\tilde{f}$ can approximate f, the tighter the bound. However, this term is difficult to compute, since it is a non-convex optimization problem in $\tilde{f}$ due to the absolute value. Therefore, in [15] they bound this quantity as follows. Observe that due to the triangle inequality:

$$|L_{\hat{P}}(h,f) - L_{\hat{P}}(h,\tilde{f}) + L_{\hat{Q}}(h,\tilde{f}) - L_{\hat{Q}}(h,f)| \leq |L_{\hat{P}}(h,f) - L_{\hat{P}}(h,\tilde{f})| + |L_{\hat{Q}}(h,\tilde{f}) - L_{\hat{Q}}(h,f)| \quad \text{(B.6)}$$

The two terms on the right hand side of this equation are still non-convex in terms of $\tilde{f}$. Now we can use the μ-admissibility of the squared loss to bound both terms on the right in terms that are convex in $\tilde{f}$. For the squared loss we have that $\mu = 2C$, where $L(h(x), f(x)) \leq C$ for all $x \in \mathcal{X}$ and $h \in H$, see appendix C of [15]. Then by the μ-admissibility of the squared loss we have (see also [15]):

$$|L_{\hat{P}}(h, f) - L_{\hat{P}}(h, \tilde{f})| \leq 2C \frac{1}{n_{\hat{P}}} \sum_{x \in \hat{P}} |f(x) - \tilde{f}(x)| \tag{B.7}$$

And:

$$|L_{\hat{Q}}(h, f) - L_{\hat{Q}}(h, \tilde{f})| \leq 2C \frac{1}{n_{\hat{Q}}} \sum_{x \in \hat{Q}} |f(x) - \tilde{f}(x)| \tag{B.8}$$

Combining Equation B.6 with Equation B.7 and Equation B.8 we find that:

$$|L_{\hat{P}}(h, f) - L_{\hat{P}}(h, \tilde{f}) + L_{\hat{Q}}(h, \tilde{f}) - L_{\hat{Q}}(h, f)| \leq 2C \left(\frac{1}{n_{\hat{P}}} \sum_{x \in \hat{P}} |f(x) - \tilde{f}(x)| + \frac{1}{n_{\hat{Q}}} \sum_{x \in \hat{Q}} |f(x) - \tilde{f}(x)| \right)$$

Combining with Equation B.5 we find our final result:

$$|L_{\hat{P}}(h, f) - L_{\hat{Q}}(h, f)| \leq \text{disc}(\hat{P}, \hat{Q}) + 2C \min_{\tilde{f} \in H} \left(\frac{1}{n_{\hat{P}}} \sum_{x \in \hat{P}} |f(x) - \tilde{f}(x)| + \frac{1}{n_{\hat{Q}}} \sum_{x \in \hat{Q}} |f(x) - \tilde{f}(x)| \right)$$

Introducing the approximation error $\eta_{\text{disc}}(\hat{P}, \hat{Q}, f)$:

$$\eta_{\text{disc}}(\hat{P}, \hat{Q}, f) = \min_{\tilde{f} \in H} \left(\frac{1}{n_{\hat{P}}} \sum_{x \in \hat{P}} |\tilde{f}(x) - f(x)| + \frac{1}{n_{\hat{Q}}} \sum_{x \in \hat{Q}} |\tilde{f}(x) - f(x)| \right)$$

We find that:

$$|L_{\hat{P}}(h, f) - L_{\hat{Q}}(h, f)| \leq \text{disc}(\hat{P}, \hat{Q}) + 2C\eta_{\text{disc}}(\hat{P}, \hat{Q}, f)$$

B.3. Why Did We Consider this Discrepancy Generalization Bound

Multiple bounds are given in [4] to motivate the use of the discrepancy. These bounds differ from other bounds in domain adaptation. In domain adaptation the bounds are typically derived by considering the density ratios of Q and P. The bounds in [4] compare the performance of the model trained on $\hat{Q}$ compared with the model trained on $\hat{P}$. These bounds are quite novel, but actually can be derived for the MMD as well if the MMD is adapted to take into account the hypothesis set (see Subsection 2.2.3). They thus do not contribute truly to explaining why the discrepancy is a more desirable bound than the MMD.

Furthermore, we found that typically these bounds from [4] in terms of the model trained on $\hat{P}$ are looser than a simpler bound from [15]. Only in case $\lambda > 1$ or if the discrepancy is very large the bounds of [4] may be tighter. We computed both bounds in all our experiments and found that in all cases the bounds of [15] were tighter, since in our experiments we typically set $\lambda < 1$. For more details see appendix B.4. Therefore, we instead study the bound from [15] that is easier to derive and understand. This bound is also directly comparable with the MMD bound derived in Subsection 2.1.2.

B.4. Discrepancy Bounds in Terms of the Oracle Hypothesis

In [4] several bounds are given in terms of the loss when the hypothesis trained on $\hat{Q}$ is compared to the oracle hypothesis, meaning the hypothesis that is obtained when training on $\hat{P}$. In this section we show that these bounds are typically weaker then the bound given in Section 2.2.2. For example consider the bound:

Theorem 8 (General pointwise bound for the squared loss[4]) *Let L be the squared loss and assume that for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$, $L(h(x), y) \leq C$ and $K(x, x) \leq r^2$ for some $C > 0$ and $r > 0$. Let h' be the hypothesis returned by kernel ridge regression when minimizing $F_{(\hat{P},f)}$ and h the one when minimizing $F_{(\hat{Q},f)}$. Then for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$,*

$$|L(h'(x), y) - L(h(x), y)| \leq \frac{r\sqrt{C}}{\lambda}(\delta_H(f, \hat{Q}, \hat{P}) + \sqrt{\delta_H(f, \hat{Q}, \hat{P})^2 + 4\lambda disc(\hat{P}, \hat{Q})}) \tag{B.9}$$

Where $\delta_H(f_P, f_Q)$ is the weighted feature discrepancy given by:

$$\delta_H(f, \hat{Q}, \hat{P}) = \inf_{h \in H} ||\frac{1}{n_{\hat{P}}} \sum_{x \in \hat{P}} [(h(x) - f(x))\psi_K(x)] - \frac{1}{n_{\hat{Q}}} \sum_{x \in \hat{Q}} [(h(x) - f(x))\psi_K(x)]||_K$$

The weighted feature discrepancy is similar to the term $\eta_{\text{disc}}(f, \hat{Q}, \hat{P})$ we have seen earlier in the bound in Section 2.2.2, it measures the approximation error we make when we approximate f by a hypothesis $h \in H$. In the realizable setting $\eta_{\text{disc}}(f, \hat{P}, \hat{Q}) = 0$ and $\delta_H(f_P, f_Q) = 0$.

This bound is derived as follows. It can be shown that:

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

Where $\mu = 2C$. The last term can be bounded by the right hand side of Equation B.9. See [4] for the full derivation. Note that this bound thus aims to minimize the distance to the oracle classifier h' in the RKHS of the kernel K.

Now we compare this bound to the bound in Section 2.2.2. For the realizable case the bound is given by:

$$|L_{\hat{P}}(h'(x), f) - L_{\hat{P}}(h(x), f)| \leq \frac{r\sqrt{C}}{\lambda}\sqrt{4\lambda \text{disc}(\hat{P}, \hat{Q})}$$

We consider the Gaussian kernel where $r = 1$, we use $\Lambda = \sqrt{\frac{1}{\lambda}}$ and $C = (\Lambda r + 1)^2$ as derived in appendix F. After rewriting we can show that:

$$L_{\hat{P}}(h(x), y) \leq L_{\hat{P}}(h'(x), y) + (\Lambda + 1)\Lambda^2 4\sqrt{||M||_2}$$

Thus the last term in this bound is in the order of Λ^3. Since λ is typically very small, $\Lambda = \frac{1}{\sqrt{\lambda}} >> 1$, and therefore this bound is typically weaker than the bound derived in Section 2.2.2 which has a term that is in the order of Λ^2. The bound here can be tighter if $||M||_2$ is very large, since it is in the square root in this equation while it is not in a square root in the bound of Section 2.2.2, or if $\Lambda < 1$. Note that this bound compares the performance of h and h' on the set $\hat{P}$ and is therefore a different kind of bound than the one derived in Section 2.2.2. We computed both bounds explicitly and found that the bound of Section 2.2.2 was typically tighter in our experiments, since in our experiments $\Lambda >> 1$.

The bound of Theorem 2 in [4] can be shown to be even weaker, it has a dependence of Λ^4. The bound in Theorem 4 of [4] is incomparable with the bound in Section 2.2.2 since it instead considers the root mean squared error instead of the mean squared error.

C

Comparison between Discrepancy and MMD

In this appendix we show that the bound of the discrepancy derived in Section 2.2.2 is always tighter or as tight as the bound of the MMD derived in Section 2.1 for the squared loss in the realizable setting where $f \in H$. This is our main theoretical result. To this end we also show how to choose the kernel K' and the corresponding set H' for the MMD to satisfy the assumptions of the bound in the realizable setting. This describes how to choose the kernel K' to take the hypothesis set of the learning algorithm into account for the MMD, and is our second important theoretical result.

First we review some notation and give a brief recap of the bounds of the discrepancy and the MMD and their assumptions. To keep matters simple we first only consider the linear kernel as the kernel for our learning algorithm. We show that choosing the kernel for the MMD as the squared kernel $K'(x, x') = K(x, x')^2$ guarantees the MMD bound to hold in the realizable setting. We give a small example in 2 dimensions to illustrate this. Then we show why the bound of the discrepancy in this setting is always tighter than the MMD bound for $d = 2$ dimensions, this is our first main result. Afterward we extend this to an arbitrary number of dimensions, and then we argue that this proof can be extended to any arbitrary kernel K: we show that K' is the squared kernel as well for any arbitrary kernel K. One may wonder whether the assumptions of the MMD are in some sense less restrictive, since the bound is looser. In the last subsection we show that the assumptions of the discrepancy and MMD are equivalent. In view of these results, one may conclude that the discrepancy bound is always preferable theoretically. Finally, we give a generalization bound for the MMD that always holds in the agnostic case as well as long as the function f deterministically determines the label.

C.1. Recap of Bounds and Notation

For simplicity we choose K as the linear kernel, thus $K(x, x') = x^T x'$. $h(x)$ is the evaluation of h on object x, in the linear kernel this is simply $h(x) = h^T x$ where h is in this case the vector describing the linear model. We consider the realizable setting where we assume $f \in H$, where H is the hypothesis set of our chosen model.

For the MMD we choose the set H' as $H' = \{\forall h \in \mathcal{H}' : ||h||_{K'} \leq \Lambda' = 4\Lambda^2\}$, where K' is the squared kernel of K, meaning $K'(x, x') = K(x, x')^2$. $\mathcal{H}'$ is the RKHS of the kernel K'. This choice of H' guarantees that the MMD bound of Theorem 1 will hold as will be shown later.

For a function $g \in H'$, $g(x)$ can be computed as: $g(x) = \langle g, \psi(x) \rangle_{K'}$. Here $\psi(x)$ is the featuremap of x which maps objects from the input space $\mathcal{X}$ to the RKHS $\mathcal{H}'$ of kernel K'. $\psi(x)$ and g are vectors in the RKHS $\mathcal{H}'$ of K'. Here we only use $\psi(x)$ to denote featuremaps of K' since the featuremap of K is given by the identity since for simplicity we use the linear kernel as hypothesis set and thus $\mathcal{X} = \mathcal{H}$.

The discrepancy is computed as the spectral norm of matrix M, which is the difference of the two covariance matrices between sets $\hat{P}$ and $\hat{Q}$:

$$M = \frac{1}{n_{\hat{P}}} X_{\hat{P}}^T X_{\hat{P}} - \frac{1}{n_{\hat{Q}}} X_{\hat{Q}}^T X_{\hat{Q}}$$

The discrepancy bound for the realizable setting is given by:

$$|L_{\hat{P}}(h,f) - L_{\hat{Q}}(h,f)| \leq 4\Lambda^2 ||M||_2 = 4\Lambda^2 \max_i |\lambda_i| = \text{disc}(\hat{P}, \hat{Q})$$

Where λ_i are the eigenvalues of the matrix M. Note in the realizable setting $\eta_{\text{disc}}(f, \hat{Q}, \hat{P}) = 0$ which is the reason the approximation term is not included in the bound above.

The MMD bound is given by:

$$|L_{\hat{P}}(h,f) - L_{\hat{Q}}(h,f)| \leq \Lambda' ||\mu_{\hat{P}} - \mu_{\hat{Q}}||_{K'} = \text{MMD}(\hat{P}, \hat{Q})$$

Recall that the MMD can be computed using:

$$\text{MMD}(\hat{P}, \hat{Q}) = \max_{\tilde{g} \in H'} \left(\frac{1}{n_{\hat{P}}} \sum_{x \in \hat{P}} \tilde{g}(x) - \frac{1}{n_{\hat{Q}}} \sum_{x \in \hat{Q}} \tilde{g}(x) \right)$$

The key assumption for the MMD bound to hold was that the loss function $g(x) = L(h(x), f(x)) = (h(x) - f(x))^2$ is contained in the set H'. Now we first determine the set H'.

C.2. Determining the Function Set for the MMD

In this section we show that the kernel K' should be chosen as the squared kernel of K in the realizable setting to satisfy the assumptions of the MMD bound that $g(x) = (h(x) - f(x))^2$ is contained in the set H'. We can find H' as follows. First we define the function $z(x) = h(x) - f(x)$. The function $||z||_K = ||h - f||_K \leq 2\Lambda$, since $||h||_K \leq \Lambda$ and $||f||_K \leq \Lambda$. Note that $g(x) = z(x)^2 = (h(x) - f(x))^2$. We define the squared kernel of K as $K'(x, x') = \langle x, x' \rangle_K^2 = (x^T x')^2 = K(x, x')^2$. The featuremap ψ of K' that maps from $\mathcal{X} = \mathcal{H}$ to $\mathcal{H}'$ is given by [26, chap. 9.1][1]:

$$\psi(x) = (x_1^2, \ldots, x_n^2, \sqrt{2}x_1x_2, \ldots, \sqrt{2}x_1x_n, \sqrt{2}x_2x_3, \ldots, \sqrt{2}x_2x_n, \ldots, \sqrt{2}x_{n-1}x_n) \tag{C.1}$$

Note the kernel K' is a PSD kernel since its featuremap exists. The function $z(x)$ can be described as $z(x) = \langle z, x \rangle_K = z^T x$. Thus the function $g(x) = z(x)^2 = \langle z, x \rangle_K^2 = K'(z, x) = \langle \psi(z), \psi(x) \rangle_{K'}$, thus $g \in H'$ with $g = \psi(z)$.

This is more complicated to show when K is any arbitrary kernel and we defer this discussion to Subsection C.6.

Furthermore we have that $||g||_{K'} = \langle \psi(z), \psi(z) \rangle_{K'} = K'(z, z) = \langle z, z \rangle_K^2 = ||z||_K^2 \leq 4\Lambda^2$, since $||z||_K \leq 2\Lambda$. Thus we have shown that $||g||_{K'} \leq 4\Lambda^2$. Thus in the realizable setting to satisfy the assumptions of the MMD we need to choose the set H' as $H' = \{\forall h \in \mathcal{H}' \mid ||h||_{K'} \leq \Lambda' = 4\Lambda^2\}$ in case we use a linear kernel K for the learning algorithm.

C.3. Illustrating Example

This example illustrates the derivation of the previous subsection and can be skipped if this is clear. For simplicity we take $d = 2$ dimensions. The function $z(x)$ is a linear function: $z(x) = f(x) - h(x) = z^T x = x_1 z_1 + x_2 z_2$. The norm of z is bounded by 2Λ:

$$||z||_K = \sqrt{z_1^2 + z_2^2} \leq 2\Lambda$$

[1] Note that actually in [26] this kernel is defined as a polynomial kernel. In our case for this polynomial kernel we have that $R = 0$ and $d = 2$, resulting in the featuremap given in Equation C.1. This is often referred to as the squared kernel.

Now if we square the function $z(x)$ we obtain the function $g(x)$:

$$g(x) = z(x)^2 = (x_1 z_1 + x_2 z_2)^2 = x_1^2 z_1^2 + \sqrt{2}\sqrt{2} x_1 x_2 z_1 z_2 + x_2^2 z_2^2$$

Note that for the squared kernel for $d = 2$ the featuremap is given by: $\psi(x) = (x_1^2, x_2^2, \sqrt{2}x_1 x_2) = (p_1, p_2, p_3)$. Here we have chosen $\{p_1, p_2, p_3\}$ as the basis for the RKHS of K'. We have that:

$$g(x) = z(x)^2 = z_1^2 p_1 + z_2^2 p_2 + \sqrt{2} z_1 z_2 p_3$$

We directly see that $\psi(z) = (z_1^2, z_2^2, \sqrt{2}z_1 z_2)_{H'}^T$ with respect to the basis of the RKHS of K', which was to be expected since this corresponds exactly with the mapping of ψ. Furthermore we observe $g(x) = \langle \psi(z), \psi(x) \rangle_{H'}$, thus $g \in \mathcal{H}'$. Now in the squared kernel K', the norm of $g(x)$ becomes:

$$||g||_{K'} = ||z^2||_{K'} = || \begin{pmatrix} z_1^2 \\ z_2^2 \\ \sqrt{2} z_1 z_2 \end{pmatrix} || = \sqrt{z_1^4 + 2 z_1^2 z_2^2 + z_2^4} = \sqrt{(z_1^2 + z_2^2)^2} = (z_1^2 + z_2^2) \leq 4\Lambda^2$$

We can show this as well using:

$$||z^2||_{K'} = \langle \psi(z), \psi(z) \rangle_{K'} = (z^T z)^2 = ||z||_K^2 \leq 4\Lambda^2$$

Thus we find that indeed $\Lambda' = 4\Lambda^2$.

C.4. Proof that the Discrepancy Bound is Tighter (Main Result)

Now we will relate the discrepancy to the MMD for $d = 2$ dimensions. The goal of this section is to show that we can compute the MMD with the matrix M as follows:

$$\text{MMD}(\hat{P}, \hat{Q}) = 4\Lambda^2 ||\mu_{\hat{P}} - \mu_{\hat{Q}}||_{K'} = 4\Lambda^2 ||M||_F$$

We first will show this result. In this form the discrepancy and MMD bounds become comparable. At the end of this section we compare both bounds and show the discrepancy bound is always tighter or as tight as the MMD bound.

Let us introduce some notation. The jth component of object i will be denoted by x_{ij}. The covariance matrix for set $\hat{P}$ becomes:

$$\frac{1}{n_{\hat{P}}} X_{\hat{P}}^T X_{\hat{P}} = \frac{1}{n_{\hat{P}}} \begin{pmatrix} \sum_{i\in\hat{P}} x_{1i}^2 & \sum_{i\in\hat{P}} x_{1i}x_{2i} \\ \sum_{i\in\hat{P}} x_{1i}x_{2i} & \sum_{i\in\hat{P}} x_{2i}^2 \end{pmatrix}$$

The matrix M is given by:

$$\begin{aligned} M =& \frac{1}{n_{\hat{P}}} X_{\hat{P}}^T X_{\hat{P}} - \frac{1}{n_{\hat{Q}}} X_{\hat{Q}}^T X_{\hat{Q}} \\ =& \begin{pmatrix} \frac{1}{n_{\hat{P}}} \sum_{i\in\hat{P}} x_{1i}^2 - \frac{1}{n_{\hat{Q}}} \sum_{i\in\hat{Q}} x_{1i}^2 & \frac{1}{n_{\hat{P}}} \sum_{i\in\hat{P}} x_{1i}x_{2i} - \frac{1}{n_{\hat{Q}}} \sum_{i\in\hat{Q}} x_{1i}x_{2i} \\ \frac{1}{n_{\hat{P}}} \sum_{i\in\hat{P}} x_{1i}x_{2i} - \frac{1}{n_{\hat{Q}}} \sum_{i\in\hat{Q}} x_{1i}x_{2i} & \frac{1}{n_{\hat{P}}} \sum_{i\in\hat{P}} x_{2i}^2 - \frac{1}{n_{\hat{Q}}} \sum_{i\in\hat{Q}} x_{2i}^2 \end{pmatrix} \\ =& \begin{pmatrix} \Delta_{11} & \Delta_{12} \\ \Delta_{21} & \Delta_{22} \end{pmatrix} \end{aligned} \tag{C.2}$$

We define Δ_{kl} to ease notation:

$$\Delta_{kl} \equiv \frac{1}{n_{\hat{P}}} \sum_{i\in\hat{P}} x_{ki}x_{li} - \frac{1}{n_{\hat{Q}}} \sum_{i\in\hat{Q}} x_{ki}x_{li}$$

Note that $\Delta_{kl} = \Delta_{lk}$. Recall that the MMD is given by the norm of the difference of the means of the sets $\hat{P}$ and $\hat{Q}$ in the RKHS of the kernel K'. Earlier we have shown that K' is the squared kernel of K. Now we explicitly calculate these means. The mean of $\hat{P}$ in the RKHS of K' is given by:

$$\mu_{\hat{P}} = \frac{1}{n_{\hat{P}}} \begin{pmatrix} \sum_{i\in\hat{P}} x_{1i}^2 \\ \sum_{i\in\hat{P}} x_{2i}^2 \\ \sum_{i\in\hat{P}} \sqrt{2}x_{1i}x_{2i} \end{pmatrix}$$

Note that this is in 3 dimensions, where we obtained these x values by using the featuremap of the quadratic kernel: $\psi(x) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)$. The difference between the means $\mu_P - \mu_Q$ can be written as:

$$\mu_{\hat{P}} - \mu_{\hat{Q}} = \begin{pmatrix} \frac{1}{n_{\hat{P}}} \sum_{i\in\hat{P}} x_{1i}^2 - \frac{1}{n_{\hat{Q}}} \sum_{i\in\hat{Q}} x_{1i}^2 \\ \frac{1}{n_{\hat{P}}} \sum_{i\in\hat{P}} x_{2i}^2 - \frac{1}{n_{\hat{Q}}} \sum_{i\in\hat{P}} x_{2i}^2 \\ \sqrt{2}(\frac{1}{n_{\hat{P}}} \sum_{i\in\hat{P}} x_{1i}x_{2i} - \frac{1}{n_{\hat{Q}}} \sum_{i\in\hat{Q}} x_{1i}x_{2i}) \end{pmatrix} = \begin{pmatrix} \Delta_{11} \\ \Delta_{22} \\ \sqrt{2}\Delta_{12} \end{pmatrix}$$

The norm of $\mu_{\hat{P}} - \mu_{\hat{Q}}$ is given by:

$$||\mu_{\hat{P}} - \mu_{\hat{Q}}||_{K'}^2 = \Delta_{11}^2 + \Delta_{22}^2 + 2\Delta_{12}^2$$

Note that the Frobenius norm of M is given by the same expression (see Equation C.2, and note that $\Delta_{12} = \Delta_{21}$):

$$||M||_F^2 = \Delta_{11}^2 + 2\Delta_{12}^2 + \Delta_{22}^2 = ||\mu_{\hat{P}} - \mu_{\hat{Q}}||_{K'}^2$$

Thus the MMD bound can be written as:

$$\text{MMD}(\hat{P}, \hat{Q}) = 4\Lambda^2 ||\mu_{\hat{P}} - \mu_{\hat{Q}}||_{K'} = 4\Lambda^2 ||M||_F$$

Which was what we set out to show. Now we compare the discrepancy and MMD bound, but first we slightly rewrite the MMD quantity.

It can be shown that the Frobenius norm is equal to the square root of the sum of squared singular values. However, since M is a real symmetric matrix, the singular values σ_i of M are equal (except for the sign) to the eigenvalues λ_i of M, and therefore here the Frobenius norm is equal to the square root of the sum of squared eigenvalues of M:

$$\text{MMD}(\hat{P}, \hat{Q}) = 4\Lambda^2 ||M||_F = 4\Lambda^2 \sqrt{\sum_i \sigma_i^2} = 4\Lambda^2 \sqrt{\sum_i \lambda_i^2}$$

The bound for the discrepancy was given by:

$$\text{disc}(\hat{P}, \hat{Q}) = 4\Lambda^2 ||M||_2 = 4\Lambda^2 \max_i(|\lambda_i|)$$

By comparing both expressions, we see that the discrepancy bound is always tighter or as tight as the MMD in this setting:

$$\text{disc}(\hat{P}, \hat{Q}) = 4\Lambda^2 ||M||_2 \leq 4\Lambda^2 ||M||_F = \text{MMD}(\hat{P}, \hat{Q})$$

C.5. Extension to Arbitrary Number of Dimensions

This can be generalized to any arbitrary number of dimensions as follows. In d dimensions, the entries of the matrix M become $M_{kl} = \Delta_{kl}$. Therefore the Frobenius norm of M is given by:

$$||M||_F^2 = \sum_{i=1\ldots d} \sum_{j=1\ldots d} \Delta_{ij}^2 = \sum_{i=1\ldots d} \Delta_{ii}^2 + \sum_{j=i+1,\ldots,d} 2\Delta_{ij}^2$$

Correspondingly, the vector $\mu_{\hat{P}} - \mu_{\hat{Q}}$ becomes:

$$\mu_{\hat{P}} - \mu_{\hat{Q}} = \begin{pmatrix} \Delta_{11} \\ \Delta_{22} \\ \vdots \\ \Delta_{dd} \\ \sqrt{2}\Delta_{12} \\ \vdots \\ \sqrt{2}\Delta_{1d} \\ \sqrt{2}\Delta_{23} \\ \vdots \\ \sqrt{2}\Delta_{2d} \\ \vdots \\ \sqrt{2}\Delta_{(d-1)d} \end{pmatrix}$$

The norm is given by:

$$\begin{aligned} ||\mu_{\hat{P}} - \mu_{\hat{Q}}||_{K'}^2 &= \sum_{i=1,\ldots,d} \Delta_{ii}^2 + \sum_{i=1,\ldots,d} \sum_{j=i+1,\ldots,d} 2\Delta_{ij}^2 \\ &= \sum_{i=1,\ldots,d} \Delta_{ii}^2 + \sum_{i \neq j} \Delta_{ij}^2 \\ &= \sum_{i=1\ldots d} \sum_{j=1\ldots d} \Delta_{ij}^2 = ||M||_F^2 \end{aligned}$$

And thus in this case it still holds that:

$$||M||_F = ||\mu_{\hat{P}} - \mu_{\hat{Q}}||_{K'}$$

Thus our analysis in the previous section holds for any arbitrary number of dimensions.

C.6. Extension to any Arbitrary Kernel

In this section we generalize these results to any arbitrary kernel K. The most difficult part in this derivation is showing that $g \in H' = \{\forall h \in \mathcal{H}' : ||h||_{K'} \leq \Lambda' = 4\Lambda^2\}$ as we did in Section C.2 for the linear kernel K. Before we do this we introduce some notation to work with two kernels and we briefly repeat some results from Section C.2. Afterward we first show that $g \in \mathcal{H}'$ and then show that $g \in H'$. Then we argue that the results from Section C.4 still hold: this was our main result that the discrepancy bound is always tighter than the MMD bound. We first do these derivations using the kernel K' since this is most easiest to show using this kernel. Then we show that this also holds for a kernel K'' which we can use to compute the MMD. This analysis shows how to choose the kernel of the MMD to take the hypothesis set into account for any arbitrary kernel K.

First we introduce some notation. We define the squared kernel K' as:

$$K'(f, h) = \langle f, h \rangle_K^2 \tag{C.3}$$

Where $f \in \mathcal{H}$ and $g \in \mathcal{H}$, where $\mathcal{H}$ is the RKHS of K. We indicate $\mathcal{H}'$ as the RKHS of K'. We assume K is a PSD kernel. By definition of K' the kernel K' is a PSD kernel since a squared kernel is known to be PSD[13, Theorem 5.3]. Now we have two kernels we have two featuremaps: $\psi_K(x)$ which maps the input space $\mathcal{X}$ to the RKHS of K (this map exists since we assume K is a PSD kernel), and $\psi_{K'}(h)$ which maps a vector from the RKHS of K to the RKHS of K'. Note that the second featuremap $\psi_{K'}(h)$ remains the same quadratic featuremap as before but now maps from $\mathcal{H}$ to $\mathcal{H}'$. See Table C.1 for an overview of the notation used.

Recall that because K is a PSD kernel we have that:

$$K(x, x') = \langle \psi_K(x), \psi_K(x') \rangle_K \tag{C.4}$$

For $x, x' \in \mathcal{X}$. Similarly for the kernel K' which is also PSD we have that:

$$K'(f, g') = \langle \psi_{K'}(f), \psi_{K'}(g) \rangle_{K'} \tag{C.5}$$

For $f, g \in \mathcal{H}$.

Now we briefly repeat a part of the derivation of C.2. We define z as:

$$z = h - f$$

The function g is given by:

$$g(x) = z(x)^2$$

As in Section C.2 we have that $||h||_K \leq \Lambda$ and $||f||_K \leq \Lambda$ since we assumed the realizable setting. Then it is straightforward to show that:

$$||z||_K = ||h - f||_K \leq 2\Lambda \tag{C.6}$$

Transformation		ψ_K		$\psi_{K'}$	
Space	$\mathcal{X}$	$\rightarrow$	$\mathcal{H}$	$\rightarrow$	$\mathcal{H}'$
Kernel			K		K'

Table C.1: This table illustrates the notation used when 2 kernels are involved.

Since h and f are in the RKHS of K, z is also in the RKHS of K. Thus we can write z as an innerproduct in the RKHS of K:

$$z(x) = \langle z, \psi_K(x) \rangle_K$$

Now we show that the function $g \in \mathcal{H}'$, in other words we show that g is in the RKHS of K'. By definition we have that:

$$g(x) = z(x)^2 = \langle z, \psi_K(x) \rangle_K^2$$

Now we can easily recognize our definition of K' in this equation (compare with Equation C.3), thus we note that:

$$g(x) = K'(z, \psi_K(x))$$

Now since K' is a PSD kernel, each kernel product corresponds to an innerproduct in its RKHS. Note that the vectors z and $\psi_K(x)$ are vectors in $\mathcal{H}$. To map these vectors to the RKHS of K' we need to use the featuremap $\psi_{K'}$. We thus apply Equation C.5 resulting in:

$$g(x) = \langle \psi_{K'}(z), \psi_{K'}(\psi_K(x)) \rangle_{K'}$$

We observe that g corresponds to the vector $\psi_{K'}(z) \in \mathcal{H}'$, and thus we have that $g \in \mathcal{H}'$.

Now we show that $||g||_{K'} \leq 4\Lambda^2$ to show that $g \in H'$. Since $g = \psi_{K'}(z) \in \mathcal{H}'$ the norm of g in K' is given by:

$$||g||_{K'} = \langle \psi_{K'}(z), \psi_{K'}(z) \rangle_{K'}$$

Now we can use Equation C.5 to rewrite this in terms of K'. We obtain:

$$||g||_{K'} = K'(z, z) \tag{C.7}$$

Using the definition of K' we find:

$$K'(z, z) = \langle z, z \rangle_K^2 = ||z||_K^2 \tag{C.8}$$

Now recall we showed earlier in Equation C.6 that $||z||_K \leq 2\Lambda$. Combining this with Equations C.7 and C.8 we find that:

$$||g||_{K'} = ||z||_K^2 \leq 4\Lambda^2$$

Thus we have shown that for any arbitrary kernel K that $g \in H' = \{\forall h \in \mathcal{H}' : ||h||_{K'} \leq \Lambda' = 4\Lambda^2\}$.

Now the proof that the discrepancy bound is tighter than the MMD bound can be shown in the exact same way as in Section C.4, only we have to work in the RKHS of K, thus everywhere x needs to be replaced by $\psi(x)_K$. Furthermore x_{ij} will become the jth component of $\psi(x_i)_K$, where x_i will be object i. This proof still holds, since the featuremap $\psi(x)_{K'}$ is still given by the featuremap of the squared kernel, however in this case the featuremap is with respect to the RKHS of K: this does not influence the proof. We showed that this holds for any arbitrary dimension, thus our results hold for a kernel with arbitrary dimension of the RKHS of K.

Finally, we show that to compute the MMD we can use the kernel $K''(x, x') = K(x, x')^2$. Since we require that kernel products are computed between objects in $\mathcal{X}$ to compute the MMD (see Equation A.8). As of now we defined the kernel $K'(f, h)$ so that it operates on $f, g \in \mathcal{H}$ and therefore we cannot use it to compute the MMD empirically using Equation A.8. To this end we show that $g \in H'' = \{\forall h \in \mathcal{H}'' : ||h||_{K''} \leq \Lambda' = 4\Lambda^2\}$ by showing that $\mathcal{H}' = \mathcal{H}''$, where $\mathcal{H}''$ is the RKHS of K''. Then we can use K'' to compute the MMD using Equation A.8 since K'' satisfies the assumptions of the MMD bound and K'' operates on objects in $\mathcal{X}$.

By definition of K'' we have that:

$$K''(x, x') = K(x, x')^2$$

Now using Equation C.4 we can show that:

$$K''(x, x') = K(x, x')^2 = \langle \psi_K(x), \psi_K(x') \rangle_K^2$$

Observe that this coincides with the definition of K' (Equation C.3), thus we can write this as:

$$K''(x, x') = \langle \psi_K(x), \psi_K(x') \rangle_K^2 = K'(\psi_K(x), \psi_K(x'))$$

Now using Equation C.5 we can write this as:

$$K''(x, x') = K'(\psi_K(x), \psi_K(x')) = \langle \psi_{K'}(\psi_K(x)), \psi_{K'}(\psi_K(x')) \rangle_{K'}$$

in other words, we see that the kernel product of K'' can be computed in the RKHS of the kernel K'. Thus, the RKHS of K' and K'' coincide! Thus we have that $\mathcal{H}' = \mathcal{H}''$ and we can generalize all results in terms of K' to the kernel K''. Therefore, g is also in the RKHS of K'', and in particular we have that $g \in H'' = \{\forall h \in \mathcal{H}'' : ||h||_{K''} \leq \Lambda' = 4\Lambda^2\}$. This could also be observed by noting that the featuremap of K'' is given by $\psi_{K''}(x) = \psi_{K'}(\psi_K(x))$ and thus maps to the space $\mathcal{H}'$, and from this it follows that $\mathcal{H}' = \mathcal{H}''$. So we can compute the MMD with this kernel K'' empirically using Equation A.8 when we want to take the hypothesis set into account, and the derivation of A.1 now holds for the kernel K'' in the realizable case.

Now we review to which kernel the kernel K'' corresponds in case we use a linear kernel for K and a Gaussian kernel for K. In the case of the linear kernel this is straightforward:

$$K''(x, x') = K(x, x')^2 = (x^T x')^2$$

thus K'' is the squared kernel. If we use a Gaussian kernel with bandwidth σ for the learning algorithm:

$$K(x, x') = \exp\left(-\frac{||x - x'||^2}{2\sigma^2}\right)$$

We obtain for the kernel K'' that:

$$K''(x, x') = K(x, x')^2 = \exp\left(-\frac{2||x - x'||^2}{2\sigma^2}\right) = \exp\left(-\frac{||x - x'||^2}{2\sigma'^2}\right)$$

Where we absorbed the factor of 2 in the exponent in σ', we obtain another Gaussian kernel with $\sigma' = \frac{\sigma}{\sqrt{2}}$. Thus, if one uses a Gaussian kernel with bandwidth σ for the learning algorithm, $\sigma' = \frac{\sigma}{\sqrt{2}}$ should be used for the kernel of the MMD if we wish to take into account the hypothesis set.

C.7. Comparison of the Assumptions of the MMD and the Discrepancy

We have shown that the bounds are comparable and that the discrepancy bound is always tighter. Does this then mean that the assumptions of the MMD are weaker — in other words, are there situations where the MMD bound can hold but where the discrepancy bound does not hold? We believe this is not the case in the realizable setting and thus that the assumptions are exactly the same, which we will argue below. Afterward also discuss the agnostic setting — in this setting we believe the MMD bound may have a smaller approximation error.

Assume that we are in the realizable setting. We may propose that the MMD considers more loss functions then the discrepancy. However, all loss functions z that are considered by the MMD are constructed by considering all possible $h \in H$ and $f \in H$ and taking their difference resulting in the function g and squaring this function. Since the mapping of the RKHS of K to the RKHS of kernel K' is one to one, the same functions are considered in the RKHS of K and K'. Since the discrepancy considers all possible $h \in H$ and all possible labeling functions $f \in H$, they thus take into account the same functions. So both methods take into account the same functions for f and h, and therefore the assumption exactly coincide.

Only in the agnostic setting may the MMD bound be more general. Since the MMD bound approximates the loss function g by a function $\tilde{g} \in H'$ (see also Appendix A.3), this loss function may be tighter, since the MMD does for example not constrain this loss function to be positive like the discrepancy. Therefore in the agnostic setting the MMD bound may have a smaller approximation error, or may the approximation error of the MMD vanish while the approximation term of the discrepancy may not vanish.

C.8. Agnostic MMD Bound that Always Holds

Since we have shown that if we take the hypothesis set into account for the MMD the MMD upper bounds the discrepancy, we can directly generalize the discrepancy bound given in Section 2.2.2 to the MMD quantity. This bound will then always hold, also in the agnostic case, as long as f is a deterministic labeling function similar to the discrepancy bound. This MMD bound has an extra term that measures the approximation error of approximating the function f by a function $h \in H$ similar to the discrepancy bound. The bound is given below:

Theorem 9 (Agnostic MMD Generalization bound) *Assume that for any $x \in \mathcal{X}$ and $h \in H$ that $L(h(x), f(x)) \leq C$. Then given any hypothesis $h \in H$ and a deterministic labeling function f we have that:*

$$|L_{\hat{P}}(h, f) - L_{\hat{Q}}(h, f)| \leq MMD(\hat{P}, \hat{Q}) + 2C\eta_{disc}(f, \hat{Q}, \hat{P})$$

Where the MMD is computed using the kernel $K'(x, x') = K(x, x')^2$, where K is the kernel of the learning algorithm, and $\eta_{disc}(f, \hat{Q}, \hat{P})$ is given by:

$$\eta_{disc}(f, \hat{Q}, \hat{P}) = \min_{\tilde{f} \in H} \left(\frac{1}{n_{\hat{P}}} \sum_{x \in \hat{P}} |\tilde{f}(x) - f(x)| + \frac{1}{n_{\hat{Q}}} \sum_{x \in \hat{Q}} |\tilde{f}(x) - f(x)| \right)$$

Note that using this technique we can generalize any bound in terms of the discrepancy to an MMD bound.

D

TED

D.1. Derivation of Stochastic TED Bound

In this appendix we briefly discuss how the bound from [7] is adapted to conform to our setting. This adaptation is required because we consider the supervised ridge regression model, and in [7] the sum of squared errors is minimized while our learning algorithm minimizes the mean squared error, see also appendix H.

Because of our different function class we find that:

$$\lambda_A = \lambda_T = \lambda n_{\hat{Q}}$$
$$\lambda_I = 0$$

Using the function class they consider (see [7, Equation 4]) we can compute the constant B for our setting. We have that:

$$||w||_2^2 \lambda_A + \lambda_I w^T X_{\hat{P}}^T L X_{\hat{P}} w = ||w||_2^2 \lambda_T = ||w||_2^2 \lambda n_{\hat{Q}} \leq B$$

Using our usual assumption that $w \in H$, we have that:

$$||w||_2^2 \leq \Lambda^2$$

Multiplying both sides by $\lambda n_{\hat{Q}}$ we find:

$$\lambda n_{\hat{Q}} ||w||_2^2 \leq \lambda n_{\hat{Q}} \Lambda^2 = B$$

Note that in the resulting bound λ_A depends on the model trained during active learning, and that Λ is used to bound the norm of the oracle hypothesis. For more detail see [7].

In [7] the following bound is given in Theorem 2:

$$\mathbb{E}\left[||X_{\hat{P}} w - X_{\hat{P}} w_o||_2^2\right] \leq (B + \sigma^2) \mathrm{tr}(X_{\hat{P}} (X_{\hat{Q}}^T X_{\hat{Q}} + \lambda_A I + \lambda_I X_{\hat{P}}^T L X_{\hat{P}})^{-1} X_{\hat{P}}^T)$$

Where w_o is the oracle hypothesis that generates the labels y according to $y_i = w_o^T x + \epsilon_i$ and ϵ is a random variable with zero mean and standard deviation σ, and w is the hypothesis trained on the set $X_{\hat{Q}}$.

Now using $\lambda_T = \lambda n_Q$, $\lambda_A = 0$ and $B = \lambda n_{\hat{Q}} \Lambda^2$ we obtain for our setting:

$$\mathbb{E}\left[||X_{\hat{P}} w - X_{\hat{P}} w_o||_2^2\right] \leq (\lambda n_{\hat{Q}} \Lambda^2 + \sigma^2) \mathrm{tr}(X_{\hat{P}} (X_{\hat{Q}}^T X_{\hat{Q}} + \lambda n_{\hat{Q}} I)^{-1} X_{\hat{P}}^T)$$

Now we divide by the number of samples to get the average loss on the set $\hat{P}$ of w with respect to w_o:

$$\mathbb{E}\left[L_{\hat{P}}(w, w_o)\right] \leq \frac{(\lambda n_{\hat{Q}} \Lambda^2 + \sigma^2)}{n_{\hat{P}}} \mathrm{tr}(X_{\hat{P}} (X_{\hat{Q}}^T X_{\hat{Q}} + \lambda n_{\hat{Q}} I)^{-1} X_{\hat{P}}^T) \tag{D.1}$$

Observe that since the trace is invariant under cyclic permutations we have that:

$$\mathbb{E}\left[L_{\hat{P}}(w, w_o)\right] \leq \frac{(\lambda n_{\hat{Q}}\Lambda^2 + \sigma^2)}{n_{\hat{P}}}\text{tr}(X_{\hat{P}}^T X_{\hat{P}}(X_{\hat{Q}}^T X_{\hat{Q}} + \lambda n_{\hat{Q}} I)^{-1}) \tag{D.2}$$

This way we can express the TED objective in terms of the covariance matrices of the sets $\hat{P}$ and $\hat{Q}$.

Now we apply a trick from [8] to derive the kernel version of this bound. Using the Woodbury matrix identity it can be shown that (see also [8, Equation 7]):

$$\begin{aligned} X_{\hat{P}}(X_{\hat{Q}}^T X_{\hat{Q}} + \lambda_T I)^{-1} X_{\hat{P}}^T &= \frac{1}{\lambda_T}(X_{\hat{P}} X_{\hat{P}}^T - X_{\hat{P}} X_{\hat{Q}}^T (X_{\hat{Q}} X_{\hat{Q}}^T + \lambda_T I)^{-1} X_{\hat{Q}} X_{\hat{P}}^T) \\ &= \frac{1}{\lambda_T}(K_{\hat{P}\hat{P}} - K_{\hat{P}\hat{Q}}(K_{\hat{Q}\hat{Q}} + \lambda_T I)^{-1} K_{\hat{P}\hat{Q}}^T) \end{aligned}$$

Substituting in the bound of Equation D.1 we obtain:

$$\begin{aligned} \mathbb{E}\left[L_{\hat{P}}(w, w_o)\right] \leq & \frac{(\lambda n_{\hat{Q}}\Lambda^2 + \sigma^2)}{n_{\hat{P}} n_{\hat{Q}}\lambda}\text{tr}(K_{\hat{P}\hat{P}} - K_{\hat{P}\hat{Q}}(K_{\hat{Q}\hat{Q}} + \lambda n_{\hat{Q}} I)^{-1} K_{\hat{P}\hat{Q}}^T) \\ = & \frac{(\Lambda^2 + \frac{\sigma^2}{n_{\hat{Q}}\lambda})}{n_{\hat{P}}}\text{tr}(K_{\hat{P}\hat{P}} - K_{\hat{P}\hat{Q}}(K_{\hat{Q}\hat{Q}} + \lambda n_{\hat{Q}} I)^{-1} K_{\hat{P}\hat{Q}}^T) \end{aligned}$$

Now we introduce our usual notation. Equivalently we can consider that the labels f are generated by $f_i = f'_i + \epsilon_i$, where $w_o = f' \in H$ and $w = h_{\hat{Q}}$ is the hypothesis trained on the set $\hat{Q}$. We obtain the bound:

Theorem 10 (Generalization bound TED [7]) *We assume the labels are generated by the process $f(x_i) = f'(x_i) + \epsilon_i$, where ϵ is a random variable with zero mean and standard deviation σ. The observations of ϵ_i are assumed to be independent. Furthermore assume that $f' \in H$. For the ridge regression model $h_{\hat{Q}}$ trained on the set $\hat{Q}$ with regularization parameter λ we can give the following generalization bound:*

$$\begin{aligned} \mathbb{E}\left[L_{\hat{P}}(h_{\hat{Q}}, f')\right] \leq & \frac{(\Lambda^2 + \frac{\sigma^2}{n_{\hat{Q}}\lambda})}{n_{\hat{P}}} tr(K_{\hat{P}\hat{P}} - K_{\hat{P}\hat{Q}}(K_{\hat{Q}\hat{Q}} + \lambda n_{\hat{Q}} I)^{-1} K_{\hat{P}\hat{Q}}^T) \\ = & \frac{(\Lambda^2 + \frac{\sigma^2}{n_{\hat{Q}}\lambda})}{n_{\hat{P}}} TED(\hat{P}, \hat{Q}) \end{aligned}$$

D.2. Derivation of Non-Stochastic TED Bound

In this section we take a similar approach as [15] to obtain a TED bound that holds in the agnostic setting. We will prove that in this section the following bound holds in the agnostic setting:

$$L_{\hat{P}}(h, f) \leq \frac{\Lambda^2}{n_{\hat{P}}}\text{TED}(\hat{P}, \hat{Q}) + 2C\eta_{TED}(\hat{P}, \hat{Q}, f)$$

Where h is the ridge regression model trained on the set $\hat{Q}$. Furthermore we require that f is a deterministic labeling function and we assume that for all $h' \in H$ and $x \in \mathcal{X}$ it holds that: $L(h'(x), f(x)) \leq C$. In this bound the approximation error of TED is defined as:

$$\eta_{TED}(\hat{P}, \hat{Q}, f) = \min_{\tilde{f} \in H}\left(\frac{1}{n_{\hat{P}}}\sum_{x \in \hat{P}}|f(x) - \tilde{f}(x)| + \frac{1}{n_{\hat{P}}}||\bar{H}||_1 \sum_{x \in \hat{Q}}|f(x) - \tilde{f}(x)|\right)$$

Note that this approximation term is similar to the approximation term of the discrepancy. First we will give the proof of this bound and afterwards we compare the approximation error of the TED generalization bound with the approximation error of the discrepancy generalization bound.

Recall that the realizable TED generalization bound can only be applied if all labels are generated by a model of our hypothesis class. The quantity $L_{\hat{P}}(h, f)$ that we aim to bound does not obey these assumptions, since f is not in H, and the model h is trained on the labels $f_{\hat{Q}}$, which are also not the outputs of a model in H. Therefore, we need to approximate f by a $\tilde{f} \in H$. Also, the model h needs to be approximated by a model $\tilde{h}$ that is trained on the labels outputted by $\tilde{f}$ on the set $\hat{Q}$. This is because the TED bound only holds if the labels that are to be predicted *and* the input labels are both the outputs of the *same* model that is generating the labels.

From here on we adopt the following notation. $\tilde{f}$ is a model in H, and $\tilde{h}$ is the model obtained by training on the set $\hat{Q}$ with labels supplied by $\tilde{f}$. h is the model trained on the set $\hat{Q}$ with the labels supplied by f.

Observe that:

$$L_{\hat{P}}(h, f) = |L_{\hat{P}}(h, f) + L_{\hat{P}}(\tilde{h}, \tilde{f}) - L_{\hat{P}}(\tilde{h}, \tilde{f})|$$

By rearranging the terms and applying the triangle inequality we can show that:

$$L_{\hat{P}}(h, f) = |L_{\hat{P}}(\tilde{h}, \tilde{f})| + |L_{\hat{P}}(h, f) - L_{\hat{P}}(\tilde{h}, \tilde{f})|$$

Now we can apply the TED bound to bound the first term, since this term obeys all assumptions of the TED bound: all labels in this case are generated by $\tilde{f} \in H$, and $\tilde{h}$ is the model trained on the labels generated by $\tilde{f}$. Thus:

$$L_{\hat{P}}(\tilde{h}, \tilde{f}) \leq \max_{\tilde{f} \in H} L_{\hat{P}}(\tilde{h}, \tilde{f}) \leq \frac{\Lambda^2}{n_{\hat{P}}} \text{TED}(\hat{P}, \hat{Q}) \tag{D.3}$$

Then we obtain the following bound which holds for any $\tilde{f} \in H$:

$$L_{\hat{P}}(h, f) = \frac{\Lambda^2}{n_{\hat{P}}} \text{TED}(\hat{P}, \hat{Q}) + |L_{\hat{P}}(h, f) - L_{\hat{P}}(\tilde{h}, \tilde{f})|$$

Since this bound holds for any $\tilde{f} \in H$, we can choose $\tilde{f}$ to minimize the bound, resulting in:

$$L_{\hat{P}}(h, f) = \frac{\Lambda^2}{n_{\hat{P}}} \text{TED}(\hat{P}, \hat{Q}) + \min_{\tilde{f} \in H} |L_{\hat{P}}(h, f) - L_{\hat{P}}(\tilde{h}, \tilde{f})| \tag{D.4}$$

The term on the right hand side plays the role of approximation error. It measures how well $\tilde{f}$ approximates f on the set $\hat{P}$, and how well $\tilde{h}$ approximates h on the set $\hat{P}$. This term is however difficult to compute since it is non-convex in $\tilde{f}$, and is difficult compare with the discrepancy approximation error. Therefore, we bound this term. Observe that:

$$|L_{\hat{P}}(h, f) - L_{\hat{P}}(\tilde{h}, \tilde{f})| = |L_{\hat{P}}(h, f) - L_{\hat{P}}(\tilde{h}, \tilde{f}) + L_{\hat{P}}(\tilde{h}, f) - L_{\hat{P}}(\tilde{h}, f)|$$

Rearranging the terms on the right hand side and applying the triangle inequality, we find that:

$$|L_{\hat{P}}(h, f) - L_{\hat{P}}(\tilde{h}, \tilde{f}) + L_{\hat{P}}(\tilde{h}, f) - L_{\hat{P}}(\tilde{h}, f)| \leq |L_{\hat{P}}(h, f) - L_{\hat{P}}(\tilde{h}, f)| + |L_{\hat{P}}(\tilde{h}, f) - L_{\hat{P}}(\tilde{h}\tilde{f})|$$

Thus observe that:

$$|L_{\hat{P}}(h, f) - L_{\hat{P}}(\tilde{h}, \tilde{f})| \leq |L_{\hat{P}}(h, f) - L_{\hat{P}}(\tilde{h}, f)| + |L_{\hat{P}}(\tilde{h}, f) - L_{\hat{P}}(\tilde{h}\tilde{f})| \tag{D.5}$$

Now we can use the μ-admissibility of the squared loss to bound both terms on the right in terms that are convex in $\tilde{f}$. For the squared loss we have that $\mu = 2C$, where $L(h(x), f(x)) \leq C$ for all $x \in \mathcal{X}$ and all $h \in H$, see appendix C of [15]. Then by the μ-admissibility of the squared loss we have (see also [15]):

$$|L_{\hat{P}}(\tilde{h}, f) - L_{\hat{P}}(\tilde{h}, \tilde{f})| \leq 2C \frac{1}{n_{\hat{P}}} \sum_{x \in \hat{P}} |f(x) - \tilde{f}(x)| \tag{D.6}$$

And:

$$|L_{\hat{P}}(h,f) - L_{\hat{P}}(\tilde{h},f)| \leq 2C\frac{1}{n_{\hat{P}}}\sum_{x\in\hat{P}}|h(x)-\tilde{h}(x)| \tag{D.7}$$

Now we aim to rewrite the term above, since this term does not occur in the discrepancy bound. We need to introduce some notation to bound this term. Recall that h is the model trained on the labels f and $\tilde{h}$ is the model trained on the labels of $\tilde{f}$. We write $h_{\hat{P}}$ and $\tilde{h}_{\hat{P}}$ as the vector of model outputs on the set $\hat{P}$ of models h and $\tilde{h}$, respectively. Furthermore, we write $f_{\hat{Q}}$ as the label vector on the set $\hat{Q}$, and $\tilde{f}_{\hat{Q}}$ as the outputs of the model $\tilde{f}$ on the set $\hat{Q}$. For ridge regression we can define a hat matrix $\bar{H}$. In terms of $\bar{H}$ we have that:

$$\begin{aligned} h_{\hat{P}} =& \bar{H}f_{\hat{Q}} \\ \tilde{h}_{\hat{P}} =& \bar{H}\tilde{f}_{\hat{Q}} \end{aligned}$$

Observe that we can bound the right hand side of Equation D.7 as follows:

$$\begin{aligned} 2C\frac{1}{n_{\hat{P}}}\sum_{x\in\hat{P}}|h(x)-\tilde{h}(x)| =& 2C\frac{1}{n_{\hat{P}}}||\bar{H}f_{\hat{Q}} - \bar{H}\tilde{f}_{\hat{Q}}||_1 \\ \leq& 2C\frac{1}{n_{\hat{P}}}||\bar{H}||_1||f_{\hat{Q}} - \tilde{f}_{\hat{Q}}||_1 \end{aligned} \tag{D.8}$$

The last step to obtain Equation D.8 was allowed due to the subordinance property of the $p=1$ operator matrix norm. Rewriting the right hand side of Equation D.8 we obtain:

$$2C\frac{1}{n_{\hat{P}}}|\bar{H}|_1|f_{\hat{Q}} - \tilde{f}_{\hat{Q}}|_1 = 2C\frac{1}{n_{\hat{P}}}||\bar{H}||_1\sum_{x\in\hat{Q}}|f(x)-\tilde{f}(x)|$$

Therefore we have that:

$$|L_{\hat{P}}(h,f) - L_{\hat{P}}(\tilde{h},f)| \leq 2C\frac{1}{n_{\hat{P}}}||\bar{H}||_1\sum_{x\in\hat{Q}}|f(x)-\tilde{f}(x)| \tag{D.9}$$

Combining Equation D.5 with Equation D.6 and D.9 we obtain:

$$|L_{\hat{P}}(h,f) - L_{\hat{P}}(\tilde{h},\tilde{f})| \leq 2C\frac{1}{n_{\hat{P}}}\sum_{x\in\hat{P}}|f(x)-\tilde{f}(x)| + 2C\frac{1}{n_{\hat{P}}}||\bar{H}||_1\sum_{x\in\hat{Q}}|f(x)-\tilde{f}(x)|$$

Now combining this result with Equation D.4 we obtain our final result:

$$|L_{\hat{P}}(h,f)| \leq \frac{\Lambda^2}{n_{\hat{P}}}\text{TED}(\hat{P},\hat{Q}) + \min_{\tilde{f}\in H} 2C\left(\frac{1}{n_{\hat{P}}}\sum_{x\in\hat{P}}|f(x)-\tilde{f}(x)| + \frac{1}{n_{\hat{P}}}||\bar{H}||_1\sum_{x\in\hat{Q}}|f(x)-\tilde{f}(x)|\right)$$

This concludes our proof.

Now we define the approximation error $\eta_{TED}(\hat{P},\hat{Q},f)$ as:

$$\eta_{TED}(\hat{P},\hat{Q},f) = \min_{\tilde{f}\in H}\left(\frac{1}{n_{\hat{P}}}\sum_{x\in\hat{P}}|f(x)-\tilde{f}(x)| + \frac{1}{n_{\hat{P}}}||\bar{H}||_1\sum_{x\in\hat{Q}}|f(x)-\tilde{f}(x)|\right) \tag{D.10}$$

Resulting in the following generalization bound:

$$L_{\hat{P}}(h,f) \leq \frac{\Lambda^2}{n_{\hat{P}}}\text{TED}(\hat{P},\hat{Q}) + 2C\eta_{TED}(\hat{P},\hat{Q},f)$$

Now we will compare the approximation error of TED and the discrepancy. For the discrepancy the approximation error is given by:

$$\eta_{\text{disc}}(f, \hat{Q}, \hat{P}) = \min_{\tilde{f} \in H} \left(\frac{1}{n_{\hat{P}}} \sum_{x \in \hat{P}} |\tilde{f}(x) - f(x)| + \frac{1}{n_{\hat{Q}}} \sum_{x \in \hat{Q}} |\tilde{f}(x) - f(x)| \right)$$

Comparing with Equation D.10, observe that the TED approximation error will be guaranteed to be smaller if:

$$\frac{1}{n_{\hat{P}}} ||\bar{H}||_1 \leq \frac{1}{n_{\hat{Q}}}$$

However, in deriving this approximation term for TED we have made the bound looser to turn the computation of the approximation term into a convex optimization problem. The same holds for the discrepancy approximation term which can also be bounded tighter. An in depth comparison of Equation D.4 which is the tightest bound for TED and Equation B.5 which is the tightest bound for the discrepancy is needed to conclude which bound is truly tighter. This result in any case indicates that the bound on the TED approximation error is comparable in size to the bound on the discrepancy approximation error.

E

Probabilistic Nuclear Discrepancy Generalization Bound

We can compute the probabilistic generalization bound using the fact that:

$$\mathop{\mathbb{E}}_{u}\left[|L_{\hat{P}}(w,w') - L_{\hat{Q}}(w,w')|\right] = \mathop{\mathbb{E}}_{u}\left[|u^T M u|\right] = \mathop{\mathbb{E}}_{\bar{u}}\left[|\sum_i \bar{u}_i^2 \lambda_i|\right] \tag{E.1}$$

This equation can be derrived by following the derrivation in Appendix B.1 where not the maximum is taken over u but by taking the expectation with respect to u.

Here the vector $\bar{u}_i$ is the projection of u on the eigenvector v_i corresponding to the eigenvalue λ_i of M. This expectation is difficult to compute. Therefore we assume for simplicity that all $\bar{u}_i$ are independent and identically distributed according to some distribution $p(\bar{u}_i)$. Even in this case the expectation remains difficult to compute, therefore we bound the term on the right hand side of Equation E.1 using the triangle inequality:

$$\mathop{\mathbb{E}}_{\bar{u}}\left[|\sum_i \bar{u}_i^2 \lambda_i|\right] \leq \mathop{\mathbb{E}}_{\bar{u}}\left[\sum_i |\bar{u}_i^2 \lambda_i|\right] \approx \left(\sum_i |\lambda_i| \int \bar{u}_i^2 p(\bar{u}_i) d\bar{u}_i\right) \tag{E.2}$$

Where the approximately equal sign comes from the i.i.d. assumption on the components of $\bar{u}$. Now it is straightforward to show that the right hand side of Equation E.2 can be rewritten as:

$$\left(\sum_i |\lambda_i| \int \bar{u}_i^2 p(\bar{u}_i) d\bar{u}_i\right) = \left(\int \bar{u}_1^2 p(\bar{u}_1) d\bar{u}_1\right) \sum_i |\lambda_i| \propto \sum_i |\lambda_i|$$

Thus we find that a probabilistic generalization bound is given by:

$$\mathop{\mathbb{E}}_{\bar{u}}\left[|L_{\hat{P}}(w,w') - L_{\hat{Q}}(w,w')|\right] \leq \left(\int \bar{u}_1^2 p(\bar{u}_1) d\bar{u}_1\right) \sum_i |\lambda_i| \propto \sum_i |\lambda_i|$$

F

Bounding the Hypothesis Set

In this section we show how to bound the hypothesis space of the ridge regression model. We require this so given a λ and the ridge regression algorithm, we know in which set the resulting ridge regression model h will be. We denote this set by the hypothesis set $H = \{h \in \mathcal{H} : ||h||_K \leq \Lambda\}$. In this section we aim to find the constant Λ. We consider two techniques to bound the hypothesis set, one from [13] and one from [4]. In [4] a new technique is used to bound Λ which can possibly result in tighter bounds than the technique of [13]. However we show that this bound on Λ is not informative in Section F.1. Afterward, we review the technique from [13] in Section F.2 to bound Λ, resulting in $\Lambda = \sqrt{\frac{1}{\lambda}}$.

F.1. New Bound that is Uninformative in Most Practical Cases

In lemma 1 of [4] the bound on the hypothesis set is chosen as:

$$||h||_K \leq \sqrt{\frac{\mu r}{\lambda}}$$

However, after careful studying of lemma 1 we have determined that the square root is a mistake, the bound should be:

$$||h||_K \leq \frac{\mu r}{\lambda} = \Lambda$$

In these bounds, r is defined as: $K(x, x) \leq r^2$, and μ is the μ admissibility of the loss (see [4] for the definition). For the squared loss, $\mu = 2C$ [4], where $L(h(x), f(x)) \leq C$ for any $x \in \mathcal{X}$ and any $h \in H$ and f[1]. For simplicity we take the Gaussian kernel where $r = 1$. To explicitly calculate Λ, we need to know C. We can fix C as:

$$\max_{h \in H} \max_{f \in \{-1,+1\}} \max_{x \in \mathcal{X}} L(h(x), f) \leq \max_{h \in H} \max_{x \in \mathcal{X}} (|h(x)| + 1)^2 \leq (\Lambda r + 1)^2 = C \tag{F.1}$$

Since for this choice of C it is guaranteed that $L(h(x), f(x)) \leq C$. We relax the assumption $f \in \{-1, +1\}$ later. In Equation F.1 we used that for any $x \in \mathcal{X}$ and any $h \in H$:

$$|h(x)| \leq \Lambda r$$

Since:

$$|h(x)| = | < \psi(x), h >_K | \leq ||\psi(x)||_K ||h(x)||_K \tag{F.2}$$

[1] Observe we use the notation C for the constant for which $\mu = 2C$. [4] uses the constant M. We use C to avoid confusion with the matrix M.

Where we used the Cauchy-Schwarz inequality in Equation F.2. We can rewrite the right hand side of Equation F.2 as:

$$||\psi(x)||_K||h(x)||_K = \sqrt{\psi(x)^T\psi(x)}||h(x)||_K = \sqrt{K(x,x)}||h(x)||_K \leq \Lambda r$$

We can now construct the quadratic inequality for Λ. We obtain:

$$||h||_K \leq \Lambda = \frac{\mu r}{\lambda} = \frac{2(\Lambda+1)^2}{\lambda}$$

We can solve this quadratic equation for Λ, in case we have equality we find:

$$\Lambda^2 + (2 - \frac{\lambda}{2})\Lambda + 1 = 0$$

The discriminant is:

$$D = (2 - \frac{\lambda}{2})^2 - 4$$

We find the solution for Λ in the equality case is only positive and real if $\lambda \geq 8$, since for this case $D \geq 0$ (we don't consider the case $\lambda \leq 0$). Thus it is only possible to bound Λ if $\lambda \geq 8$. However, typically $\lambda << 1$, and thus this bound on Λ is not applicable in many cases.

F.2. Informative Bound

Following [13, Lemma 11.1] we can bound the hypothesis space as $\Lambda = \frac{1}{\sqrt{\lambda}}$. We give a summary of the proof here.

The kernel ridge regression algorithm minimizes the following objective with respect to $h \in H$:

$$\frac{1}{m}\sum_{x\in\hat{S}} L(h(x), f(x)) + \lambda||h||_K^2 \tag{F.3}$$

Where $\hat{S}$ is any empirical dataset. Consider any arbitrary element $h^* \in H$ returned by our KRR algorithm. In the minimization of Equation F.3, the element $h' = 0$ is also considered since h' is included in any set H since $\Lambda > 0$. Therefore, since h' is considered in the minimization, h' is either the best hypothesis ($h^* = h'$) or a different hypothesis h^* has a smaller objective value than h'. Thus:

$$\frac{1}{m}\sum_x L(h^*(x), f(x)) + \lambda||h^*||_K^2 \leq \frac{1}{m}\sum_x L(0, f(x)) + \lambda 0$$
$$\frac{1}{m}\sum_x L(h^*(x), f(x)) + \lambda||h^*||_K^2 \leq \frac{1}{m}\sum_x L(0, f(x))$$
$$\frac{1}{m}\sum_x L(h^*(x), f(x)) + \lambda||h^*||_K^2 \leq 1$$

Here we used that the losses for $h' = 0$ cannot be larger than 1 since $f \in \{-1, +1\}$, we relax this assumption later. Since the squared loss is non negative, we have as well that:

$$\lambda||h^*||_K^2 \leq 1 \tag{F.4}$$

Note that by the same argument it holds that:

$$\frac{1}{m}\sum_x L(h^*(x), f(x)) \leq 1$$

Rewriting Equation F.4 we find that:

$$||h^*||_K^2 \leq \frac{1}{\lambda}$$
$$||h^*||_K \leq \sqrt{\frac{1}{\lambda}} = \Lambda$$

Thus for any empirical dataset, we obtain h^* after training with $||h^*||_K \leq \sqrt{\frac{1}{\lambda}}$ using the KRR algorithm, so we can choose our hypothesis set as: $H = \{\forall h \in \mathcal{H}| \quad ||h||_K \leq \Lambda = \frac{1}{\sqrt{\lambda}}\}$. In case $f(x) \notin \{+1, -1\}$ we can adapt the bound as follows. Take $|f(x)| \leq f_{\max}$. Then it is straightforward to show that for any h^* obtained from the RR optimization obeys:

$$||h^*||_K \leq \frac{f_{\max}}{\sqrt{\lambda}}$$

And we can bound the hypothesis set with $\Lambda = \frac{f_{\max}}{\sqrt{\lambda}}$.

G

Why Did We Use the Gaussian Kernel?

This appendix addresses the question: why did we use the Gaussian kernel in the experiments with the benchmark datasets? We especially adress why we did not use the simpler linear kernel. We answer this using two theoretical arguments.

First and foremost, these active learning techniques were constructed to work in both the agnostic and realizable scenario. First we briefly discuss the realizeable setting. If we for example use a linear kernel to convert the benchmark datasets to the realizeable setting, the active learning problem becomes too easy to be informative. In this case we only approximately need to select the number of samples equal to the dimensionality. One may wonder, why did we then not use more high-dimensional linear datasets? Well, since if we use the Gaussian kernel, we essentially convert the datasets to a very high dimensional dataset. In this case the samples may be in a subspace of the Hilbert space with the number of dimensions equal to the number of samples. For example for the `ringnorm` dataset this was the case with our settings of the hyperparameters. So these experiments are also very informative for the performance in high-dimensional datasets.

In the agnostic scenario it is important for all bounds to approximate the labeling function reasonably well by a function from the hypothesis set. We use the Gaussian kernel since it is a universal appropriator. We briefly argue why in this case the approximation errors will be small.

All bounds share the following common term in the approximation error:

$$\min_{h \in H} \frac{1}{n_{\hat{P}}} \sum_{x \in \hat{P}} |h(x) - f(x)|$$

This term can be rewritten as:

$$\min_{h \in H} \frac{1}{n_{\hat{P}}} \sum_{x \in \hat{P}} |h(x) - f(x)| = \min_{h \in H} \frac{1}{n_{\hat{P}}} ||h - f||_1$$

Where we use the notation h and f to indicate the vector of outputs of the model and labeling function on the set $\hat{P}$. It is straightforward to show that:

$$\min_{h \in H} \frac{1}{n_{\hat{P}}} ||h - f||_1 \leq \min_{h \in H} ||h - f||_\infty$$

Now if we assume that the regularization parameter λ is small, we have that:

$$\min_{h \in H} ||h - f||_\infty \approx \min_{h \in \mathcal{H}} ||h - f||_\infty$$

Where $\mathcal{H}$ is the entire RKHS associated with the kernel K of the hypothesis set. Now since the Gaussian kernel is universal, it can be shown that if f is bounded [23]:

$$\min_{h \in \mathcal{H}} ||h - f||_\infty < \epsilon$$

For any $\epsilon > 0$. Therefore if we choose ϵ small we have that:

$$\min_{h \in \mathcal{H}} ||h - f||_\infty \approx 0$$

Thus if the regularization parameter is sufficiently small, we have that:

$$\min_{h \in H} \frac{1}{n_{\hat{P}}} \sum_{x \in \hat{P}} |h(x) - f(x)| \approx 0$$

For the other approximation terms in all bounds this can be shown similarly. For non-universal kernels, even if the regularization parameter is small (or zero), we may have that this approximation error will not vanish. For example for the linear kernel it is very unlikely that the approximation error will vanish. Note that we have assumed that the regularization parameter sufficiently small. This may not be the case in general, also not in our experiments, and therefore the approximation errors in the agnostic setting may not be zero, however they will likely be small.

This is also another argument why we used the Gaussian kernel for the benchmark datasets in the realizeable setting. Since the Gaussian kernel likely can model the labels better, if we use the Gaussian kernel to approximate the labels for the realizeable setting, the labeling function likely more accurately approximates the labeling function of the agnostic case. Therefore, performance in this realizeable setting will be more informative for performance in the agnostic setting.

We have another theoretical argument to justify the use of a universal kernel when the MMD and discrepancy are used. In [4] and [27] it is shown that only and only if $\hat{P} = \hat{Q}$, the discrepancy and MMD quantities are equal to zero. For non-universal kernels, for example in the linear kernel, we have that the discrepancy or MMD quantities are equal to zero while the $\hat{P} \neq \hat{Q}$. In particular, if the covariance matrices in the linear kernel are equal for the sets $\hat{P}$ and $\hat{Q}$ the MMD HS and the discrepancy will be zero. This indicates that these measures cannot fully characterize the difference between empirical distributions that for example have different means or different higher order moments. The MMD and the discrepancy are motivated by the fact they can measure the difference between empirical distributions, to this end this comparison should include all moments of the empirical distribution. For more details see the discussion in [4].

In this Appendix we have given a brief theoretical argumentation for our choice of the Gaussian kernel that we use in our experiments on the benchmark datasets.

H

Ridge Regression

We choose the learning algorithm that minimizes the following objective for $h \in H$:

$$F_{(\hat{Q},f)} = \frac{1}{n_{\hat{Q}}} \sum_{i=1}^{n_{\hat{Q}}} (h(x_i) - f(x_i))^2 + \lambda ||h||_K^2$$

This is the training algorithm required for the derivation of the discrepancy bound to hold. However, note that Ridge Regression typically minimizes:

$$F'_{(\hat{Q},f)} = \sum_{i=1}^{n_{\hat{Q}}} (h(x_i) - f(x_i))^2 + \lambda_T ||h||_K^2$$

Where we use a different regularization λ_T to indicate that this is the *traditional* formulation. It can be shown that if $\lambda = \frac{\lambda_T}{n_{\hat{Q}}}$ minimizing $F_{(\hat{Q},f)}$ and $F'_{(\hat{Q},f)}$ results in the same model h. The closed form solution of w minimizing F' is given in the linear kernel by:

$$w = (X_{\hat{Q}}^T X_{\hat{Q}} + \lambda_T I)^{-1} X_{\hat{Q}}^T y_{\hat{Q}}$$

Where we adopted the notation $y_{\hat{Q}}$ to indicate the label vector of the objects in the set $\hat{Q}$. When a kernel K is used we have that the prediction of h on an object x is given by:

$$h(x) = \sum_i \alpha_i K(x_i, x)$$

In this case, the vector α is given by:

$$\alpha = (K_{\hat{Q}\hat{Q}} + \lambda_T I)^{-1} y_{\hat{Q}}$$

See also [13]. If we want to compute the predictions on all objects in $\hat{P}$ by the model trained on $\hat{Q}$, we can use the equation:

$$h_{\hat{P}} = \bar{H} y_{\hat{Q}}$$

Where $h_{\hat{P}}$ is used to indicate the vector of predictions on the set $\hat{P}$. The matrix $\bar{H}$ is called the hat matrix, and is given by:

$$\bar{H} = K_{PQ}(K_{QQ} + \lambda_T I)^{-1}$$

Using the substitution $\lambda_T = \lambda n_{\hat{Q}}$ we can now compute the hypothesis h or w that minimizes $F_{(\hat{Q},f)}$ in a straightforward manner.

I

Comparison with Batch MMD Active Learner

In this work we focus on sequential active learners instead of batch active learners because we are more interested in the underlying objectives of these active learners than their optimization procedure. In this section we show that the state-of-the-art MMD batch active learner of [6] actually performs worse than the sequential active learner in terms of the MMD objective.

We used the 'quadprog' function of MATLAB to solve the quadratic programming problem described in [6] for batch active learning. For the optimization we used the following settings: we limited the optimization to 1000 iterations, and we set the function, constraint and x difference tolerances to 10^{-6}. We used a batch size of 5 samples. We compute the MMD between the sets $\hat{P}$ and $\hat{Q}$ of the selected examples by the MMD batch learner and the MMD sequential active learner on the datasets listed in Table 3.1. We repeated the experiment 100 times and averaged the resulting MMD values.

The results are shown in Figure I.1. We see from the results that the objective of MMD HS sequential is consistently lower. Thus the batch optimization fails to achieve a lower MMD value, and therefore the sequential active learner is actually preferable in terms of the underlying objective. This likely occurs because the convex relaxation of the constraints change the optimization objective significantly.

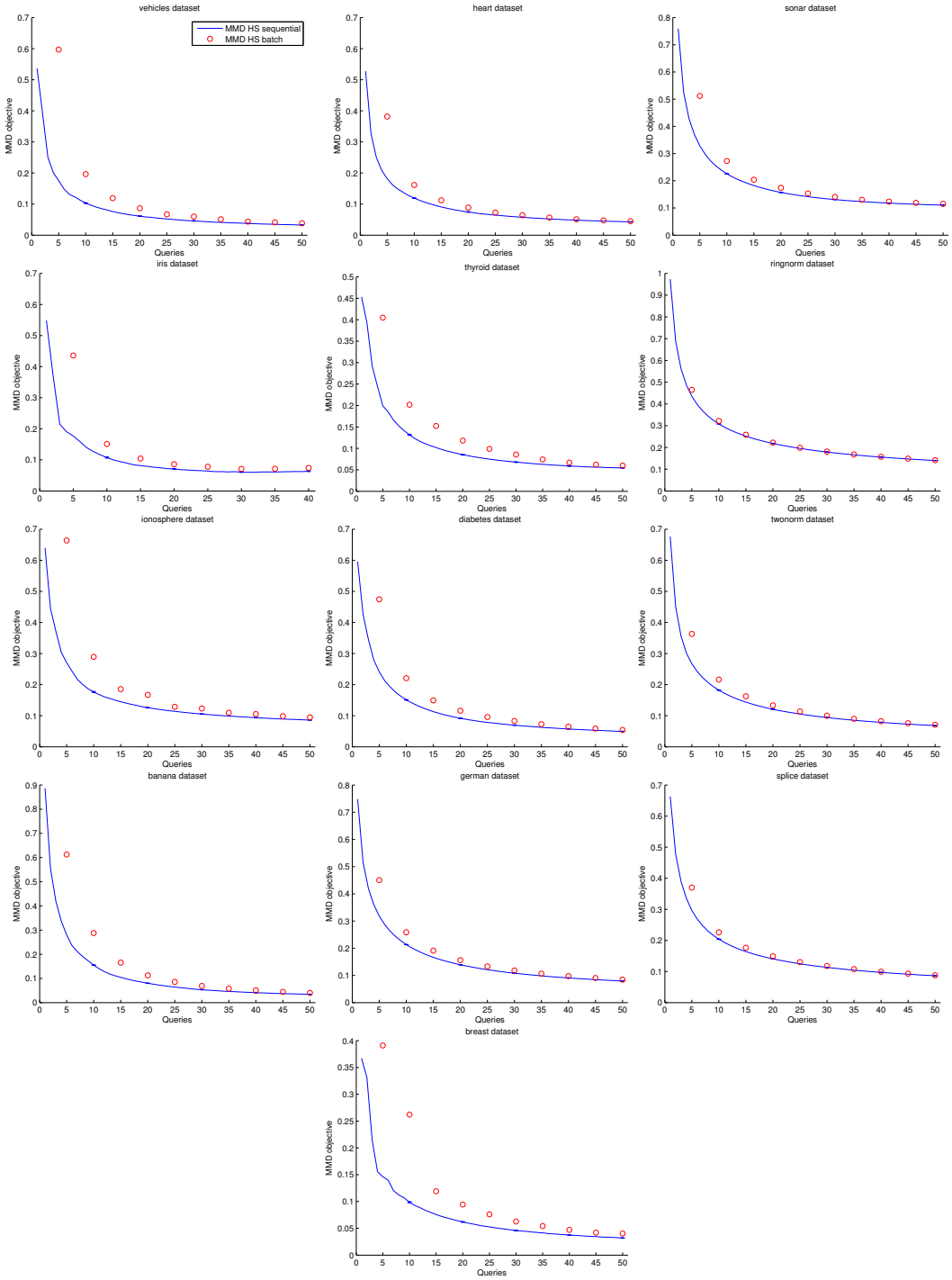


Figure I.1: Comparison between the sequential MMD HS and batch MMD HS active learner. We observe the objective value for the sequential active learner is consistently lower and thus the batch optimization does not seem beneficial in terms of the MMD objective.

J

Computing the Projection of u on the Eigenvectors of M

The computation of $\bar{u}_i$, the projection of u onto the eigenvector v_i of M is non-trivial. Observe that here v_i is a vector and not a component. We give a detailed description in this appendix how to compute $\bar{u}_i$. The equation for $\bar{u}_i$ is:

$$\bar{u}_i = \frac{u^T v_i}{\sqrt{v_i^T v_i}} \tag{J.1}$$

The difficulty in this derivation is finding the vector v_i in case kernels are used. In that case we need to find v_i expressed in terms of the datamatrix X. Then we can apply the 'kernel trick' to compute Equation J.1.

Note that in the linear kernel we have:

$$Mv_i = \lambda_i v_i \tag{J.2}$$

In case of the linear kernel it is straightforward to compute v_i. To compute v_i when kernels are used, first we show that v_i can be expressed in terms of the datamatrix X, and afterward we find this expression of v_i in terms of X. Note that:

$$Mv_i = \sum_{j=1}^{n_{\hat{P}}} d_j x_j x_j^T v_i = \sum_{j=1}^{n_{\hat{P}}} (x_j^T v_i) d_j x_j = \lambda_i v_i$$

Thus we have that:

$$\sum_{j=1}^{n_{\hat{P}}} \frac{(x_j^T v_i) d_j}{\lambda_i} x_j = v_i$$

Thus we have that each eigenvector v_i is a linear combination of the vectors x_j. Here the sum is taken over all objects $x \in \hat{P}$. Since $\hat{Q} \in \hat{P}$, this includes all data the active learner has access to. Then we can write each eigenvector v_i as:

$$v_i = X_{\hat{P}}^T \alpha_i \tag{J.3}$$

Thus we can express each vector v_i using the datamatrix $X_{\hat{P}}$. Now we will have to find the vector α_i to find v_i. We substitute the equation above in equation J.2 to obtain:

$$MX_{\hat{P}}^T \alpha_i = \lambda_i X_{\hat{P}}^T \alpha_i$$

Now we multiply left with $DX_{\hat{P}}$ on both sides to obtain:

$$DX_{\hat{P}}MX_{\hat{P}}^T\alpha_i = \lambda_i DX_{\hat{P}}X_{\hat{P}}^T\alpha_i$$

Observe that this is equal to:

$$M_K^T M_K^T \alpha_i = \lambda_i M_K^T \alpha_i$$

Where M_K was defined in Equation B.4. Now we define $\beta_i = M_K^T\alpha_i$. Then we find:

$$M_K^T\beta_i = \lambda_i\beta_i \tag{J.4}$$

We can compute the eigenvectors β by computing the eigendecomposition of M_K^T. This is possible even when using kernels, since M_K is expressed in terms of the kernel matrix. However we require the vector α_i to compute the eigenvector v_i. Thus now we will aim to express α_i in terms of β_i. Observe that if we multiply equation J.4 by $(M_K^T)^{-1}$ on both sides we obtain:

$$\beta_i = \lambda_i(M_K^T)^{-1}\beta_i \tag{J.5}$$

Now observe that due to the definition of β_i we have that:

$$\beta_i(M_K^T)^{-1} = \alpha_i \tag{J.6}$$

Combining equation J.5 and J.6 we find that:

$$\alpha_i = \frac{\beta_i}{\lambda}$$

Substituting this in equation J.3 we find the vector v_i:

$$v_i = X_{\hat{P}}^T\frac{\beta_i}{\lambda} \tag{J.7}$$

Now we have found v_i. Now we can proceed to compute u_i.

Note that due to the representer theorem we have that:

$$u = w' - w = X_{\hat{D}}^T c' - X_{\hat{Q}}^T c = X_{\hat{D}}^T\tilde{c} \tag{J.8}$$

Here w' is given as a linear combination of $X_{\hat{D}}$, which we define as the complete datamatrix. This datamatrix includes the training and testset, since w' can be (for example) be obtained by training on the whole dataset where the original binary labels of the dataset are used. However note that for any $w' \in H$ the model w' can be written in this way. Similarly, since w is trained on the dataset $\hat{Q}$, we can write w as a linear combination of objects in $\hat{Q}$. Combining equation J.7 and J.8 with equation J.1 we find that:

$$u_i = \frac{\tilde{c}X_{\hat{D}}X_{\hat{P}}^T\frac{\beta_i}{\lambda_i}}{\sqrt{\frac{\beta_i^T}{\lambda_i}X_{\hat{P}}X_{\hat{P}}^T\frac{\beta_i}{\lambda_i}}} = \frac{\tilde{c}K_{\hat{D}\hat{P}}\beta_i}{\sqrt{\beta_i K_{\hat{P}\hat{P}}\beta_i}}$$

K

Additional Results

In most sections we did not include all learning curves and results, since most of the time we summarized the results using illustrative learning curves or tables. However, these of course do not tell the whole story. In this appendix all results that were not printed in the main matter are shown. Observe that on each page all results on all benchmark datasets are always shown in the same order. Therefore, one can easily compare methods across different settings.

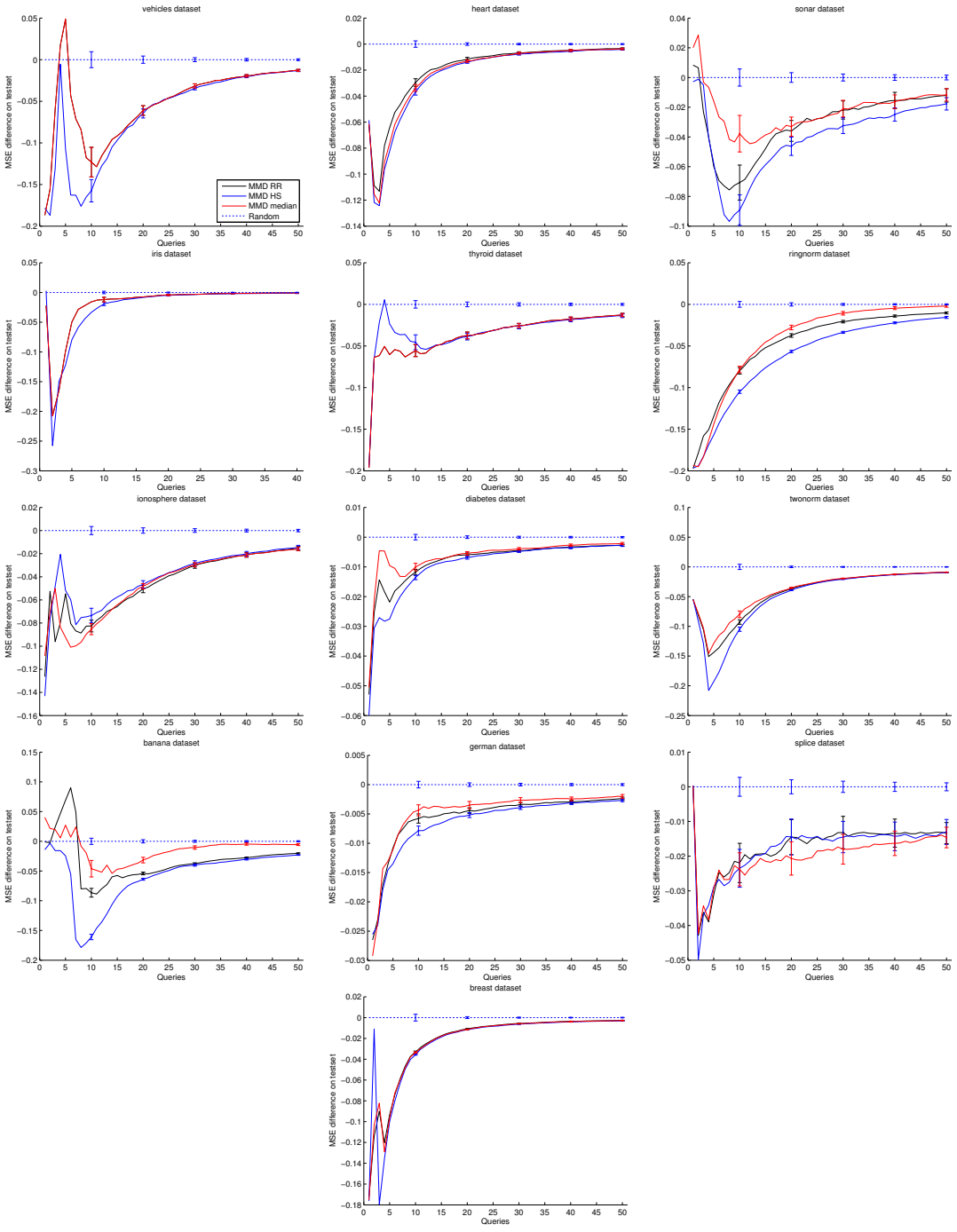


Figure K.1: Comparison between MMD HS which takes the hypothesis set into account according to our theoretical analysis and MMD median and MMD RR which do not on all benchmark datasets in the realizable setting where $f \in H$.

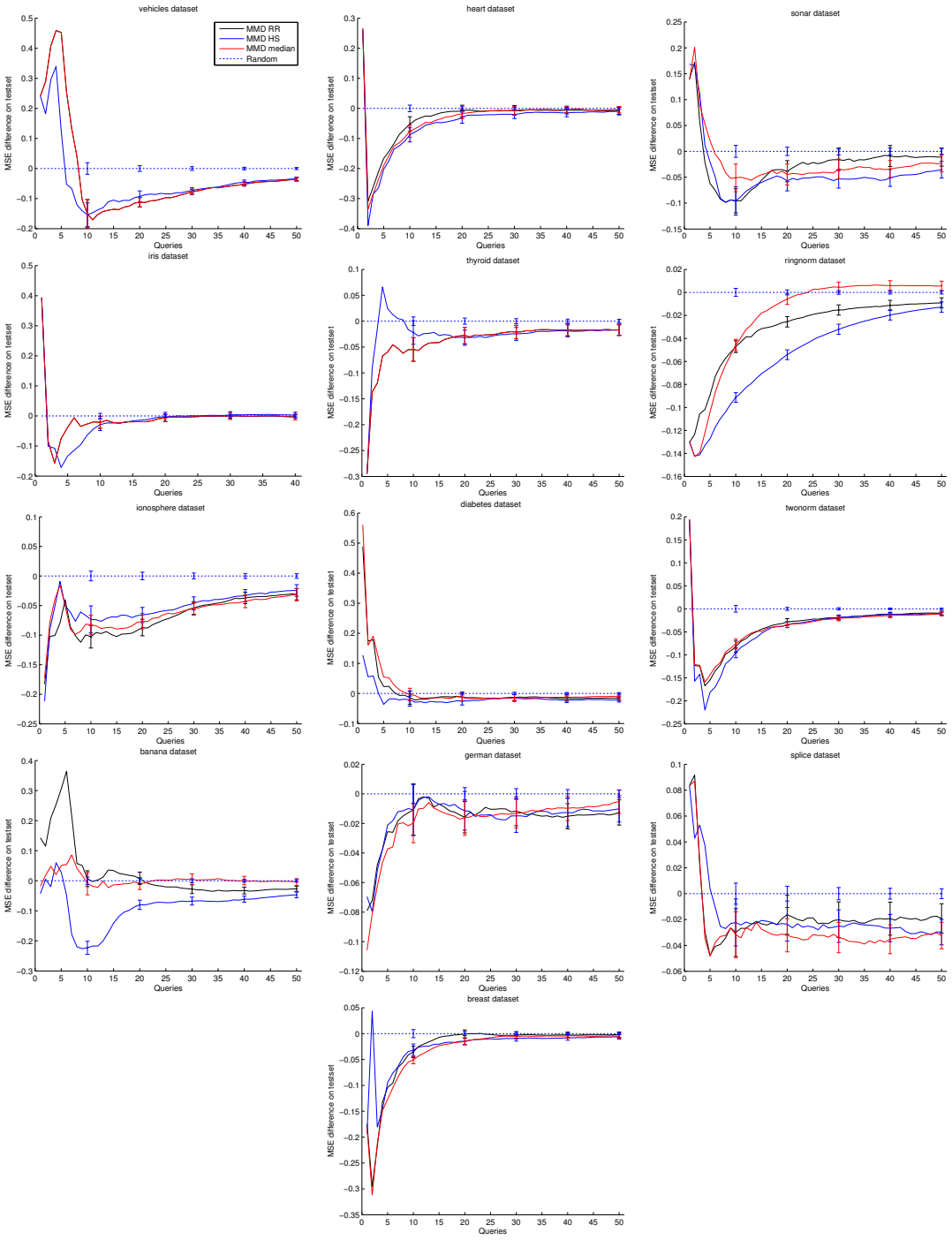


Figure K.2: Comparison between MMD HS which takes the hypothesis set into account according to our theoretical analysis and MMD median and MMD RR which do not on all benchmark datasets in the agnostic setting where $f \notin H$

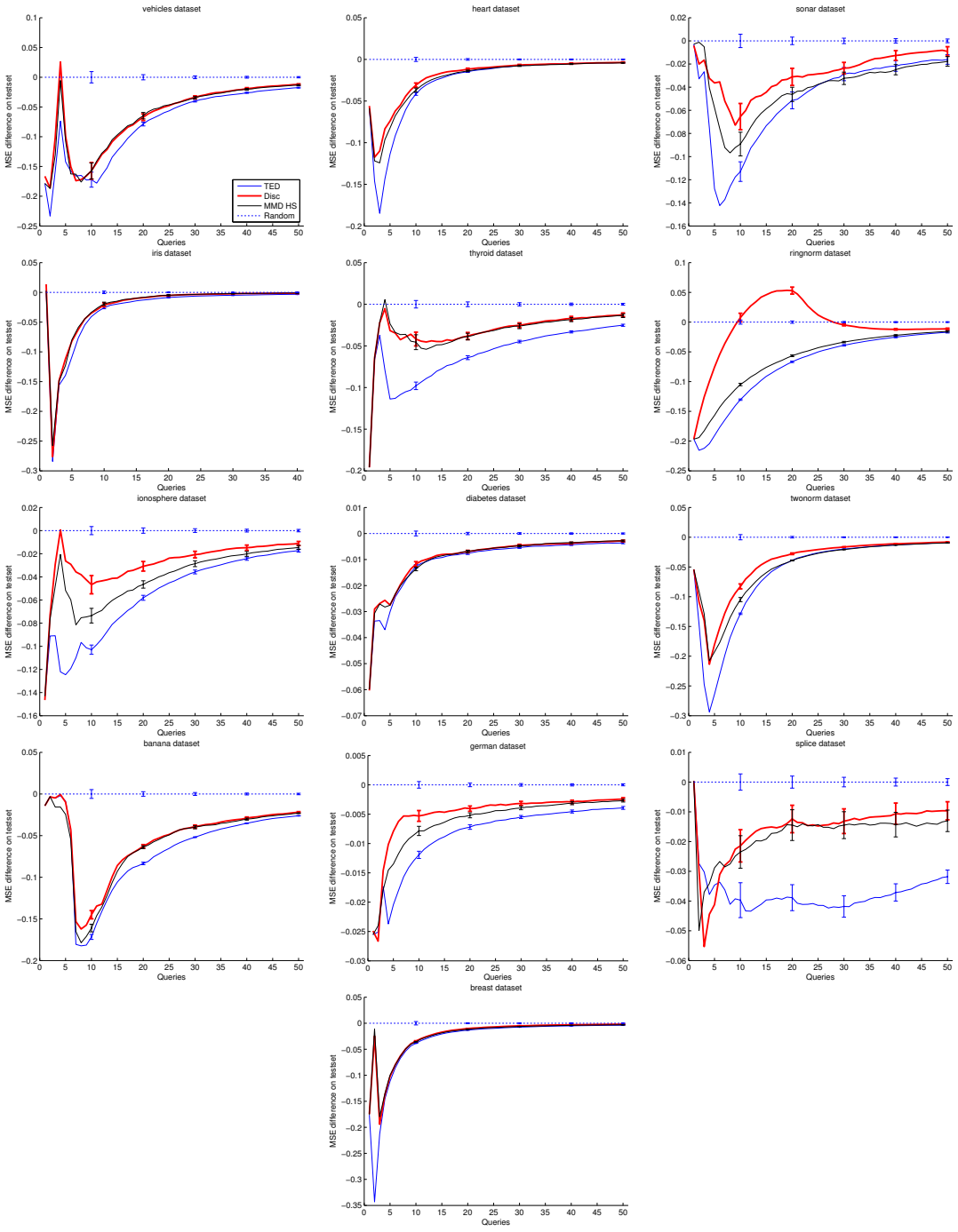


Figure K.3: Results on all benchmark datasets for the realizable case where $f \in H$. TED often performs the best while the discrepancy often performs the worst.

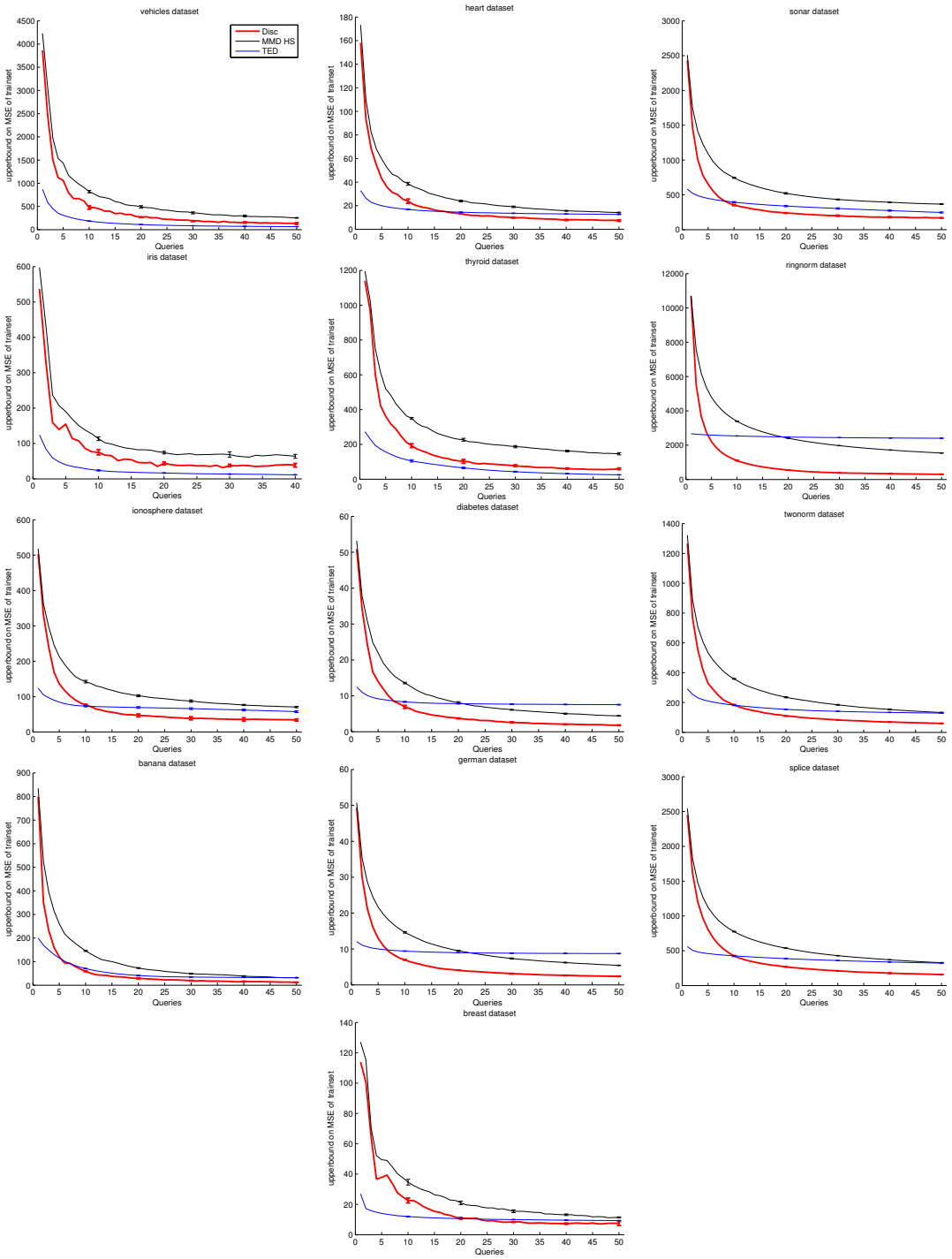


Figure K.4: The values of the generalization bounds on the performance of the trained model during active learning on the set $\hat{P}$ for the realizable setting where $f \in H$. In this setting the bounds are guaranteed to hold.

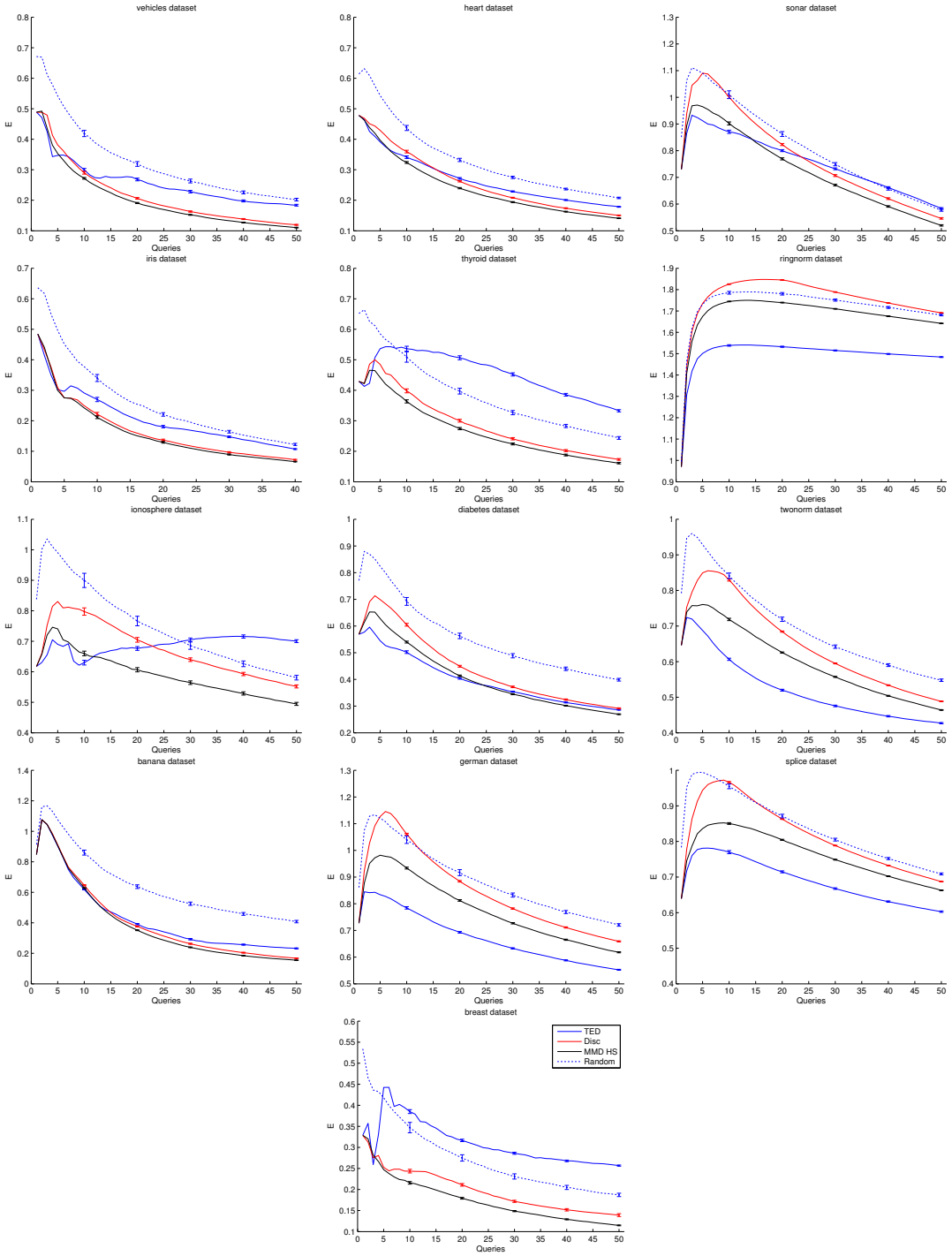


Figure K.5: We plot the quantity E during the active learning experiments. Observe that in some cases the value of E is much larger for the discrepancy than the MMD, and sometimes E of the discrepancy becomes even larger than E of random sampling.

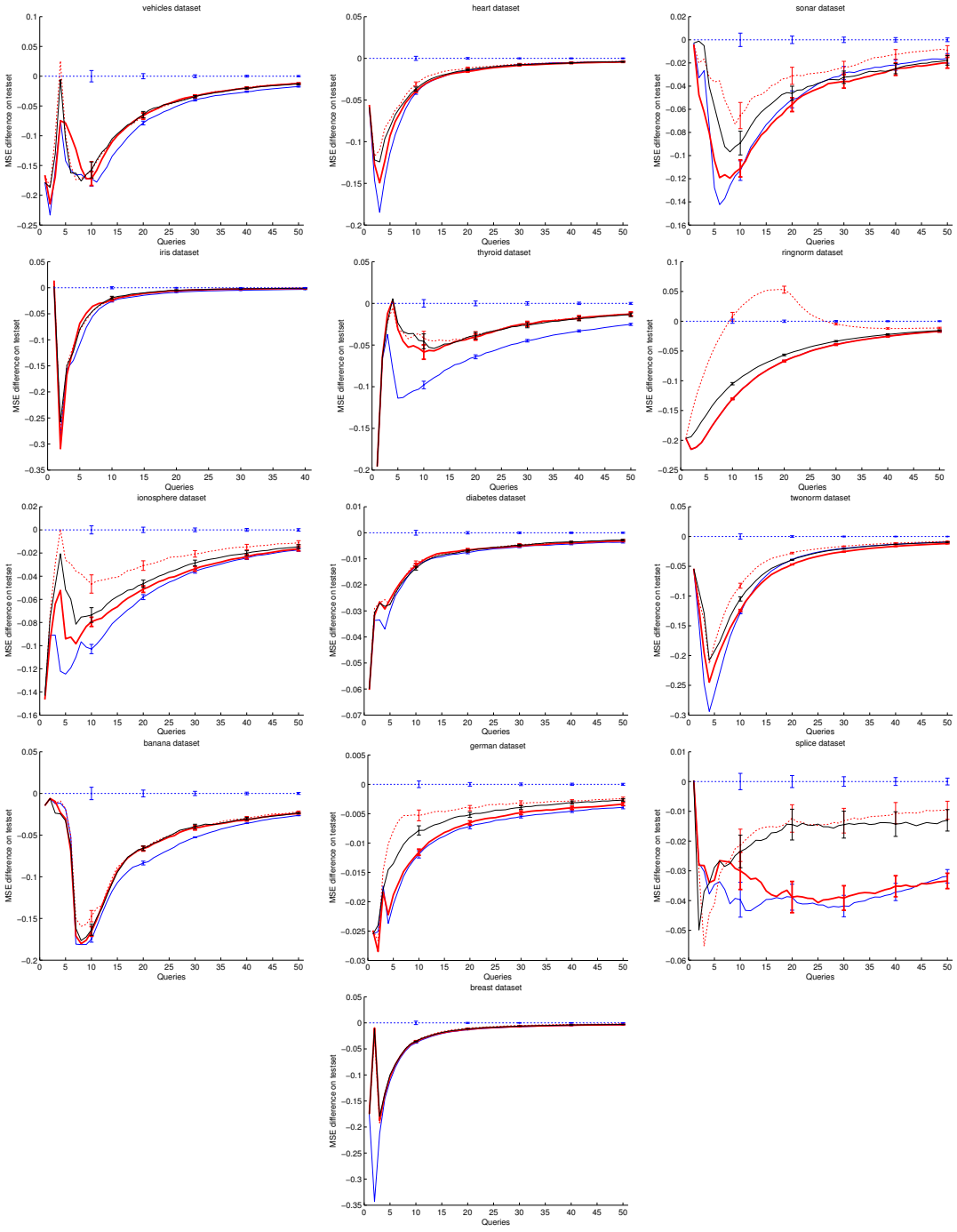


Figure K.6: Comparison of the nuclear discrepancy active learner with the other active learning methods on some of the benchmark datasets for the realizable case where $f \in H$. Observe that the nuclear discrepancy improves a lot upon the discrepancy and sometimes upon the MMD HS active learner. In particular, on the `ringnorm` dataset the nuclear discrepancy performs much better than the discrepancy and matches the performance of TED.

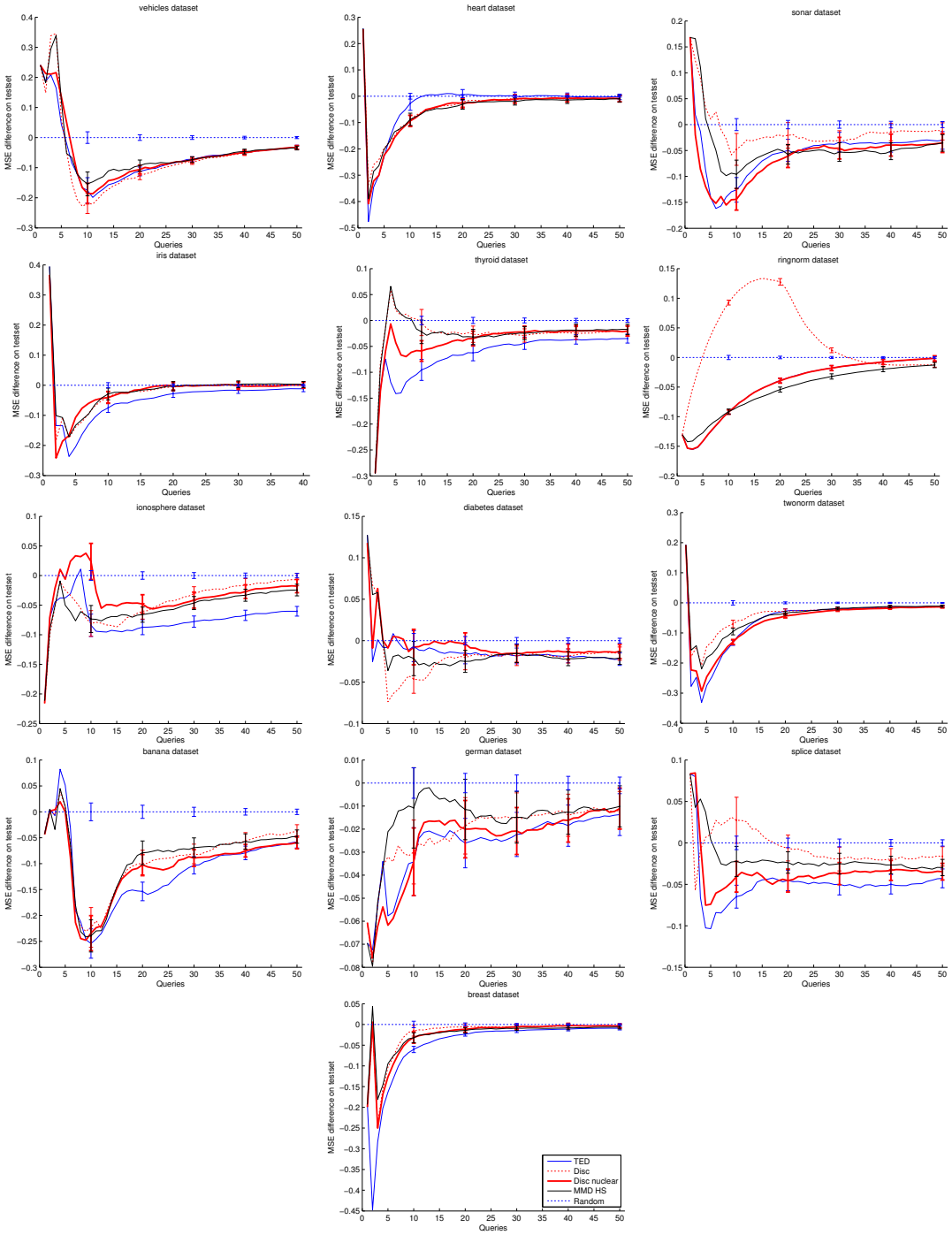


Figure K.7: Some illustrating learning curves of the active learners on real world data in the agnostic setting where $f \notin H$. Observe that the learning curves of TED and the nuclear discrepancy overlap for the `ringnorm` dataset.

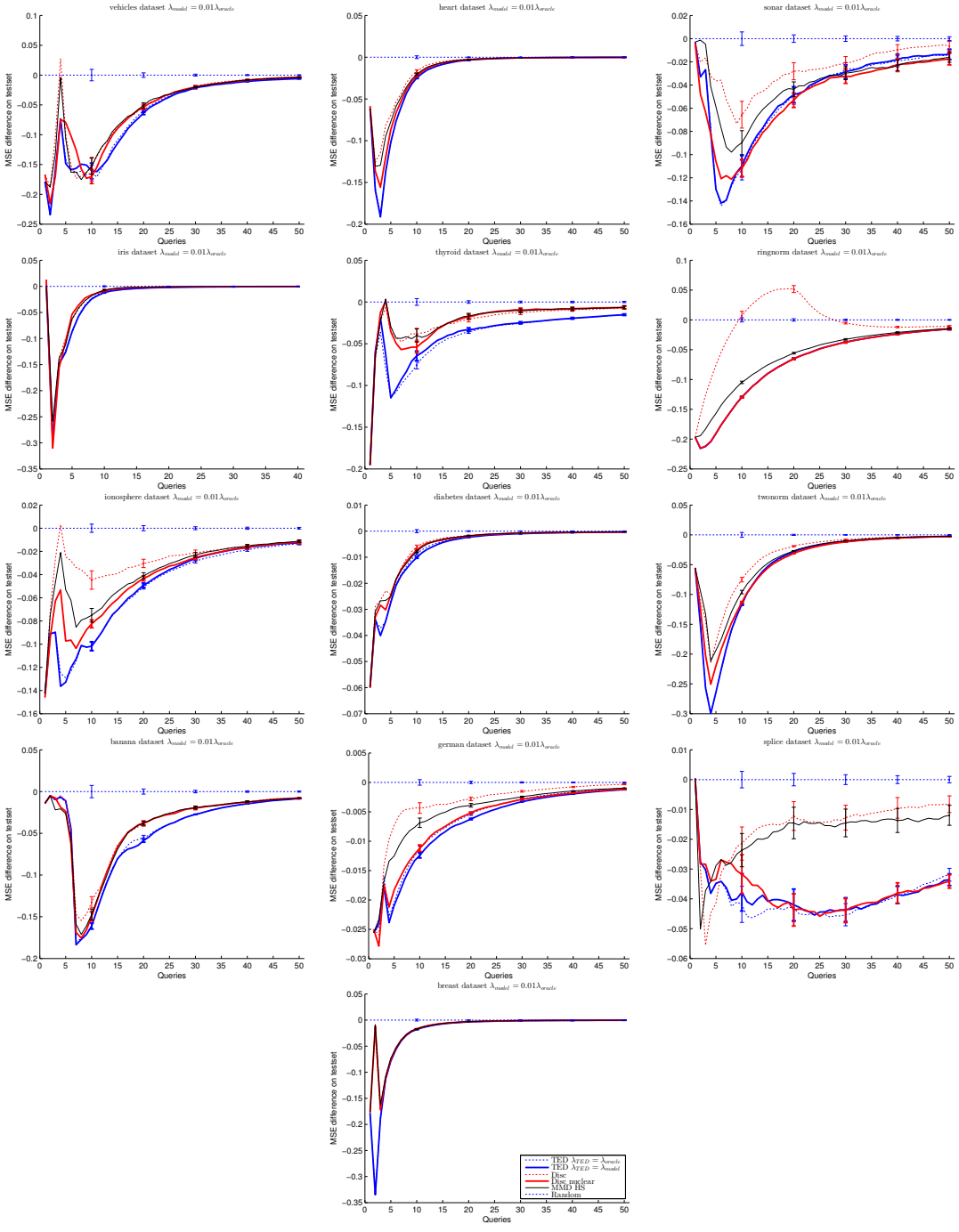


Figure K.8: Setting where there is model mismatch with respect to the regularization parameter. Here we set $\lambda_{\text{model}} = 0.01\lambda_{\text{oracle}}$. In this case we are still in the realizable setting. Observe that the learning curves are very similar to the regular realizable setting.

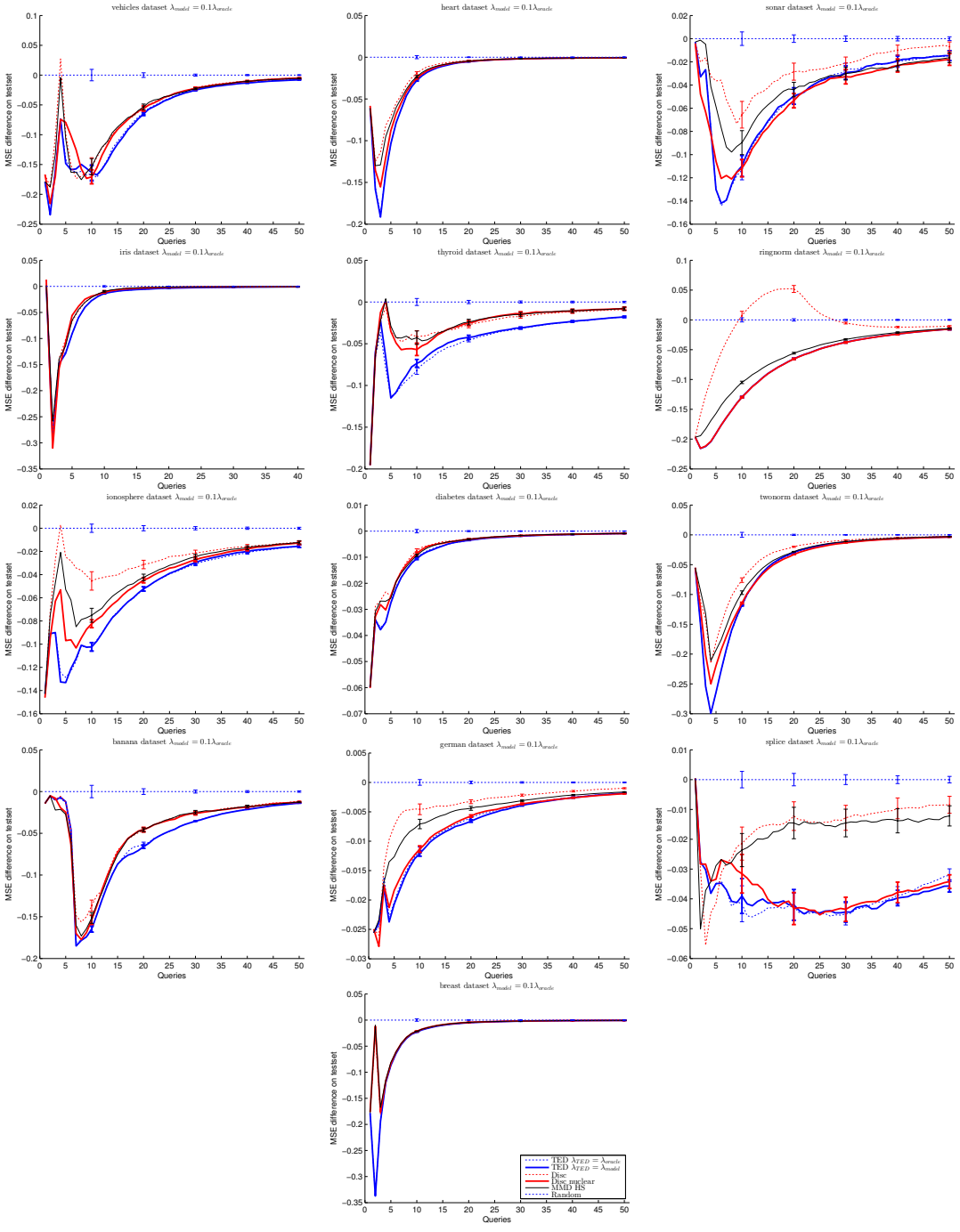


Figure K.9: Setting where there is model mismatch with respect to the regularization parameter. Here we set $\lambda_{\text{model}} = 0.1\lambda_{\text{oracle}}$. In this case we are still in the realizable setting. Observe that the learning curves are very similar to the regular realizable setting.

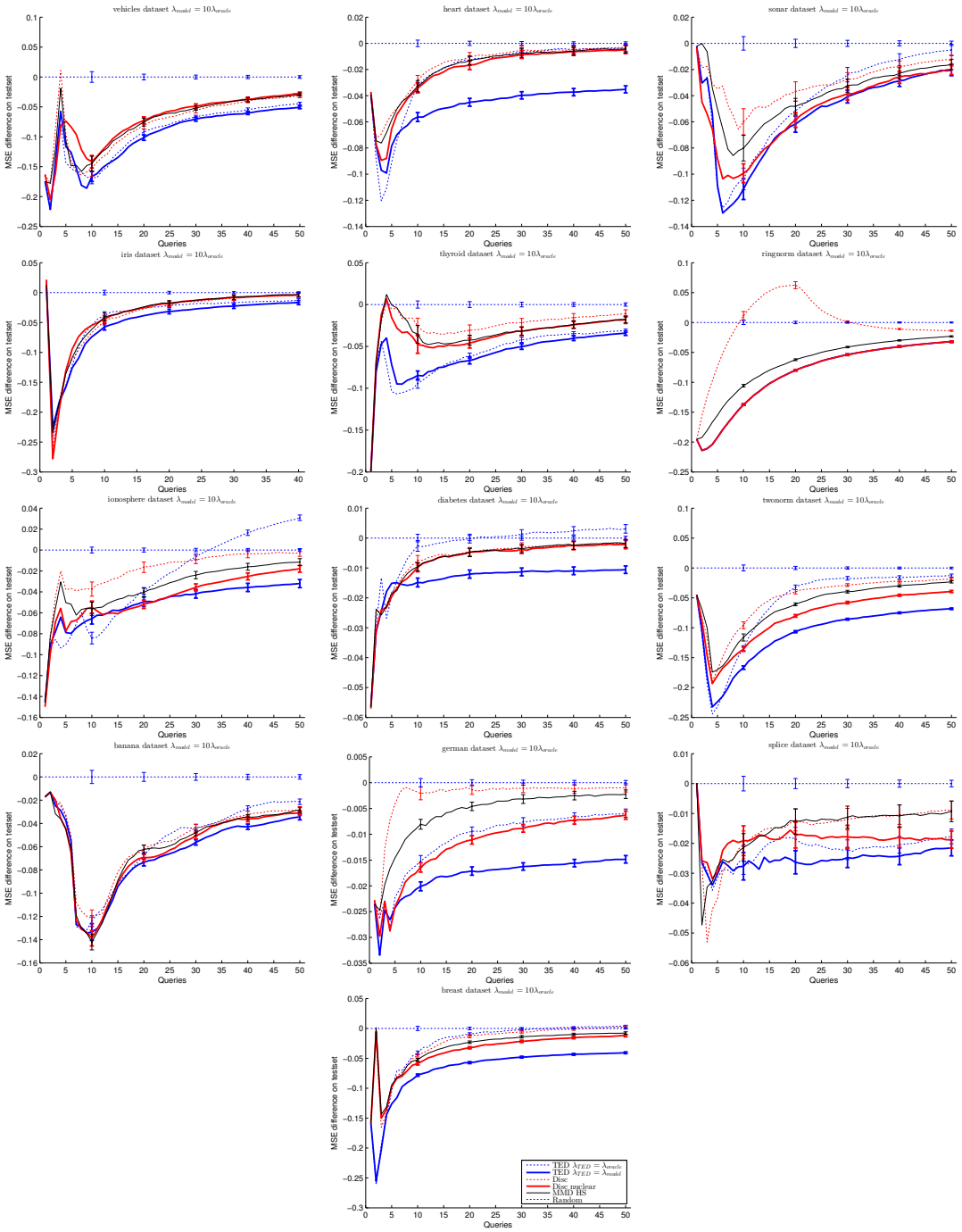


Figure K.10: All learning curves for the setting where there is model mismatch with respect to the regularization parameter. Here we set $\lambda_{\text{model}} = 10\lambda_{\text{oracle}}$. In this case we are not in the realizable setting. Observe that TED with the proper regularization parameter (λ_{model}) generally outperforms all other active learning methods as in the realizable setting. Furthermore, observe that TED with the proper regularization parameter (λ_{model}) performs significantly better than TED with an improper regularization parameter (λ_{oracle}). TED with an improper regularization parameter can even perform worse than random sampling. This shows that TED adapts its active learning strategy in a meaningful way based on the regularization parameter, and suggests that taking the regularization parameter of the model into account is beneficial for active learning.

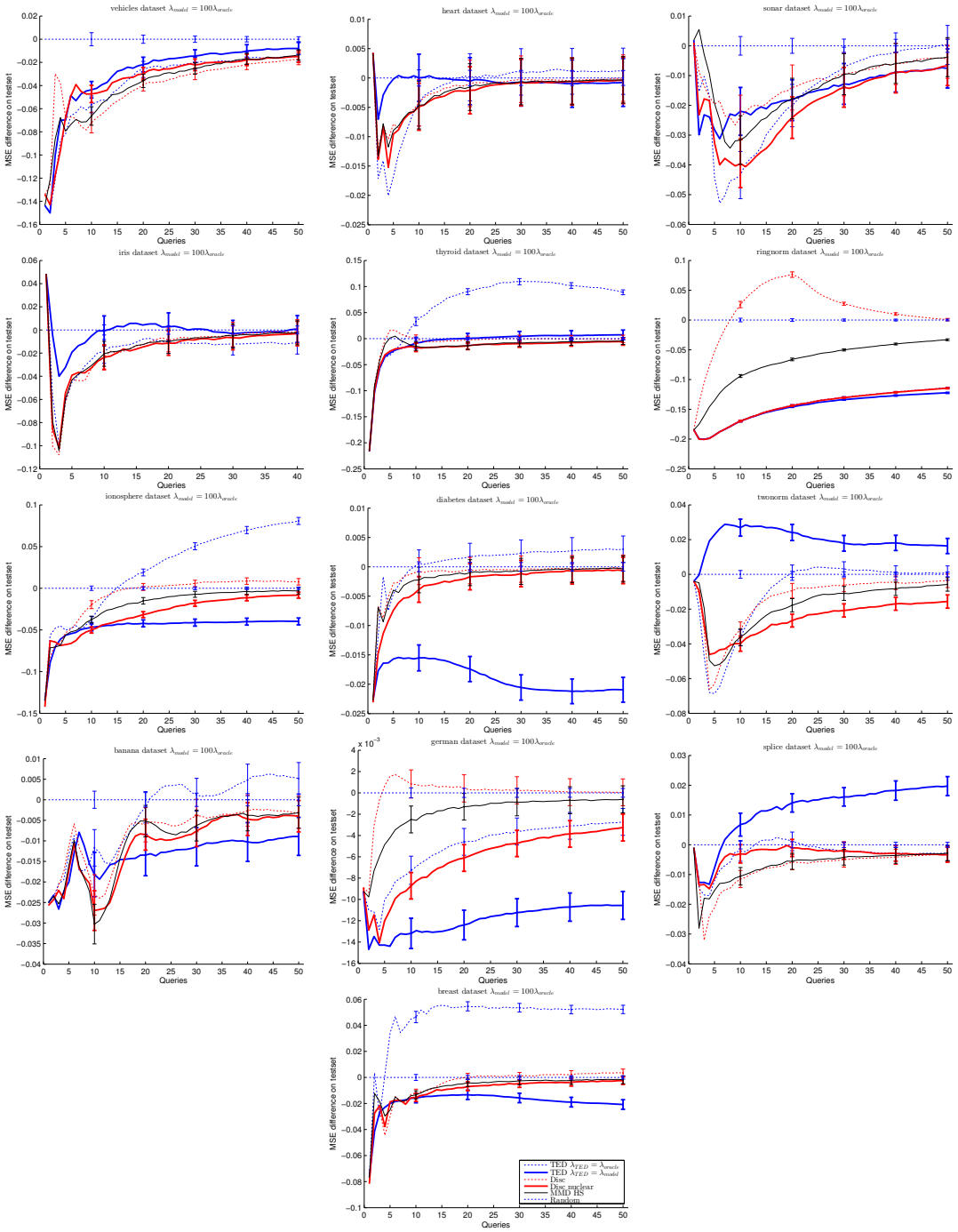


Figure K.11: All learning curves for the setting where there is model mismatch with respect to the regularization parameter. Here we set $\lambda_{\text{model}} = 100\lambda_{\text{oracle}}$. In this case we are not in the realizable setting. Observe that TED with the proper regularization parameter (λ_{model}) sometimes performs better than other methods, but in some cases also performs much worse, even worse than random sampling. This shows that TED cannot deal with a too large model misspecification or large regularization parameter. Observe that the MMD HS and nuclear discrepancy active learners are relatively robust to model misspecification and in all cases perform better than random sampling.

L

Detailed Experimental Settings

In this appendix we give all detailed settings used in all experiments ensuring everything can be reproduced.

First, note that always when computing the ridge regression model its model outputs, we use the Cholesky decomposition instead of computing the inverse matrix $(K_{\hat{Q}\hat{Q}} + \lambda I)$, since computing the inverse explicitly can be inaccurate. The Cholesky decomposition is very stable even when this matrix is ill-conditioned.

First we give additional information how to reproduce the results in Section 4.1. The dataset consists of 300 objects divided over N clusters. The separation between each cluster is 2 units, and samples in each cluster are uniformly distributed with length 1. We use a Gaussian kernel with $\sigma = 0.5$ and set the regularization parameter to 10^{-3}. The labeling function follows a checkerboard pattern. We use a training and test set as described in Chapter 3, and also pre-process the data as described in Chapter 3 to make this data conform to the realizable setting.

Here we give additional information how to reproduce the results in Section 4.2.1. The three fixed samples are located at $(0, 1)$,$(0, 0)$ and $(0, -1)$. The clusters are generated by uniformly distributed objects. The cluster centers are $(0, -2)$ and $(0, 2)$. The uniform distributions have height 0.2 and width 0.2. In Figure 4.3b we display the minimum objective of each method for the considered samples. We used a small regularization parameter of $\lambda = 0.01$ to compute the TED objective. Note that it does not matter which λ is chosen for TED, it always orders the objects in the same way, however if λ is chosen too large TED will have numerical difficulty in choosing the best sample.

Here we gives additional information how to reproduce the results in Section 4.3.1. For simplicity, we set all labels to 1, meaning each dimension is 'equally' important. We could also randomly choose f from H where each f would be equally likely, but ultimately this will be averaged out and we will obtain the same results. When performing experiments, we do not sample from this distribution but simply take this exact dataset, and we evaluate on the same dataset. This way the distribution of the data is always the same, and we can compare the empirical samples selected by TED and the discrepancy with the distribution of the complete dataset (otherwise we could not generate Figure 4.6a). When using a training and test set we will obtain similar results. We set the regularization parameter to a small value of $\lambda = 10^{-5}$. The performance of the random active learner is the mean performance over 1000 runs. Since the dataset is fixed, TED and the discrepancy have deterministic performance, and so these results are not averaged.

Here we give additional information how to reproduce the results in Section 4.4.2. This 2D dataset has 16 Gaussian distributed clusters with means on a grid. The distance between the means of two neighboring clusters is always 2 in the horizontal or vertical dimension. Each Gaussian has a covariance matrix of $0.07I$ and contains 40 objects. We use a Gaussian kernel with $\sigma = 0.1$ so the labels of one cluster hardly influence the labels in other clusters similar to the linear dataset in the previous subsection. We use $\lambda = 0.01$. For the MMD active learner we use a value of $\sigma = \frac{0.01}{\sqrt{2}}$ to

take the hypothesis set into account. The labels in this dataset were generated by a checkerboard pattern. A ridge regression model with above parameters was fitted to the complete dataset, and the outputs of the model were used as labels in the experiment. This assures that we are in the realizable setting where $f \in H$, and thus the assumptions of the bounds are satisfied. We split the data in a training set (65%) and test set (35%). The initial labeled set is given by 30 objects in the bottom left corner. We repeat the experiment 100 times.

Here we give additional information how to reproduce the results in Section 4.6.1. We use the same dataset and settings as described for Subsection 4.3.1, the differences are described below. We generate the labels by $f = 1 + \epsilon$, where ϵ is zero mean Gaussian noise, with $\sigma = 1.5$. We performed this experiment with $\lambda = 10^{-2.5}$, repeating the experiments 1000 times for each active learner. We again always use the full dataset (no train / test split). We however generate 'fresh' labels to evaluate the active learner, so we can accurately estimate the generalization error. Even though the TED and discrepancy active learner choose deterministic samples, since the labels are noisy, each experiment needs to be repeated multiple times. For higher values of σ the result are similar. For $\sigma < 1$ the effect of the noise is too small, and the performance of TED becomes better. For larger values of λ the difference between both methods will become harder to observe, since TED will also select samples more according to their proportions as was observed in the Subsection 4.3.1.

For any experiment using the benchmark datasets (experiments of sections 4.2.2, 4.3.2, 4.6.2 and 4.5) the parameter settings used are displayed in Table L.1. The experimental setup is further completely described in Chapter 3 for both the realizable and agnostic setting. Finally, note that for all experiments concerning the dataset `banana` we used only 50 repeats instead of 100 due to excessive computational times.

Dataset	σ	$\log_{10}(\lambda)$
`vehicles`	5.270	-3.0
`heart`	5.906	-1.8
`sonar`	7.084	-2.6
`iris`	2.313	-2.2
`thyroid`	1.720	-2.6
`ringnorm`	1.778	-3.0
`ionosphere`	4.655	-2.2
`diabetes`	2.955	-1.4
`twonorm`	5.299	-2.2
`banana`	0.645	-2.2
`german`	4.217	-1.4
`splice`	9.481	-2.6
`breast`	4.217	-1.8

Table L.1: Table with parameters used for the benchmark datasets