



Nuclear discrepancy for single-shot batch active learning

Tom J. Viering¹ · Jesse H. Krijthe² · Marco Loog¹

Received: 17 September 2018 / Revised: 19 March 2019 / Accepted: 6 June 2019 / Published online: 26 June 2019
© The Author(s) 2019

Abstract

Active learning algorithms propose what data should be labeled given a pool of unlabeled data. Instead of selecting randomly what data to annotate, active learning strategies aim to select data so as to get a good predictive model with as little labeled samples as possible. Single-shot batch active learners select all samples to be labeled in a single step, before any labels are observed. We study single-shot active learners that minimize generalization bounds to select a representative sample, such as the maximum mean discrepancy (MMD) active learner. We prove that a related bound, the discrepancy, provides a tighter worst-case bound. We study these bounds probabilistically, which inspires us to introduce a novel bound, the nuclear discrepancy (ND). The ND bound is tighter for the expected loss under optimistic probabilistic assumptions. Our experiments show that the MMD active learner performs better than the discrepancy in terms of the mean squared error, indicating that tighter worst case bounds do not imply better active learning performance. The proposed active learner improves significantly upon the MMD and discrepancy in the realizable setting and a similar trend is observed in the agnostic setting, showing the benefits of a probabilistic approach to active learning. Our study highlights that assumptions underlying generalization bounds can be equally important as bound-tightness, when it comes to active learning performance. Code for reproducing our experimental results can be found at <https://github.com/tomviering/NuclearDiscrepancy>.

Keywords Active learning · Maximum mean discrepancy · Discrepancy · Kernel methods

Editors: Karsten Borgwardt, Po-Ling Loh, Evimaria Terzi, Antti Ukkonen.

✉ Tom J. Viering
t.j.viering@gmail.com
<http://tomviering.nl>

Jesse H. Krijthe
jkrijthe@gmail.com

Marco Loog
m.loog@tudelft.nl

¹ Pattern Recognition Laboratory, Mekelweg 5, 2628 CD Delft, The Netherlands

² Institute for Computing and Information Sciences, Radboud University Nijmegen, Toernooiveld 212, 6525 EC Nijmegen, The Netherlands

1 Introduction

Supervised machine learning models require enough labeled data to obtain good generalization performance. For many practical applications such as medical diagnosis or video topic prediction labeling data can be expensive or time consuming (Settles 2012). Often in these settings unlabeled data is abundant. In active learning an algorithm chooses unlabeled samples for labeling (Cohn et al. 1994). The idea is that models can perform better with less labeled data if the labeled data is chosen carefully instead of randomly. Active learning makes the most of a small labeling budget and can reduce labeling costs.

Several works use upperbounds on the expected loss to motivate particular active learning strategies (Gu and Han 2012; Ganti and Gray 2012; Gu et al. 2012, 2014; Wang and Ye 2013). We study pool-based active learners that choose queries that explicitly minimize generalization bounds and investigate the relation between bounds and active learning performance. We evaluate generalization with respect to the surrogate loss in the classification setting and use the kernel regularized least squares model (Rifkin et al. 2003), a popular model in active learning (Huang et al. 2010; Wang and Ye 2013). Our focus is on active learners that select a batch of queries in a single shot (Contardo et al. 2017). This means that there is no label information available at the time the batch of queries is determined. Since the active learners have only unlabeled data at their disposal they aim to select the most representative subset of the unlabeled pool. This is different from batch mode or sequential active learning, where after requesting labels from the oracle the algorithm has to determine new queries, creating a feedback loop. The advantage of zero-shot active learning is that all queries can be computed ahead of time, and collected labels do not have to be fed into the active learner.

For applications this can be very convenient: it simplifies the annotation setup. Furthermore, active learning algorithm may require substantial amounts of time to compute the next query. In situations where annotation have to be done by domain experts whose time is costly this can be impractical. For example, if we were to apply active learning to to the problem of Esteva et al. (2017), who build a deep learning model to classify skin cancer, sequential or batch mode active learning strategies usually train a model as intermediate step before being able to determine the next query. For deep models this could take several hours. With zero-shot active learning the dermatologist can annotate all queries without waiting once.

Another example where requesting labels is costly is personalized machine learning models such as for movie recommendation. Here applications may ask feedback from end-users to improve their service. This problem can also be studied using the active learning framework (Harpale and Yang 2008). Asking end-users for feedback usually interrupts their activity in the application. Therefore, we may only interrupt the user a limited amount of times. Using zero-shot active learning users only have to be interrupted once and can answer multiple queries without waiting for new queries to be determined.

The Maximum Mean Discrepancy (MMD) is used for batch-mode active learning by Chattopadhyay et al. (2012) to match the marginal distribution of the selected samples to the marginal distribution of all unlabeled samples. This active learner has been shown to minimize a generalization bound (Wang and Ye 2013). The MMD is a divergence measure (Gretton et al. 2012) which is closely related to the Discrepancy divergence measure of Mansour et al. (2009), both have been used in domain adaptation (Huang et al. 2007; Cortes and Mohri 2014).

Using the Discrepancy, we show that we can get a tighter worst case generalization bound than the MMD in the realizable setting. Tighter bounds are generally considered better as

they estimate the expected loss more accurately. One might therefore expect the Discrepancy to lead to better queries in active learning.

We show, however, that the Discrepancy and MMD generalization bounds can be derived, using a probabilistic analysis, from pessimistic assumptions. We subsequently apply the principle of maximum entropy to derive probabilistic assumptions that are more optimistic, inspiring us to introduce the Nuclear Discrepancy (ND) bound. Under these optimistic assumptions the ND provides a tighter bound on the expected loss than the MMD, while the Discrepancy bound is the loosest.

We compare the active learning performance of the proposed ND bound to the existing MMD and Discrepancy bounds. Our hypothesis is that we often find ourselves in a more optimistic average-case scenario than a worst-case scenarios. To this end we empirically study the behavior of the active learners on 13 datasets, and we investigate whether probabilistic assumptions or worst-case assumptions better model observed behavior in our experiments.

In the realizable setting a model from the model class can perfectly predict the groundtruth labels, as in this setting there is no model mismatch or model misspecification. For this we show that the tightness relations between the generalization bounds is strict. As such, for the realizable case, our theory gives the strongest predictions for the ranking of the active learners in terms of performance. In the agnostic case, where no such model may exist, the tightness relations can change, which renders our theory less applicable. We perform experiments in both settings to see the effect of the theoretical assumptions not being fulfilled.

We study the realizable setting since it is more amendable to theoretical analysis. This setting is often studied in active learning and is still a topic of active investigation (Tosh and Dasgupta 2017). The general case of the agnostic case is much harder to analyze. To illustrate this, we remark that it has been observed that if a model class is sufficiently wrongly chosen, active learning can even decrease model performance (Settles 2011; Attenberg and Provost 2011; Loog and Yang 2016; Yang and Loog 2018).

These counter-intuitive behaviors further underline the need for further theoretical studies. We believe that by improving our understanding of simpler active learning settings (realizable case) will contribute to improved understanding of more difficult active learning settings (agnostic case).

To this end, our study provides new quantitative tightness relations between the MMD, Discrepancy and ND bound under different probabilistic assumptions. We investigate the connection between bound tightness and active learning performance. Our most important conclusion is that not only bound tightness is important for performance, but that appropriate assumptions are equally important.

1.1 Overview and contributions

First we discuss related work in Sect. 2. In Sect. 3 we describe the considered active learning setting and notation. We present our theoretical results regarding the MMD and Discrepancy in Sect. 4. In Sect. 5 we motivate our novel Nuclear Discrepancy bound. We evaluate the proposed active learners experimentally in Sect. 6. In Sect. 7 we give a discussion and in Sect. 8 we give the conclusions of this work. All proofs, additional background theory and experimental results are given in the Appendix. The main contributions of this work are:

1. An improved MMD bound for active learning and a more informed way to choose the kernel of the MMD in the context of learning.
2. A proof that the Discrepancy bound on the worst case loss is tighter than the MMD bound.
3. A probabilistic interpretation of the MMD bound.

Table 1 Visual summary of our work

Bound	Probabilistic Assumption			Experiments
	Worst-Case Section 4.3	Pessimistic-Case Section 4.4	Average-Case Section 5	Performance Section 6
Discrepancy	Tightest	Loosest	Loosest	Worst
MMD	Intermediate	Tightest	Intermediate	Intermediate
Nuclear Discrepancy (proposed)	Loosest	Intermediate	Tightest	Best

This table gives an overview of the newly proven tightness relations between the generalization bounds and the experimental results. Observe that the tightness relations under the ‘Average-Case’ correlate well with the experimental performance of the active learners. Therefore, we stipulate that the ‘Average-Case’ is the most accurate assumption for our considered active learning setting. Note that the tightness relations only hold under the conditions of Theorem 2, and that the experimental performance shown here best reflect the performance in the realizable setting. In the agnostic setting the ranking of the active learning methods is less clear, but the same trend is observed

4. The Nuclear Discrepancy (ND) bound that provides the tightest bound on the expected loss under probabilistic assumptions that follow from the principle of maximum entropy.
5. A probabilistic analysis that explains the differences in empirical performance (in terms of the mean squared error) achieved by the active learners.

In Table 1 we give a visual summary of our work. It shows all formal results and shows in which sections to find them. It also shows the relation between the theory and experiments, and the main findings of the experiments.

2 Related work

Many active learning methods have been proposed, Settles (2012) provides an excellent introduction and overview. Our work is related to active learning methods that select representative samples (Xu et al. 2003). Most active learning strategies of this kind are combined with an uncertainty criteria (Xu et al. 2003; Chattopadhyay et al. 2012; Wang and Ye 2013; Huang et al. 2010), and often the representative component is used to diversify queries when chosen in batches in order to avoid redundancy (Xu et al. 2003; Wang and Ye 2013). This is different from our considered setting: since there is no labeled data and we have to choose all queries in one shot, our only option is to select representative samples, since uncertainty criteria can only be computed if some labels are known.

A closely related well-known concept to our work is that of (Transductive or) Optimal Experimental Design (Yu et al. 2006). Here also no labeled data is required to select queries for the case of the linear regression model. These methods aim to minimize some form of posterior variance of the model. A closely related statistical approach relies on maximization of the Fisher Information to reduce model uncertainty (Hoi et al. 2006). However, for these approaches it is often required to explicitly specify a noise model (such as Gaussian i.i.d. noise), while in this work we consider deterministic labeling functions.

Our work is motivated by several active learners that minimize generalization bounds. Gu and Han (2012) uses the Transductive Rademacher Complexity generalization bound to perform active learning on graphs. Gu et al. (2012) show that the strategy of Yu et al. (2006) also minimizes a generalization bound, and extend the method to work with a semi-

supervised model. Ganti and Gray (2012) introduce an active learning strategy that uses importance weighting to ensure asymptotic consistency of the actively learned model. Their strategy minimizes a generalization bound for the squared loss under some conditions on the data distribution. Gu et al. (2014) introduce a strategy that minimizes a generalization bound on the risk for logistic regression. Wang and Ye (2013) also use a generalization bound based on the MMD to perform active learning, but we will describe this work later in more detail when discussing all methods that use the MMD.

Many theoretical active learning works motivate algorithms by generalization bounds, for example one of the first active learning algorithms ‘CAL’ (Cohn et al. 1994) and its agnostic generalization A^2 (Balcan et al. 2009) have been thoroughly analyzed using generalization bounds by making use of the Disagreement Coefficient (Hanneke 2007). Most of these theoretical works consider worst-case performance guarantees, where the distribution is chosen by an adversary subject to constraints. Balcan and Urner (2016) provides a short and concise overview of these and other recent theoretical active learning works. In contrast with our work, these algorithms consider generalization in terms of zero-one loss instead of squared loss and do not apply to one shot active learning.

A straightforward approach to one shot active learning is through clustering: cluster the data and request the labels of the cluster centers (Bod et al. 2011; Hu et al. 2010; Zhu et al. 2008; Nguyen and Smeulders 2004). However, unlike our work, these methods are not motivated by generalization bounds. Obtaining bounds for such approaches may be difficult because the clustering algorithm and machine learning model may rely on different assumptions. To still get bounds one can use the clustering algorithm instead to also provide predictions for new samples (Urner et al. 2013). Instead, we stick to the regularized least squares model and use the MMD and Discrepancy to get bounds for this model. Our approach can be used to derive bounds and corresponding active learning strategies for any kernelized L_2 regularized model, however, in this work we only focus on the squared loss.

Our work is closely related to that of Chattopadhyay et al. (2012): we use a greedy version of their proposed active learning algorithm. Chattopadhyay et al. (2012) are the first to use the MMD for active learning in a batch-mode setting. An in-depth empirical analysis shows that the MMD outperforms other active learning criteria as judged by the zero-one error when used with kernelized SVMs. They show that the MMD easily can be combined with uncertainty-based active learning approaches and transfer learning. Since we consider one-shot active learning we don’t consider the uncertainty-based component of their algorithm. In follow up work active learning and transfer learning is solved jointly using the MMD (Chattopadhyay et al. 2013).

Our theoretical analysis of the MMD bound extends the analysis of Wang and Ye (2013). Wang and Ye (2013) show that active learning by minimization of the MMD and the empirical risk can be seen as minimizing a generalization bound on the true risk. They introduce an active learner that balances exploration (distribution matching using MMD) with exploitation (a form of uncertainty sampling). They show empirically that their proposed algorithm is competitive with several other active learning strategies as evaluated by the zero-one error using kernelized SVMs.

We build upon the generalization bound of Wang and Ye (2013) and improve it. Their bound considers the underlying distribution of the unlabeled pool and labeled (queried) sample, however, this is problematic because the labeled sample is non-i.i.d. due to dependence of the queries of the active learner. We resolve this issue and introduce an additional term η that measures the error of approximating the worst-case loss function.

Mansour et al. (2009) introduce the Discrepancy generalization bound for domain adaptation with general loss functions. In a follow up work, Cortes and Mohri (2014) contrast the

Discrepancy with the MMD generalization bound: they argue that the Discrepancy is favorable from a theoretical point of view because it takes the loss function and hypothesis set of the model into account, while the MMD does not. This means that the MMD bound for an SVM and regularized least squares model would be exactly the same, while the Discrepancy bound specializes to the chosen model and surrogate loss. They derive an efficient domain adaptation algorithm and empirically show that the Discrepancy improves upon the MMD in several regression adaptation tasks.

Prior to our work, the Discrepancy measure (Cortes and Mohri 2014) has not yet been used to perform active learning. We show that by choosing the kernel for the MMD carefully, we can adapt the MMD to take the hypothesis set and loss into account, addressing one of the theoretical limitations of the MMD identified by Cortes and Mohri (2014). Under these conditions we find that we can compare the MMD and Discrepancy bounds in terms of tightness. This quantitative comparison of these bounds is novel and was not considered before.

Germain et al. (2013) adapt the Discrepancy for the zero-one loss to a PAC-Bayes setting in order to do domain adaptation. Their analysis is specifically for the zero-one loss, while we consider the squared loss. Their PAC-Bayes framework is significantly different from our analysis: instead of minimizing a surrogate loss, they use a Gibbs classifier, and they minimize bounds on the expected risk directly. This involves a non-convex optimization problem. Instead, we simply minimize the empirical risk and consider deterministic models, similar to most PAC style analysis. This makes our analysis simpler. Furthermore, they propose a framework to jointly minimize the empirical risk and domain divergence. To this end, their algorithm requires labeled data which is unavailable in zero-shot active learning, making it unsuitable for our zero-shot setting.

In Cortes et al. (2019) a new domain adaptation algorithm based on a new divergence measure, the Generalized Discrepancy, is introduced. The algorithm consists of two stages: first it minimizes the Discrepancy, afterward it minimizes the empirical risk and the Generalized Discrepancy jointly. The strategy of Cortes et al. (2019) is difficult to apply to active learning for two reasons. First of all, their algorithm requires labeled data to minimize the empirical risk and the General Discrepancy jointly, which is impossible in our zero-shot active learning setting. Second, their algorithm requires i.i.d. samples from the unlabeled pool to estimate the hyperparameter r . This would require costly random queries in the active learning setting. Because of these reasons, we believe their algorithm is more suitable to a joint active and domain adaptation setting (such as considered by Chattopadhyay et al. (2013)) where more labeled data is available.

Our theoretical analysis is substantially different from the analysis of Cortes et al. (2019). Because Cortes et al. (2019) use labeled data, they can make a more accurate characterization of possible worst case scenario's, refining the worst-case scenario of the Discrepancy to obtain tighter bounds. We take an orthogonal approach: we consider probabilistic generalization bounds that hold in expectation. Instead of considering a worst-case, we make probabilistic assumptions to get to a plausible average-case. Cortes et al. (2019) compare the Generalized Discrepancy and Discrepancy bounds in terms of tightness. We compare the tightness of the bounds of the MMD, Discrepancy and Nuclear Discrepancy. We show several orderings of the tightness of the bounds under different probabilistic assumptions, while Cortes et al. (2019) only takes a worst-case approach.

In summary, our work differs from previous works by considering instead of worst-case analysis (Cortes et al. 2019; Cortes and Mohri 2014), a probabilistic analysis of generalization bounds. Unlike most other works that use generalization bounds for domain adaptation (Cortes et al. 2019; Cortes and Mohri 2014; Germain et al. 2013), we use bounds to perform

active learning. For the MMD active learner, studied by Chattopadhyay et al. (2012); Wang and Ye (2013), we give new theoretical results: an improved bound for active learning and we provide a principled way to choose the kernel for the MMD. We give new quantitative comparisons of bound tightness for the MMD and Discrepancy in multiple settings, while before these bounds were compared only qualitatively (Cortes and Mohri 2014). Furthermore, we study the novel question: how does bound tightness relate to active learning performance?

3 Setting and notation

Let $\mathcal{X} = \mathbb{R}^d$ denote the input space and \mathcal{Y} the output space. Like Cortes and Mohri (2014) we assume there is a function $f : \mathcal{X} \rightarrow \mathcal{Y}$ that determines the outputs and there is an unknown distribution with density P over \mathcal{X} from which we get an independent and identically distributed (i.i.d.) unlabeled sample $\hat{P} = (x'_1, \dots, x'_{n_{\hat{P}}}) \in \mathcal{X}^{n_{\hat{P}}}$. We study single-shot batch active learners that given the unlabeled pool \hat{P} selects a batch $\hat{Q}_n \subset \hat{P}$ of n samples before observing any labels. The active learner submits the batch to the labeling oracle that provides the labels of the batch. A kernel regularized least squares (KRLS) model is trained on \hat{Q}_n^{lab} , where lab indicates a labeled dataset.

We take the kernel of the model K to be positive semi-definite (PSD), and denote the reproducing kernel Hilbert space (RKHS) as \mathcal{H} where $\|h\|_K$ denotes the norm in \mathcal{H} . A model corresponds to $h \in \mathcal{H}$ and is obtained by minimizing

$$L_{\hat{Q}}(h, f) + \mu \|h\|_K^2$$

for $h \in \mathcal{H}$ when trained on \hat{Q}^{lab} , where we follow the convention of Cortes and Mohri (2014). $L_{\hat{Q}}(h, f)$ is the average empirical loss of h on \hat{Q} with outputs given by f :

$$L_{\hat{Q}}(h, f) = \frac{1}{n_{\hat{Q}}} \sum_{x \in \hat{Q}} l(h(x), f(x)),$$

where $l : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ is a loss function. For KRLS l is the squared loss: $l(h(x), f(x)) = (h(x) - f(x))^2$, then $L_{\hat{Q}}(h, f)$ is the mean squared error (MSE) on \hat{Q} . Model complexity is controlled by the regularization parameter $\mu > 0$. We choose

$$H = \left\{ h \in \mathcal{H} : \|h\|_K \leq \Lambda = \frac{f_{\max}}{\sqrt{\mu}} \right\}$$

as our hypothesis set where $f_{\max} = \sup_{x \in \mathcal{X}} |f(x)|$. Training KRLS always leads to a solution $h \in H$ (Mohri et al. 2012, Lemma 11.1).

In classification typically we are interested in the zero-one error (accuracy), however, our study focuses on the squared loss (the surrogate loss). We use the squared loss because we can relate the bounds of the MMD, Nuclear Discrepancy and Discrepancy in closed form and compare them quantitatively. Since our goal is to investigate the correlation between bound tightness and performance, this is essential to our study.

We have made the standard assumption that the data comes from an unknown distribution P . The goal of the active learner is to choose a batch of queries in such a way as to minimize the expected loss of the model under this distribution P :

$$L_P(h, f) = \int_{\mathcal{X}} (h(x) - f(x))^2 P(x) dx. \quad (1)$$

Ideally we would want to train our model on \hat{P}^{lab} , since small $L_{\hat{P}}(h, f)$ will lead to small $L_P(h, f)$ if the model complexity is appropriate, as illustrated by the following theorem (Mohri et al. 2012, p. 240).

Theorem 1 (Generalization bound Squared Loss (Mohri et al. 2012)) *Let l be the squared loss. For any $\delta > 0$, with probability at least $1 - \delta$ over an i.i.d. sample \hat{P} of size $n_{\hat{P}}$ from P , the following inequality holds for all $h \in H$:*

$$L_P(h, f) \leq L_{\hat{P}}(h, f) + 4MR_m(H) + M^2 \frac{\log(\frac{1}{\delta})}{2n_{\hat{P}}} \tag{2}$$

Here $R_m(H)$ is the Rademacher complexity of the hypothesis set H , and M is a constant such that $|h(x) - f(x)| \leq M$ for all $x \in \mathcal{X}$ and all $h \in H$.

If the model complexity is appropriate $R_m(H)$ will be small. The third term is small when the pool \hat{P} is large. If both of these criteria are met, it is unlikely that we overfit as reflected by a tight bound. Then training on \hat{P}^{lab} will likely minimize $L_P(h, f)$.

Ideally we would train on \hat{P}^{lab} , however, since we only have access to the unlabeled sample \hat{P} this is impossible. Therefore we upperbound $L_{\hat{P}}(h, f)$ instead. This upperbound is minimized by the active learners. The studied bounds are of the form

$$L_{\hat{P}}(h, f) \leq L_{\hat{Q}}(h, f) + \text{obj}(\hat{P}, \hat{Q}) + \eta.$$

Due to training $L_{\hat{Q}}(h, f)$ will be relatively small. The term η is a constant that cannot be minimized during active learning since it depends on \hat{P}^{lab} . However, if the model misspecification is small, η will be small. Therefore we ignore this term during active learning, this is also (sometimes implicitly) done in other works (Huang et al. 2007; Chattopadhyay et al. 2012; Cortes and Mohri 2014). Thus the active learners choose the batch \hat{Q} to minimize $\text{obj}(\hat{P}, \hat{Q})$. This objective can be the MMD, disc or disc_N which will be introduced in the next sections. This term measures the similarity between the unlabeled pool \hat{P} and the batch \hat{Q} . Minimizing it leads to selecting a representative sample.

We consider two settings. In the agnostic setting binary labels are used, i.e., $\mathcal{Y} = \{-1, +1\}$, and generally we have $f \notin H$. In the realizable setting $f \in H$, so a model of our hypothesis set can perfectly reproduce the labels as there is no model misspecification. In this case \mathcal{Y} is a subset of \mathbb{R} . In the realizable setting η can become zero under some conditions, which allows us to compare the tightness of the bounds and enables our probabilistic analysis.

$K(x, x')$ indicates the kernel function between x and x' . We mainly use the Gaussian kernel $K(x, x') = \exp(-\|x - x'\|_2^2 / (2\sigma^2))$ where σ , the bandwidth, is a hyperparameter of the kernel. For the MMD we require a second PSD kernel, $K_{\mathcal{L}}$. We indicate its RKHS and bandwidth (for a Gaussian kernel) by $\mathcal{H}_{\mathcal{L}}$ and $\sigma_{\mathcal{L}}$, respectively. All vectors are column vectors. $X_{\hat{P}}$ and $X_{\hat{Q}}$ are the $n_{\hat{P}} \times d$ and $n_{\hat{Q}} \times d$ matrices of the sets \hat{P} and \hat{Q} .

4 Analysis of existing bounds

First we provide an improved MMD generalization bound for active learning which is inspired by Cortes et al. (2019). Then we review a bound in terms of the Discrepancy of Cortes et al. (2019) and we review how to compute the Discrepancy quantity (Mansour et al. 2009). We show that the MMD can be computed using a novel eigenvalue analysis, and thereby making the MMD and Discrepancy bounds comparable. We wrap up the section with a probabilistic

interpretation of both bounds. As a roadmap for the reader we give an overview of the tightness relations in Table 1 which will be proven in this section and the next section.

4.1 Improved MMD bound for active learning

The MMD measures the similarity between the two unlabeled samples \hat{Q} and \hat{P} . Using this criterion we give a generalization bound similar to the one given by Wang and Ye (2013) suitable for active learning. The empirical MMD quantity is given by

$$\text{MMD}(\hat{P}, \hat{Q}) = \max_{\tilde{l} \in H_{\mathcal{L}}} \left(\frac{1}{n_{\hat{P}}} \sum_{x \in \hat{P}} \tilde{l}(x) - \frac{1}{n_{\hat{Q}}} \sum_{x \in \hat{Q}} \tilde{l}(x) \right).$$

here \tilde{l} is the worst-case function from a set of functions $H_{\mathcal{L}}$. We take the standard choice $H_{\mathcal{L}} = \{h \in \mathcal{H}_{\mathcal{L}} : \|h\|_{K_{\mathcal{L}}} \leq \Lambda_{\mathcal{L}}\}$. In Appendix A.1 we revisit how to compute the MMD quantity. We extend the technique of Cortes et al. (2019) to give a generalization bound in terms of the MMD. To get a bound for the MMD we approximate the loss function $g(h, f)(x) = l(h(x), f(x))$ using $H_{\mathcal{L}}$.

Proposition 1 (Agnostic MMD worst case bound) *Let l be any loss function $l : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$. Then for all $h \in H$ and any labeling function $f : \mathcal{X} \rightarrow \mathcal{Y}$ we have*

$$L_{\hat{P}}(h, f) \leq L_{\hat{Q}}(h, f) + \text{MMD}(\hat{P}, \hat{Q}) + \eta_{\text{MMD}}, \tag{3}$$

where $\eta_{\text{MMD}} = 2 \min_{\tilde{l} \in H_{\mathcal{L}}} \max_{h \in H, x \in \hat{P}} |g(h, f)(x) - \tilde{l}(x)|$.

Here η_{MMD} measures the approximation error since we may have that $g(h, f) \notin H_{\mathcal{L}}$.

Our MMD bound above differs in two aspects from the bound of Wang and Ye (2013). Wang and Ye (2013) estimate the MMD between the distributions P and Q . However, to estimate the MMD between distributions i.i.d. samples are required (Gretton et al. 2012, Appendix A.2). The sample \hat{Q} is not i.i.d. since it is chosen by an active learner.

Our bound allows for non-i.i.d. samples since it estimates the MMD between empirical samples and is therefore better suited for active learning. The second novelty is that we measure the error of approximating the loss function $g(h, f)$ using the term η_{MMD} . This allows us to adjust the MMD to the hypothesis set H and loss l similar to the Discrepancy measure of Cortes and Mohri (2014). We give the theorem below with a small proof sketch for the simplified case of the linear kernel. See the Appendix for the full proof.

Theorem 2 (Adjusted MMD) *Let l be the squared loss and assume $f \in H$ (realizable setting). If $K_{\mathcal{L}}(x_i, x_j) = K(x_i, x_j)^2$ and $\Lambda_{\mathcal{L}} = 4\Lambda^2$, then $g(h, f) \in H_{\mathcal{L}}$ and thus $\eta_{\text{MMD}} = 0$.*

Proof sketch Here we give a proof sketch for the case where K is the linear kernel: $K(x_i, x_j) = x_i^T x_j$. Then $h(x) = w_h^T x$ and $f(x) = w_f^T x$, and $g(h, f) = ((w_f - w_h)^T x)^2$ is a quadratic function of x . The featuremap of the kernel $K_{\mathcal{L}}(x_i, x_j) = K(x_i, x_j)^2$ are all monomials of degree 2 (Shawe-Taylor and Cristianini 2004, chap. 9.1). Therefore $H_{\mathcal{L}}$ can be used to model any quadratic function such as $g(h, f)$. Therefore if $\Lambda_{\mathcal{L}}$ is chosen appropriately we have $g(h, f) \in H_{\mathcal{L}}$.

Corollary 1 *Let l be the squared loss and $f \in H$ and let K be a Gaussian kernel with bandwidth σ . If $K_{\mathcal{L}}$ is a Gaussian kernel with bandwidth $\sigma_{\mathcal{L}} = \frac{\sigma}{\sqrt{2}}$ and $\Lambda_{\mathcal{L}} = 4\Lambda^2$ then $\eta_{\text{MMD}} = 0$.*

Compared to other works Theorem 2 gives a more informed way to choose the MMD kernel in the context of learning.¹ Typically, a Gaussian kernel is used for the MMD with $\sigma_{\mathcal{L}} = \sigma$. However, Corollary 1 shows that if $\sigma_{\mathcal{L}} = \sigma$, we may have that $\eta_{\text{MMD}} \neq 0$ even in the realizable setting, since $\sigma_{\mathcal{L}}$ is too large—the true loss function $g(h, f)$ is less smooth than the functions in $H_{\mathcal{L}}$. This is undesirable since η_{MMD} cannot be minimized during active learning. Our choice for $\sigma_{\mathcal{L}}$ is preferable, as it ensures $\eta_{\text{MMD}} = 0$ in the realizable setting.

4.2 Discrepancy bound

The Discrepancy is defined as

$$\text{disc}(\hat{P}, \hat{Q}) = \max_{h, h' \in H} |L_{\hat{P}}(h', h) - L_{\hat{Q}}(h', h)|. \tag{4}$$

Observe it depends on H and l and therefore automatically adjusts to the loss and hypothesis set. We give a bound of Cortes et al. (2019) in terms of the Discrepancy.

Theorem 3 (Agnostic Discrepancy worst case bound (Cortes et al. 2019)) *Assume that for all $x \in \mathcal{X}$ and for all $h \in H$ that $l(h(x), f(x)) \leq C$ and let l be the squared loss. Then for all $h \in H$ and any labeling function $f : \mathcal{X} \rightarrow \mathcal{Y}$ we have*

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

where $\eta_{\text{disc}} = 4C \min_{\tilde{f} \in H} \max_{x \in \hat{P}} |\tilde{f}(x) - f(x)|$.

Here η_{disc} measures the model misspecification. In the realizable setting, $f \in H$, and $\eta_{\text{disc}} = 0$.

4.3 Eigenvalue analysis

We show the relation between the Discrepancy and MMD using a novel eigenvalue analysis. To this end we introduce the matrix $M_{\hat{P}, \hat{Q}}$ to compute the Discrepancy.

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

For notational convenience we will often write M instead of $M_{\hat{P}, \hat{Q}}$. The matrix M measures the difference between two sets of samples using their second-order moment. Considering its kernelized version such comparison can implicitly take higher-order moments into account as well. In particular, for a Gaussian kernel all moments of the samples are compared and we have that $M = 0$ only if $\hat{P} = \hat{Q}$.

In the following we will look at the eigendecomposition of M . Since M is the difference between two covariance matrices, it can have positive and negative eigenvalues. A positive (negative) eigenvalue means that in the direction of the corresponding eigenvector \hat{P} has more (less) variance than \hat{Q} . Recall that in active learning, our aim is to approximate \hat{P} using representative samples \hat{Q} , and thus small absolute eigenvalues are desirable, because this would indicate that in the direction of the corresponding eigenvector \hat{P} is well approximated by \hat{Q} .

¹ The MMD is also used in other contexts, for example, the MMD can be used to determine if two sets of samples originate from the same distribution (Gretton et al. 2012).

Theorem 4 (Discrepancy computation (Mansour et al. 2009)) *Assume K is the linear kernel, $K(x_i, x_j) = x_i^T x_j$, and l is the squared loss, then*

$$\text{disc}(\hat{P}, \hat{Q}) = 4\Lambda^2 \max_i |\lambda_i| = 4\Lambda^2 \|\lambda\|_\infty. \quad (5)$$

where λ_i are the eigenvalues of M , and λ is the vector of eigenvalues of M .

Note that h' will later play the role of f , the true labeling function. The theorem shows that in the worst case, the h and h' that maximize the Discrepancy in Eq. 4 are chosen exactly in the direction where \hat{Q} and \hat{P} differ most, i.e., the direction of the largest absolute eigenvalue. Cortes and Mohri (2014) show that we can replace M by M_K to compute the Discrepancy for any PSD kernel.²

Before we can give our main result we require some additional notation. Assume that the eigenvalues λ_i of M are ordered by absolute value where $|\lambda_1|$ is the largest absolute eigenvalue. λ indicates the vector of eigenvalues, with $r = \text{rank}(M)$ non-zero eigenvalues. e_i is the normalized (unit-length) eigenvector corresponding to λ_i . By careful analysis we can realize the relationship between M and the featuremap of the squared kernel to show that the MMD can be computed as follows.

Theorem 5 (MMD Computation) *Let $K_{\mathcal{L}}(x_i, x_j) = K(x_i, x_j)^2$ and $\Lambda_{\mathcal{L}} = 4\Lambda^2$, then*

$$\text{MMD}(\hat{P}, \hat{Q}) = 4\Lambda^2 \|\lambda\|_2. \quad (6)$$

This theorem shows that the MMD measures differences between the samples \hat{Q} and \hat{P} differently. The Discrepancy only measures similarity along one dimension, namely the direction where the samples differ the most. The MMD considers all dimensions to compare the samples \hat{Q} and \hat{P} . Due to the square in the Euclidean norm, the MMD gives directions that differ more more weight in the comparison.

Corollary 2 *Under the conditions of Theorem 2, $\text{disc}(\hat{P}, \hat{Q}) \leq \text{MMD}(\hat{P}, \hat{Q})$.*

Under these conditions the Discrepancy bound (Theorem 3) is tighter than the MMD bound (Proposition 1), since $\eta_{\text{MMD}} = \eta_{\text{disc}} = 0$. Since the Discrepancy bound is tighter, one may expect that active learning by minimization of the Discrepancy may result in better active learning queries than minimization of the MMD, in particular if η_{MMD} and η_{disc} are small or zero.

4.4 Probabilistic analysis

We show the MMD can provide a tighter bound on the expected loss under certain probabilistic assumptions. From this point on we assume the conditions of Theorem 2 and take h to be the model trained on the set \hat{Q} , and f to be the true labeling function. In addition, define $u = h - f$ and $U = \{u \in \mathcal{H} : \|u\|_K \leq 2\Lambda\}$ and let $\bar{u}_i = u^T e_i$, where e_i is the eigenvector of M .

Then $\|u\|_K = \|\bar{u}\|_K \leq 2\Lambda$, since \bar{u} is a rotated version of u . It is more convenient to work with \bar{u} , since then the matrix M diagonalizes: $u^T M u = \sum_i \bar{u}_i \lambda_i$.

The difference u is the unknown error our trained model h makes compared with the true model f . By making different probabilistic assumptions about the distribution of u we can

² See the Appendix (Eq. 17) for the definition of M_K , additional details and the proof of this theorem. All our theoretical results that follow hold for both M and M_K . For simplicity we use M in the main text.

arrive at different bounds. We now provide the building block for our probabilistic bounds. By noting that $L_{\hat{P}}(h, f) - L_{\hat{Q}}(h, f) = u^T M u$ and by making use of the triangle inequality, we find the following.

Lemma 1 (Probabilistic bound) *Assume³ u is distributed according to a pdf $p(u)$ over U . Then*

$$\mathbb{E}_u L_{\hat{P}}(h, f) \leq \mathbb{E}_u L_{\hat{Q}}(h, f) + \mathbb{E}_u G(u, M), \tag{7}$$

where we defined $G(u, M) = \sum_i \bar{u}_i^2 |\lambda_i|$.

Observe that $G(u, M)$ is a weighted sum, where each $|\lambda_i|$ is weighted by \bar{u}_i^2 . Recall that $L_{\hat{Q}}(h, f)$ is generally small due to the training procedure of the model, thus generally $\mathbb{E}_u L_{\hat{Q}}(h, f)$ will be small as well. Therefore we focus our probabilistic analysis on the term $\mathbb{E}_u G(u, M)$. By giving bounds on this quantity, we derive several probabilistic bounds that hold in expectation w.r.t. u .

The Discrepancy can be interpreted to put all probability mass on $u = 2\Lambda e_1$.

Proposition 2 (Worst case: Probabilistic Discrepancy) *Given the pdf $p(u) = \delta(u - 2\Lambda e_1)$ where $\delta(x)$ is the Dirac delta distribution. Then*

$$\mathbb{E}_u L_{\hat{P}}(h, f) \leq \mathbb{E}_u L_{\hat{Q}}(h, f) + \text{disc}(\hat{P}, \hat{Q}) \tag{8}$$

Only one $u \in U$ can be observed under this pdf. This is a worst case distribution because this $p(u)$ maximizes $\mathbb{E}_u G(u, M)$. The Discrepancy assumes that the model error u points exactly in the direction that causes us to make the biggest error on \hat{P} . Under this distribution the Discrepancy gives a tighter bound on the expected loss than the MMD because of Corollary 2. Under a different $p(u)$ the MMD bound is tighter.

Theorem 6 (Pessimistic case: Probabilistic MMD) *Let $p(u)$ be a pdf on U_s such that⁴*

$$\mathbb{E}_u \bar{u}_i^2 = 4\Lambda^2 |\lambda_i| (\sqrt{r} \|\lambda\|_2)^{-1}, \tag{9}$$

then

$$\mathbb{E}_u L_{\hat{P}}(h, f) \leq \mathbb{E}_u L_{\hat{Q}}(h, f) + \frac{1}{\sqrt{r}} \text{MMD}(\hat{P}, \hat{Q}) \leq \mathbb{E}_u L_{\hat{Q}}(h, f) + \text{disc}(\hat{P}, \hat{Q}).$$

Unlike for the distribution of the Discrepancy, for the above $p(u)$ it is possible to observe different model errors u . However, the model error u in this case is biased: Equation 9 suggests that u is more likely to point in the direction of eigenvectors with large absolute eigenvalues. This assumption is pessimistic since large absolute eigenvalues can contribute more to $\mathbb{E}_u G(u, M)$. Another way to interpret this is that model errors are more likely to occur in directions where \hat{Q} and \hat{P} differ more. Because \hat{Q} and \hat{P} differ more in those directions, these model errors can count more towards the MSE on \hat{P} .

For this $p(u)$ the MMD bound is tighter. If the probabilistic assumption of the MMD is more accurate, we can expect that the MMD active learner will yield better active learning queries than the Discrepancy.

³ This could be motivated for example, by placing a prior on f , then u would be a random variable. Another motivation is that we do not know u , and need to model it somehow to come to applicable generalization bounds. The Discrepancy assumes a worst-case scenario (it maximizes with respect to u), while we now consider assuming a distribution on u .

⁴ To deal with infinite-dimensional RKHS we choose $p(u)$ on U_s instead of U , where U_s is the part of U restricted to the span of $X_{\hat{P}}$. Here r is the effective dimension: $r = \text{dim}(U_s)$. This is necessary, otherwise sampling uniformly from an infinite-dimensional sphere can lead to problems. See Appendix C for more details.

5 Nuclear discrepancy

In this section we motivate the optimistic probabilistic assumption that leads to the Nuclear Discrepancy (ND) bound. First, let us introduce the Nuclear Discrepancy quantity

$$\text{disc}_N(\hat{P}, \hat{Q}) = 4\Lambda^2 \|\lambda\|_1.$$

In the absence of any prior knowledge, we choose the pdf $p(u)$ according to the well established principle of maximum entropy. This principle dictates that in case nothing is known about a distribution, the distribution with the largest entropy should be chosen (Jaynes 1957). Accordingly, we choose $p(u)$ uniform over U , which leads to the following.

Theorem 7 (Optimistic case: Probabilistic ND) *Let $p(u)$ be uniform over all $u \in U_s$, then⁴*

$$\mathbb{E}_u L_{\hat{P}}(h, f) \leq \mathbb{E}_u L_{\hat{Q}}(h, f) + \frac{1}{r+2} \text{disc}_N(\hat{P}, \hat{Q}).$$

In addition we have that $\text{disc}_N(\hat{P}, \hat{Q}) \leq \sqrt{r} \text{MMD}(\hat{P}, \hat{Q}) \leq r \text{disc}(\hat{P}, \hat{Q})$.

Under the uniform distribution, u is unbiased: each direction for the model error is equally likely. This is more optimistic than the assumption of the MMD, where u was biased towards directions that could larger errors on \hat{P} . Because now u is not biased, $\mathbb{E}_u G(u, M)$ is smaller under this $p(u)$ than in Theorems 2 and 6 and so this $p(u)$ is more optimistic. The Nuclear Discrepancy (ND) owns its name to the fact that it is proportional to the nuclear matrix norm of M .

An appealing property of this choice of $p(u)$ is that, given a fixed \hat{P} , any choice of \hat{Q} does not influence $p(u)$. For the Discrepancy and the MMD, choosing different \hat{Q} leads to different $p(u)$. Thus choosing queries changes the distribution of $p(u)$ and thus also implicitly the distribution of h and f . Instead, for the ND, our queries don't influence the distribution of h and f . This assumption seems reasonable, since f is usually assumed to be fixed and independent of our actions.

Under the uniform distribution the ND provides the tightest bound on the expected loss, while the MMD bound is looser, and the Discrepancy bound is the loosest. Therefore, if this probabilistic assumption is the most accurate, minimization of the Nuclear Discrepancy may lead to the best queries for active learning, followed by the MMD and Discrepancy, in that order.⁵

6 Experiments

We explain the setup and baselines, afterward we review our main results: the realizable setting. We discuss the results and examine the probabilistic assumptions empirically. Somewhat similar results are observed in the agnostic setting which we will briefly discuss. An additional experiment investigates the influence of subsampling of datasets on our results. This subsampling experiment and all results of the agnostic case are discussed in detail in the Appendix.

⁵ As an aside, note that $\text{MMD}(\hat{P}, \hat{Q}) \leq \text{disc}_N(\hat{P}, \hat{Q})$, since $\|\lambda\|_2 \leq \|\lambda\|_1$. Therefore, by upperbounding the MMD in (3) we can also give a (looser) worst-case bound in terms of the ND for the agnostic case.

6.1 Experimental setup and baselines

An overview of the experimental procedure is given in Algorithm 1. A training set (65%) and test set (35%) are used—the training set corresponds to \hat{P} and we indicate the testset by \hat{T} . We use the active learners to select batches of size $n = 1, 2, \dots, 50$. For computational reasons we select batches in a sequential greedy fashion. Initially at $t = 0$ the batch is empty: $\hat{Q}_0 = \emptyset$. In iteration $1 \leq t \leq n$ the active learner selects a sample x_t from the unlabeled pool $\hat{U}_{t-1} = \hat{P} \setminus \hat{Q}_{t-1}$ according to $x_t = \arg \min_{s \in \hat{U}_{t-1}} \text{obj}(\hat{P}, \hat{Q}_{t-1} \cup s)$. We perform experiments multiple times to ensure significance of the results. We call each repetition a run, and for each run a new training and test split is used. During one run, we evaluate each active learner using the described procedure of Algorithm 1.

Algorithm 1: Zero shot active learning

```

input : Unlabeled trainingset  $\hat{P}$ , Testset  $\hat{T}$ , labeling budget  $n$ , active learning criterium
          $\text{obj} \in \{\text{MMD}, \text{disc}, \text{disc}_N\}$ , hyperparameters of model  $\mu, \sigma$ 
output: MSE performance on testset  $T$ 
1  $\hat{Q}_0 \leftarrow \emptyset;$  // Init batch
2  $\hat{U}_0 \leftarrow \hat{P};$  // Init unlabeled pool
3 for  $t \leftarrow 1$  to  $n$  do
4    $x_t \leftarrow \arg \min_{s \in \hat{U}_{t-1}} \text{obj}(\hat{P}, \hat{Q}_{t-1} \cup s);$  // Find optimal query
5    $\hat{Q}_t \leftarrow \hat{Q}_{t-1} \cup x_t;$  // Update batch
6    $\hat{U}_t \leftarrow \hat{P} \setminus \hat{Q}_t;$  // Update unlabeled pool
7 end
8 Request all labels for objects  $\hat{Q}_n$  to obtain labeled dataset  $\hat{Q}_n^{\text{lab}}$ ;
9 Train kernel regularized least squares model  $h$  on  $\hat{Q}_n^{\text{lab}}$  with hyperparameters  $\mu, \sigma$ ;
10 Compute mean squared error (MSE) of  $h$  on unseen testset  $T$ ;

```

As baseline we use random sampling and a greedy version of the state-of-the-art MMD active learner (Chattopadhyay et al. 2012; Wang and Ye 2013). We compare the baselines with our novel active learners: the Discrepancy active learner and the Nuclear Discrepancy active learner.

The methods are evaluated on 13 datasets that originate either from the UCI Machine Learning repository (Lichman 2013) or were provided by Cawley and Talbot (2004). See Appendix E for the dataset names and characteristics. Furthermore, we perform an experiment on the image dataset MNIST. The MNIST dataset (LeCun et al. 1998) consists of images of handwritten digits of size 28×28 pixels. By treating each pixel as a feature, the dimensionality of this dataset is 784 which is relatively high dimensional. Like Yang and Loog (2018) we construct 3 difficult binary classification problems: 3vs5, 7vs9 and 5vs8.

To make datasets conform to the realizable setting we use the approach of Cortes and Mohri (2014): we fit a model of our hypothesis set to the whole dataset and use its outputs as labels.

To set reasonable hyperparameters we use a similar procedure as Gu et al. (2012). We use labeled data before any experiments are performed to perform model selection to determine hyperparameters (σ and μ of the KRLS model). This can be motivated by the fact that in practice a related task or dataset may be available in order to obtain a rough estimate of the hyperparameter settings. This procedure makes sure η_{MMD} and η_{disc} are small in the agnostic setting.

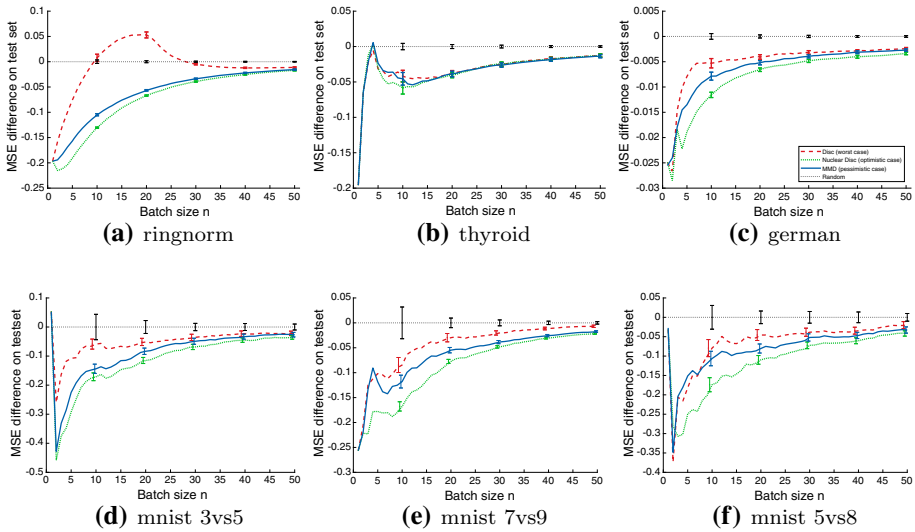


Fig. 1 Learning curves for several datasets for the realizable setting. Results are averaged over 100 runs. The MSE is measured with respect to random sampling (lower is better)

Recall that the active learners minimize bounds on $L_{\hat{p}}(h, f)$. Therefore active learners then implicitly also minimizes a bound on $L_P(h, f)$, see Theorem 1. By choosing hyperparameters in the described way above, we ensure that the Rademacher complexity term $R_m(H)$ is not too large and we don't overfit. We measure performance on an independent test set in order to get an unbiased estimate of $L_P(h, f)$.

To aid reproducibility we give all hyperparameters and additional details in Appendix E. We set $\sigma_{\mathcal{L}}$ according to our analysis in Corollary 1.

6.2 Realizable setting

First we benchmark the active learners in the realizable setting. In this setting we are assured that $\eta = 0$ in all bounds and therefore we eliminate unexpected effects that can arise due to model misspecification. We study this scenario to validate our theoretical results and gain more insight, furthermore, note that this scenario is also studied in adaptation (Cortes and Mohri 2014).

Several learning curves are shown in Fig. 1, all curves can be found in Appendix H.1. The MSE of the active learner minus the mean performance (per query) of random sampling is displayed on the y-axis (lower is better). The curve is averaged over 100 runs. Error bars represent the 95% confidence interval of the mean computed using the standard error.

We summarize results on all datasets using the Area Under the (mean squared error) Learning Curve (AULC) in Table 2. The AULC is a different metric than the well known AUROC or AUPRC measures. The AUROC measure summarize the performance of a model for different misclassification costs (type I and type II costs) and the AUPRC is useful when one class is more important than the other, such as in object detection.

By contrast, AULC is specifically suited to active learning, and summarizes the performance of an active learning algorithm for different number of labeling budgets (O'Neill et al. 2017; Huijser and van Gemert 2017; Settles and Craven 2008). Low AULC is obtained when

Table 2 Area Under the mean squared error Learning Curve (AULC) for the strategies in the realizable setting, averaged over 100 runs

Dataset	Random	Discrepancy	MMD	Nuclear Discrepancy
vehicles	11.1 (2.2)	8.0 (1.0)	7.9 (0.9)	7.9 (0.9)
heart	3.5 (0.8)	2.3 (0.3)	2.2 (0.3)	2.1 (0.3)
sonar	13.9 (1.7)	12.5 (1.2)	11.9 (1.1)	11.3 (1.2)
thyroid	6.8 (1.5)	5.2 (0.9)	5.1 (0.9)	5.0 (1.0)
ringnorm	13.2 (1.2)	12.7 (0.8)	10.0 (0.3)	9.4 (0.3)
ionosphere	7.0 (1.3)	5.6 (0.8)	5.0 (0.8)	4.6 (0.6)
diabetes	1.7 (0.4)	1.2 (0.1)	1.2 (0.1)	1.2 (0.1)
twonorm	6.4 (1.2)	4.1 (0.4)	3.7 (0.4)	3.3 (0.3)
banana	7.5 (0.9)	5.0 (0.4)	4.8 (0.3)	4.8 (0.3)
german	1.4 (0.3)	1.2 (0.1)	1.1 (0.1)	1.0 (0.1)
splice	10.8 (1.3)	9.9 (0.8)	9.9 (0.9)	9.0 (0.9)
breast	3.4 (0.9)	2.1 (0.2)	2.1 (0.2)	2.0 (0.2)
mnist 3vs5	29.5 (4.3)	26.9 (2.3)	25.0 (2.1)	23.8 (1.7)
mnist 7vs9	13.2 (2.5)	10.9 (1.4)	10.0 (1.0)	8.9 (0.7)
mnist 5vs8	30.1 (3.4)	26.9 (2.7)	26.1 (2.3)	24.5 (2.1)

Bold indicates the best result, or results that are not significantly worse than the best result, according to a paired t-test ($p = 0.05$). Parenthesis indicate standard deviation

an active learner quickly learns a model with low MSE. If a method in the table is bold, it either means it is the best method (as judged by the mean), or if it is not significantly worse than the best method (as judged by the t-test).

Significance improvement is judged by a paired two tailed t-test (significance level $p = 0.05$). We may use a paired test since during one run all active learners are evaluated using the same training and test split.

In the majority of the cases the MMD improves upon the Discrepancy (see Table 2). The results on the ringnorm dataset are remarkable, here the Discrepancy sometimes performs worse than random sampling, see Fig. 1. We observe that generally the Discrepancy performs the worst. These results illustrates that tighter worst case bounds do not guarantee improved performance. The proposed ND active learner significantly improves upon the MMD in 9 out of the 13 datasets tested. Here we counted MNIST once, while we remark that on all subproblems the ND improves significantly on the MMD. This provides evidence that the proposed method can also deal with high-dimensional datasets. In case the ND does not perform the best, it ties with the MMD or Discrepancy. The ND never performs significantly worse. This ranking of the methods exactly corresponds to the order of the bounds given by Theorem 7 under our optimistic probabilistic assumptions. This supports our hypothesis that we find ourselves more often in a more optimistic average-case scenario.

6.3 Decomposition of probabilistic bounds

Since we are in the realizable setting we can compute $u = h - f$ with the true labeling function f and our trained model h . Thus we can compute each term in the sum of $G(u, M)$

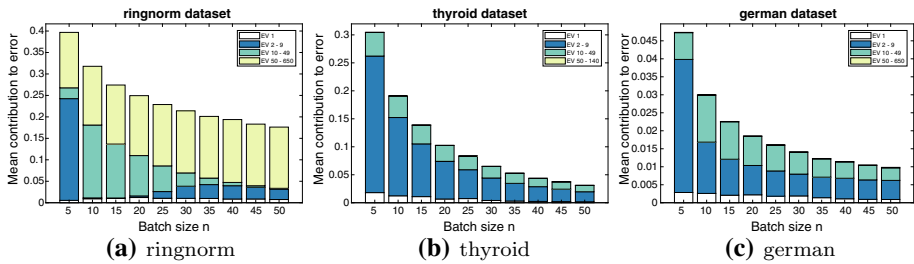


Fig. 2 Decomposition of the sum $G(u, M)$ during active learning for several datasets. EV1 indicates the contribution of λ_1 , EV2-9 indicate the summed contributions of $\lambda_2, \dots, \lambda_9$, etc. Averaged over 100 runs of the random active learner. λ_1 in most cases contributes little and in general all λ_i contribute to $G(u, M)$. This supports the optimistic probabilistic assumptions

in (7) during the experiments.⁶ We show the contribution of each eigenvalue to $G(u, M)$. In Fig. 2 we show this decomposition using a stacked bar chart during several active learning experiments of the baseline active learner ‘Random’.⁷ Here EV1 indicates the largest absolute eigenvalue, its contribution is given by $\bar{u}_1^2|\lambda_1|$ (see also (7)). EV 2 - 9 to indicate the summed contribution: $\sum_{i=2}^9 \bar{u}_i^2|\lambda_i|$, etc. The mean contributions over 100 runs are shown.

Observe that the contribution of $|\lambda_1|$ to $G(u, M)$ is often small, it is shown by the small white bar at the bottom of the barchart. Therefore the Discrepancy active learner chooses suboptimal samples: its strategy is optimal for a worst-case scenario $G(u, M) = 4\Lambda^2|\lambda_1|$ that is very rare. We observe that typically all λ_i contribute to $G(u, M)$ supporting our probabilistic assumption.

6.4 Agnostic setting

For completeness, we briefly mention the agnostic setting, for all details see Appendix F. In the agnostic setting the rankings of methods can change and performance differences become less significant. The ND still improves more upon the MMD than the reverse, however, the trend is less significant. Because our assumption $\eta = 0$ is violated our theoretical analysis is less applicable.

For the MNIST experiments we however find that the results for some subproblems almost coincides with the realizable setting: apparently, for the MNIST dataset the model misspecification is very small. This may be because the dataset is of relatively high dimensionality.

6.5 Influence of subsampling

We briefly mention an additional experiment that we have performed on the splice dataset to see how subsampling affects performance. To this end we measure the performance while we vary the pool size \hat{P} by changing the amount of subsampling. This to investigate how the proposed methods would perform for problems with a larger scale. For all details please see Appendix G, here we will summarize our findings.

For small pool sizes all active learners experience a drop in performance. We find the larger the pool, the better the performance, up until some point at which the performance levels

⁶ See Appendix D for details how to compute $G(u, M)$ in case kernels are used.

⁷ Results for other strategies are similar. Results on all datasets are given in Appendix H.2.

off. The experiment provides evidence that if finer subsampling is used or larger datasets are used, methods typically improve in performance up to a point where performance levels off.

7 Discussion

In the experiments we have observed that in the realizable setting the order of the bounds under our more optimistic probabilistic assumptions give the best indication of active learning performance. The empirical decomposition of $G(u, M)$ during experiments also supports our hypothesis that we generally find ourselves in a more optimistic scenario instead of a worst case scenario.

Still it is meaningful to look at worst-case guarantees, though the worst-case should be expected to occur. The worst-case assumed by the Discrepancy can never occur in the realizable setting, and we believe it is also highly unlikely in the agnostic setting. The strength of our probabilistic approach is that it considers all scenarios equally and does not focus too much on specific scenarios, making the strategy more robust.

Our work illustrates that the order of bounds can change under varying conditions and thus tightness of bounds is not the whole story. The conditions under which the bounds hold are equally important, and should reflect the mathematical setting as much as possible. For example, in a different setting where an adversary would pick u , the Discrepancy active learner would be most appropriate. This insight illustrates that not only by obtaining tighter bounds active learning performance can be improved, but by finding more appropriate assumptions (bound-based) active learners can be improved as well.

Our work supports the idea of Germain et al. (2013) who introduce a probabilistic version of the Discrepancy bound for the zero-one loss (Ben-David et al. 2010). Our conclusions also support that the direct Cortes et al. (2019) takes: by using more accurate assumptions to better characterize the the worst case scenario, performance may be improved.

In our study we have focused on minimizing the mean squared error. It would be interesting to investigate the extension of the Nuclear Discrepancy to other loss functions, in particular the zero-one loss. As far as we can see, however, such an extension is not trivial. The above mentioned probabilistic version of the Discrepancy by Germain et al. (2013) may provide some inspiration to achieve this, but they offer a PAC Bayes approach that cannot be easily adapted to the probabilistic setting we consider.

Where the experiments in the realizable setting provide clear insights, the results concerning the agnostic setting are not fully understood. A more in depth experimental study of the agnostic setting is complicated by unexpected effects of η . Since probabilistic bounds are the most informative in the realizable setting, it is of interest to consider probabilistic bounds for the agnostic setting as well.

In our experiments we have used greedy optimization to compute the batch \hat{Q}_n . It is theoretically possible to optimize a whole batch of queries in one global optimization step. However, for the MMD this problem is known to be NP-hard (Chattopadhyay et al. 2012). Minimizing the Discrepancy is also non-trivial, as illustrated by the involved optimization procedure required by Cortes and Mohri (2014) for domain adaptation. Note that their optimization problem is easier than the optimization problem posed by active learning, where binary constraints are necessary. Since the objective value of the Nuclear Discrepancy is given by an expectation which can be approximated using sampling, we believe it may be possible to speed up the optimization by using approximations.

In this work we have only considered single-shot batch active learning. In regular batch-mode active learning label information of previously selected samples can be used to improve

query selection. This can be accommodated in our active learner by refining $p(u)$ using label information. Our results have implications for adaptation as well. We suspect our suggested choice of $\sigma_{\mathcal{L}}$ may improve the MMD domain adaptation method (Huang et al. 2007). Furthermore, our results suggest that the ND is a promising objective for adaptation.

8 Conclusion

To investigate the relation between generalization bounds and active learning performance, we gave several theoretical results concerning the bound of the MMD active learner and the Discrepancy bound. In particular, we showed that the Discrepancy provides the tightest worst-case bound. We introduced a novel quantity; Nuclear Discrepancy, motivated from optimistic probabilistic assumptions derived from the principle of maximum entropy. Under these probabilistic assumptions the ND provides the tightest bound on the expected loss, followed by the MMD, and the Discrepancy provides the loosest bound.

Experimentally, we observed that in the realizable setting the Discrepancy performs the worst, illustrating that tighter worst-case bounds do not guarantee improved active learning performance. Our optimistic probabilistic analysis clearly matches the observed behavior in the realizable setting: the proposed ND active learner improves upon the MMD, and the MMD improves upon the Discrepancy active learner. We find that even on the high-dimensional image dataset MNIST our method is competitive. A similar, weaker, trend is observed in the agnostic case. One of our key conclusions is that not only bound tightness is important for active learning performance, but that appropriate assumptions are equally important.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

Appendix

A Background theory

A.1 MMD

The MMD quantity can be computed in practice by rewriting it as follows:

$$\begin{aligned} \text{MMD}(\hat{P}, \hat{Q}) &= \max_{\tilde{l} \in H_{\mathcal{L}}} \frac{1}{n_{\hat{P}}} \sum_{x \in \hat{P}} \langle \tilde{l}, \psi_{K_{\mathcal{L}}}(x) \rangle_{K_{\mathcal{L}}} - \frac{1}{n_{\hat{Q}}} \sum_{x \in \hat{Q}} \langle \tilde{l}, \psi_{K_{\mathcal{L}}}(x) \rangle_{K_{\mathcal{L}}} \\ &= \max_{\tilde{l} \in H_{\mathcal{L}}} \langle \tilde{l}, \mu_{\hat{P}} - \mu_{\hat{Q}} \rangle_{K_{\mathcal{L}}} \end{aligned} \quad (10)$$

$$= \Lambda_{\mathcal{L}} \|\mu_{\hat{P}} - \mu_{\hat{Q}}\|_{K_{\mathcal{L}}}. \quad (11)$$

In the first step we used that $\tilde{l}(x) = \langle \tilde{l}, \psi_{K_{\mathcal{L}}}(x) \rangle_{K_{\mathcal{L}}}$ due to the reproducing property (Mohri et al. 2012, p. 96). Here $\psi_{K_{\mathcal{L}}}$ is the featuremap from $\mathcal{X} \rightarrow H_{\mathcal{L}}$. The second step follows from the linearity of the inner product. In (10) we defined $\mu_{\hat{P}} = \frac{1}{n_{\hat{P}}} \sum_{x \in \hat{P}} \psi_{K_{\mathcal{L}}}(x)$ and similarly for $\mu_{\hat{Q}}$, note that $\mu_{\hat{Q}}, \mu_{\hat{P}} \in H_{\mathcal{L}}$. The last step follows from the fact that the vector in $H_{\mathcal{L}}$ maximizing the term in (10) is

$$\frac{\mu_{\hat{P}} - \mu_{\hat{Q}}}{\|\mu_{\hat{P}} - \mu_{\hat{Q}}\|_{K_{\mathcal{L}}}} \Lambda_{\mathcal{L}}$$

Because of the symmetry of $\|\mu_{\hat{P}} - \mu_{\hat{Q}}\|_{K_{\mathcal{L}}}$ with respect to \hat{P} and \hat{Q} , this derivation also holds if we switch \hat{P} and \hat{Q} . Therefore:

$$\max_{\tilde{l} \in H_{\mathcal{L}}} \left| \frac{1}{n_{\hat{P}}} \sum_{x \in \hat{P}} \tilde{l}(x) - \frac{1}{n_{\hat{Q}}} \sum_{x \in \hat{Q}} \tilde{l}(x) \right| = \Lambda_{\mathcal{L}} \|\mu_{\hat{P}} - \mu_{\hat{Q}}\|_{K_{\mathcal{L}}}$$

Therefore for all $\tilde{l} \in H_{\mathcal{L}}$ the following holds

$$\left| \frac{1}{n_{\hat{P}}} \sum_{x \in \hat{P}} \tilde{l}(x) - \frac{1}{n_{\hat{Q}}} \sum_{x \in \hat{Q}} \tilde{l}(x) \right| \leq \text{MMD}(\hat{P}, \hat{Q}) \tag{12}$$

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

$$\begin{aligned} \text{MMD}(\hat{P}, \hat{Q}) &= \Lambda_{\mathcal{L}} \sqrt{\langle \mu_{\hat{Q}}, \mu_{\hat{Q}} \rangle_{K_{\mathcal{L}}} - 2\langle \mu_{\hat{P}}, \mu_{\hat{Q}} \rangle_{K_{\mathcal{L}}} + \langle \mu_{\hat{P}}, \mu_{\hat{P}} \rangle_{K_{\mathcal{L}}}} \\ &= \Lambda_{\mathcal{L}} \sqrt{\text{MMD}_{\text{comp}}(\hat{Q}, \hat{Q}) - 2\text{MMD}_{\text{comp}}(\hat{P}, \hat{Q}) + \text{MMD}_{\text{comp}}(\hat{P}, \hat{P})} \end{aligned}$$

where we introduced $\text{MMD}_{\text{comp}}(\hat{R}, \hat{S}) = \frac{1}{n_{\hat{R}}n_{\hat{S}}} \sum_{x \in \hat{R}, x' \in \hat{S}} K_{\mathcal{L}}(x, x')$.

A.2 Discrepancy

In this section we calculate the discrepancy analytically for the squared loss in the linear kernel as in Mansour et al. (2009). We then extend the computation to any arbitrary kernel as in Cortes and Mohri (2014). Finally, we prove the agnostic generalization bound in terms of the Discrepancy (Theorem 3). The theorems and proofs here were first given by Mansour et al. (2009), Cortes and Mohri (2014), and Cortes et al. (2019) but we repeat them here for completeness.

Lemma 2 (Mansour et al. 2009) *For $h, h' \in H$ we have*

$$\left| L_{\hat{P}}(h, h') - L_{\hat{Q}}(h, h') \right| = \left| \sum_{i=1}^r \tilde{u}_i^2 \lambda_i \right|. \tag{13}$$

Proof We can show

$$L_{\hat{P}}(h, h') = \frac{1}{n_{\hat{P}}} (X_{\hat{P}}h - X_{\hat{P}}h')^T (X_{\hat{P}}h - X_{\hat{P}}h') = \frac{1}{n_{\hat{P}}} u^T X_{\hat{P}}^T X_{\hat{P}} u$$

using some algebra, where $u = h - h'$. Rewrite $L_{\hat{Q}}(h, h')$ similarly and subtract them to find

$$L_{\hat{P}}(h, h') - L_{\hat{Q}}(h, h') = u^T M u. \tag{14}$$

Since M is a real symmetric matrix, M is a normal matrix and admits an orthonormal eigendecomposition with real eigenvalues

$$M = \sum_i^d e_i \lambda_i e_i^T.$$

Here λ_i is the i th eigenvalue and e_i is the corresponding orthonormal eigenvector. Since M is normal its eigenvectors form an orthonormal basis for \mathbb{R}^d . Therefore we can express u in terms of e :

$$u = \sum_i^d \bar{u}_i e_i$$

Where \bar{u}_i is the projection of u on e_i , $\bar{u}_i = e_i^T u$. Note \bar{u} is a rotated version of u and therefore both have the same norm, $\|u\|_2 = \|\bar{u}\|_2$. Now we can rewrite (14) as

$$u^T M u = \sum_i^d u^T e_i \lambda_i e_i^T u = \sum_{i=1}^r \bar{u}_i^2 \lambda_i. \tag{15}$$

Note that M has $r = \text{rank}(M)$ non-zero eigenvalues. Combining (14) and (15) and taking the absolute value on both sides shows the result. \square

Now we are ready to compute the Discrepancy for the linear kernel.

Theorem 8 (Discrepancy computation (Mansour et al. 2009)) *Assume K is the linear kernel, $K(x_i, x_j) = x_i^T x_j$, and l is the squared loss, then*

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

where λ_i are the eigenvalues of $M_{\hat{P}, \hat{Q}} = M$.

Proof First we use Lemma 2.

$$\text{disc}(\hat{P}, \hat{Q}) = \max_{\|\bar{u}\| \leq 2\Lambda} \left| \sum_i^r \bar{u}_i^2 \lambda_i \right| = \max \left(\max_{\|\bar{u}\| \leq 2\Lambda} \sum_i^r \bar{u}_i^2 \lambda_i, \max_{\|\bar{u}\| \leq 2\Lambda} \sum_i^r -\bar{u}_i^2 \lambda_i \right)$$

Now we solve the left term in the maximization. Observe that this is a weighted sum where each \bar{u}_i weighs each eigenvalue λ_i . To maximize this quantity we put as much weight as possible on the largest positive eigenvalue: $u = e_{i_{\max}} 2\Lambda$, where $i_{\max} = \arg \max_i \lambda_i$. We find

$$\max_{\|\bar{u}\| \leq 2\Lambda} \sum_i^d \bar{u}_i^2 \lambda_i = 4\Lambda^2 \max_i \lambda_i.$$

To solve the second maximization, introduce $\bar{\lambda}_i = -\lambda_i$. Then we maximize the same quantity as before but now λ replaced by $\bar{\lambda}$. It follows that the maximum is attained for $u = e_{i_{\min}} 2\Lambda$, where $i_{\min} = \arg \min_i \lambda_i$. We find

$$\text{disc}(\hat{P}, \hat{Q}) = 4\Lambda^2 \max(\lambda_i, \bar{\lambda}_i),$$

eliminating the maximum proves the result. \square

Now we will describe how to compute the Discrepancy in case we work with an arbitrary kernel K . In this case we have to work in the RKHS \mathcal{H} of the kernel K . Define $z(x) = \psi_K(x)$, and let $Z_{\hat{P}}$ be the datamatrix where each row is given by $z(x) : x \in \hat{P}$. Define $Z_{\hat{Q}}$ in the analogously. In this case Theorem 8 still holds, and the Discrepancy is given by the eigenvalues of M_Z :

$$M_Z = \frac{1}{n_{\hat{P}}} Z_{\hat{P}}^T Z_{\hat{P}} - \frac{1}{n_{\hat{Q}}} Z_{\hat{Q}}^T Z_{\hat{Q}} \tag{16}$$

However, now we run into problems, since for an arbitrary kernel K the dimensions of \mathcal{H} can be very large or infinite, such as the case for the Gaussian kernel. Then we clearly cannot compute the matrix M_Z or its eigenvalues.

In the following we show that M_Z and M_K have the same eigenvalues. Then, to compute the Discrepancy with any kernel K , we can simply use the eigenvalues of M_K . First, let us define M_K .

$$M_K = K_{\hat{P}\hat{P}}D \tag{17}$$

where $K_{\hat{P}\hat{P}}$ is the $n_{\hat{P}} \times n_{\hat{P}}$ matrix where entry i, j is given by $K(x_i, x_j)$, and where D is a diagonal matrix where

$$D_{ii} = \begin{cases} \frac{1}{n_{\hat{P}}} - \frac{1}{n_{\hat{Q}}} & \text{if } x_i \in \hat{Q} \\ \frac{1}{n_{\hat{P}}} & \text{otherwise.} \end{cases}$$

Lemma 3 (Cortes and Mohri 2014) *The eigenvalues of M_Z and M_K are the same.*

Proof Recalling that $\hat{Q} \in \hat{P}$, and using some algebra, it can be shown that M_Z can be written as

$$M_Z = Z_{\hat{P}}^T D Z_{\hat{P}}.$$

Now we suggestively write M_Z and M_K as

$$\begin{aligned} M_Z &= (Z_{\hat{P}}^T D) Z_{\hat{P}} \\ M_K &= Z_{\hat{P}} (Z_{\hat{P}}^T D) = K_{\hat{P}\hat{P}} D \end{aligned}$$

Here we used the fact that $K(x_i, x_j) = \langle \psi_K(x_i), \psi_K(x_j) \rangle_K$ (kernel trick) to rewrite M_K . Since the matrix product AB and BA have the same eigenvalues (Cortes and Mohri 2014), M_K and M_Z have the same eigenvalues. \square

Theorem 9 *Let K be any arbitrary PSD kernel. Then*

$$\text{disc}(\hat{P}, \hat{Q}) = 4A^2 \max_i |\lambda_i| = 4A^2 \|\lambda\|_{\infty} \tag{18}$$

where λ is the vector of eigenvalues of the matrix M_K or M_Z , where M_K was defined in (17) and M_Z was defined in (16).

Proof First observe that Theorem 8 still holds, but we have to replace M by M_Z in case we use any arbitrary PSD kernel K . Then the result follows from Lemma 3. \square

Finally, we give the proof of the generalization bound in terms of the Discrepancy for the agnostic setting. Note that this proof was already given by Cortes et al. (2019), here we repeat their proof for completeness.

Proof of Theorem 3 Since $l(h(x), f(x)) \leq C$, we have that the squared loss l is μ -admissible (Cortes et al. 2019) with $\mu = 2C$, meaning that

$$|l(h(x), f(x)) - l(h'(x), f(x))| \leq 2C|h(x) - h'(x)| \tag{19}$$

holds for all $h, h' \in H$ and any $f : \mathcal{X} \rightarrow \mathcal{Y}$. Let \tilde{f} be any arbitrary element from H . By adding and subtracting terms and applying the triangle inequality, we can show that

$$\begin{aligned} |L_{\hat{P}}(h, f) - L_{\hat{Q}}(h, f)| &\leq |L_{\hat{P}}(h, \tilde{f}) - L_{\hat{Q}}(h, \tilde{f})| + |L_{\hat{P}}(h, f) - L_{\hat{P}}(h, \tilde{f})| \\ &\quad + |L_{\hat{Q}}(h, \tilde{f}) - L_{\hat{Q}}(h, f)|. \end{aligned}$$

The first term on the right hand side is by definition bounded by the Discrepancy. For the second term we can show

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

The first inequality follows from applying (19) to each summand. We can bound the third term in the same way, since $\hat{Q} \in \hat{P}$. Bounding the first term using the Discrepancy and the last two terms with the bound above we find

$$|L_{\hat{P}}(h, f) - L_{\hat{Q}}(h, f)| \leq \text{disc}(\hat{P}, \hat{Q}) + 2\mu \max_{x \in \hat{P}} |f(x) - \tilde{f}(x)|$$

holds for all $\tilde{f} \in H$. The result follows from minimizing the right hand side with respect to \tilde{f} , bounding $L_{\hat{P}}(h, f) - L_{\hat{Q}}(h, f)$ with its absolute value and reordering terms. \square

B Proofs

B.1 Proof of agnostic MMD worst case bound (Proposition 1)

Proof of Proposition 1 Let \tilde{l} be any element from $H_{\mathcal{L}}$ and define $g_{\hat{P}} = \frac{1}{n_{\hat{P}}} \sum_{x \in \hat{P}} g(h, f)(x)$ and define $g_{\hat{Q}}$ similarly. Define $\tilde{l}_{\hat{P}} = \frac{1}{n_{\hat{P}}} \sum_{x \in \hat{P}} \tilde{l}(x)$ and $\tilde{l}_{\hat{Q}}$ analogously. Using the triangle inequality we can show

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

The first term is bounded by the MMD, see (12). For the second term we have $|g_{\hat{P}} - \tilde{l}_{\hat{P}}| \leq \max_{x \in \hat{P}} |g(h, f)(x) - \tilde{l}(x)|$. This bound also holds also for the third term since $\hat{Q} \in \hat{P}$. Bounding the second and third term and maximizing over $h \in H$ we find that

$$|L_{\hat{P}}(h, f) - L_{\hat{Q}}(h, f)| \leq \text{MMD}(\hat{P}, \hat{Q}) + 2 \max_{h \in H, x \in \hat{P}} |g(h, f)(x) - \tilde{l}(x)|$$

holds for any $\tilde{l} \in H_{\mathcal{L}}$ and any $h \in H$. The result follows by choosing \tilde{l} to minimize the right hand side, bounding $L_{\hat{P}}(h, f) - L_{\hat{Q}}(h, f)$ by the bound on its absolute value and reordering terms. \square

B.2 Adjusting the MMD to the loss and hypothesis set (Theorem 2, Corollary 1)

In the main text we have given a sketch of the proof for the linear kernel. Here, we show a rigorous proof for the linear kernel, and afterward we give the proof for any arbitrary kernel K . The technique of the proof stays the same for any arbitrary kernel K , however, we have to do more bookkeeping.

Theorem 10 (Adjusted MMD linear kernel) *Let l be the squared loss and assume $f \in H$ (realizable setting), furthermore assume K is the linear kernel, $K(x_i, x_j) = x_i^T x_j$. If $K_{\mathcal{L}}(x_i, x_j) = K(x_i, x_j)^2$ and $\Lambda_{\mathcal{L}} = 4\Lambda^2$, then $g(h, f) \in H_{\mathcal{L}}$ and thus $\eta_{\text{MMD}} = 0$.*

Proof of Theorem 10 Let $u = h - f$. Fix h and f , then we will write $g(x)$ as shorthand for $g(h, f)(x) = l(h(x), f(x))$. Then $g(x) = u(x)^2$. Since K is the linear kernel, we have

Table 3 This table illustrates the notation used when 2 kernels are involved

Transformation	ψ_K	$\psi_{K'}$
Space	$\mathcal{X} \rightarrow \mathcal{H}$	$\rightarrow \mathcal{H}'$
Kernel		$K \rightarrow K'$

that $h(x) = h^T x$, $f(x) = f^T x$ and $u(x) = u^T x$. Furthermore, we have $\mathcal{H} = \mathcal{X}$, and thus $\psi_K(x) = x$. Furthermore, $\psi_{K_{\mathcal{L}}} : \mathcal{H} \rightarrow \mathcal{H}_{\mathcal{L}}$ is given by (Shawe-Taylor and Cristianini 2004, chap. 9.1)⁸:

$$\psi_{K_{\mathcal{L}}}(x) = (x_1^2, x_2^2, \sqrt{2}x_1x_2, x_3^2, \sqrt{2}x_1x_3, \sqrt{2}x_2x_3, x_4^2, \sqrt{2}x_1x_4, \sqrt{2}x_2x_4, \sqrt{2}x_3x_4, \dots) \tag{20}$$

Since the featuremap of $K_{\mathcal{L}}$ exists, it is a PSD kernel, meaning that $K_{\mathcal{L}}(x, x') = \langle \psi_{K_{\mathcal{L}}}(x), \psi_{K_{\mathcal{L}}}(x') \rangle_{K_{\mathcal{L}}}$. Then we can write

$$g(x) = (u^T x)^2 = K(u, x)^2 = K_{\mathcal{L}}(u, x) = \langle \psi_{K_{\mathcal{L}}}(u), \psi_{K_{\mathcal{L}}}(x) \rangle_{K_{\mathcal{L}}}$$

thus $g \in \mathcal{H}_{\mathcal{L}}$ with $g = \psi_{K_{\mathcal{L}}}(u)$. Now what remains to show is that $g \in H_{\mathcal{L}}$, or with other words, that $\|g\|_{K_{\mathcal{L}}} \leq 4\Lambda^2$. We can show that

$$\|g\|_{K_{\mathcal{L}}} = \langle \psi(u), \psi(u) \rangle_{K_{\mathcal{L}}} = K_{\mathcal{L}}(u, u) = (u^T u)^2 = \|u\|_K^2 \leq 4\Lambda^2$$

where the last step follows from that $h, f \in H$, and therefore $\|u\|_K \leq 2\Lambda$. This shows that $g(h, f) \in H_{\mathcal{L}}$, therefore $\eta_{\text{MMD}} = 0$. □

Before we prove the more general case for any kernel K , let us introduce some additional notation. Also, before we show the proof for the kernel $K_{\mathcal{L}}$, we first do the proof for the kernel K' which is slightly simpler, later we extend the result to $K_{\mathcal{L}}$. We define the squared kernel K' as:

$$K'(f, h) = \langle f, h \rangle_K^2 \tag{21}$$

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 of a PSD kernel is known to be PSD (Mohri et al. 2012, Theorem 5.3). Now we have two kernels we have two featuremaps: $\psi_K(x) : \mathcal{X} \rightarrow \mathcal{H}$ and $\psi_{K'} : \mathcal{H} \rightarrow \mathcal{H}'$. Note that the second featuremap can still be computed with (20). See Table 3 for an overview of the notation used.

Recall that because K is PSD kernel we have that:

$$K(x, x') = \langle \psi_K(x), \psi_K(x') \rangle_K \tag{22}$$

For $x, x' \in \mathcal{X}$. Similarly for the kernel K' which is also PSD we have that:

$$K'(f, g) = \langle \psi_{K'}(f), \psi_{K'}(g) \rangle_{K'} \tag{23}$$

For $f, g \in \mathcal{H}$. Again we define u as:

$$u = h - f$$

⁸ Note that actually in (Shawe-Taylor and Cristianini 2004) 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 (20). This is often referred to as the squared kernel.

Theorem 11 (Adjusted MMD for K') *Let l be the squared loss and assume $f \in H$ (realizable setting), and assume K is a PSD kernel. If $K'(f, h) = \langle f, h \rangle_K^2$ and $\Lambda_{\mathcal{L}} = 4\Lambda^2$, then $g(h, f) \in H' = \{h \in \mathcal{H}' : \|h\|_{K'} \leq \Lambda_{\mathcal{L}}\}$ where \mathcal{H}' is the RKHS of K' .*

Proof We have $g(x) = u(x)^2$. Since $h, f \in \mathcal{H}$, $u \in \mathcal{H}$ and thus

$$u(x) = \langle u, \psi_K(x) \rangle_K$$

The first step is to show that the function $g \in \mathcal{H}'$. By definition:

$$g(x) = u(x)^2 = \langle u, \psi_K(x) \rangle_K^2$$

Now we can easily recognize our definition of K' in this equation (compare with (21)), thus we note that:

$$g(x) = K'(u, \psi_K(x)) = \langle \psi_{K'}(u), \psi_{K'}(\psi_K(x)) \rangle_{K'}$$

Where the second equality is obtained by applying (23). We observe that g corresponds to the vector $\psi_{K'}(u) \in \mathcal{H}'$, and thus we have that $g \in \mathcal{H}'$.

The second step is to show that $g \in H'$, with other words that $\|g\|_{K'} \leq 4\Lambda^2$. Since $g = \psi_{K'}(u) \in \mathcal{H}'$ the norm of g in K' is given by

$$\|g\|_{K'}^2 = \langle \psi_{K'}(u), \psi_{K'}(u) \rangle_{K'}$$

Now we can use (23) to rewrite this in terms of K' . We obtain:

$$\|g\|_{K'}^2 = K'(u, u) \tag{24}$$

Using the definition of K' we find:

$$K'(u, u) = \langle u, u \rangle_K^2 = \|u\|_K^4 \tag{25}$$

Now since $\|h\|_K \leq \Lambda$ and $\|f\|_K \leq \Lambda$ since $h, f \in H$, we have

$$\|u\|_K = \|h - f\|_K \leq 2\Lambda$$

Combining this with Eqs. 24 and 25 we find that:

$$\|g\|_{K'} = \|u\|_K^2 \leq 4\Lambda^2$$

Thus we have shown that $g \in \mathcal{H}'$. □

However, do we now have $g \in H_{\mathcal{L}}$? As of now we defined the kernel $K'(f, h)$ so that it operates on $f, g \in \mathcal{H}$. This does not coincide with the kernel $K_{\mathcal{L}}$. Therefore, we will now argue that $\mathcal{H}' = \mathcal{H}_{\mathcal{L}}$, and thus that in general the result generalizes to any kernel K , proving Theorem 2.

Proof of Theorem 2 By definition of $K_{\mathcal{L}}$ we have that:

$$K_{\mathcal{L}}(x, x') = K(x, x')^2$$

Now using (22) we can show that:

$$K_{\mathcal{L}}(x, x') = K(x, x')^2 = \langle \psi_K(x), \psi_K(x') \rangle_K^2$$

Observe that this coincides with the definition of K' (21), thus we can write this as:

$$K_{\mathcal{L}}(x, x') = \langle \psi_K(x), \psi_K(x') \rangle_K^2 = K'(\psi_K(x), \psi_K(x'))$$

Now using (23) we can write this as:

$$K_{\mathcal{L}}(x, x') = K'(\psi_K(x), \psi_K(x')) = \langle \psi_{K'}(\psi_K(x)), \psi_{K'}(\psi_K(x')) \rangle_{K'}$$

in other words, we see that the kernel product of $K_{\mathcal{L}}$ can be computed in the RKHS of the kernel K' . Thus, the RKHS of K' and $K_{\mathcal{L}}$ coincide! Thus we have that $\mathcal{H}' = \mathcal{H}_{\mathcal{L}}$. Therefore, Theorem 11 implies that we can generalize all results in terms of K' to the kernel $K_{\mathcal{L}}$. Therefore, g is also in the RKHS of $K_{\mathcal{L}}$, and in particular we have that $g \in H_{\mathcal{L}}$, and therefore $\eta_{\text{MMD}} = 0$. \square

Remark 1 Another way to understand this is to see that the featuremap of $K_{\mathcal{L}}$ is given by $\psi_{K_{\mathcal{L}}}(x) = \psi_{K'}(\psi_K(x))$ and thus maps to the space \mathcal{H}' , and from this it follows that $\mathcal{H}' = \mathcal{H}_{\mathcal{L}}$.

Proof of Corollary 1 Theorem 2 tells us to choose $K_{\mathcal{L}}(x, x') = K(x, x')^2$ to obtain $\eta_{\text{MMD}} = 0$. We can show

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

where we absorbed the factor of 2 in the exponent in $\sigma_{\mathcal{L}}$, so $\sigma_{\mathcal{L}} = \frac{\sigma}{\sqrt{2}}$. \square

B.3 MMD Computation (Theorem 5 and Corollary 2)

First we prove the Theorem 5 in case K is the linear kernel for $d = 2$, afterward we extend the proof to any dimension, and finally we prove Theorem 5 for any PSD kernel.

Theorem 12 (MMD Computation linear kernel $d = 2$) *Let $K_{\mathcal{L}}(x_i, x_j) = K(x_i, x_j)^2$ and $\Lambda_{\mathcal{L}} = 4\Lambda^2$. Furthermore, assume K is the linear kernel, $K(x_i, x_j) = x_i^T x_j$ and $d = 2$, then*

$$\text{MMD}(\hat{P}, \hat{Q}) = 4\Lambda^2 \|\lambda\|_2. \tag{26}$$

Proof If K is the linear kernel, $\mathcal{H} = \mathcal{X}$ and $K_{\mathcal{L}}$ defines a featuremap $\psi_{K_{\mathcal{L}}}(x) : \mathcal{X} \rightarrow \mathcal{H}_{\mathcal{L}}$ which is given by $\psi_{K_{\mathcal{L}}}(x) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)$, see (Shawe-Taylor and Cristianini 2004, chap. 9.1). From (11) we find

$$\text{MMD}(\hat{P}, \hat{Q}) = \Lambda_{\mathcal{L}} \|\mu_{\hat{P}} - \mu_{\hat{Q}}\|_{K_{\mathcal{L}}}$$

where $\mu_{\hat{P}} = \frac{1}{n_{\hat{P}}} \sum_{x \in \hat{P}} \psi_{K_{\mathcal{L}}}(x)$ and $\mu_{\hat{Q}}$ is analogously defined. Now using the fact $\mu_{\hat{P}} - \mu_{\hat{Q}} = (M_{11}, M_{22}, \sqrt{2}M_{12})^T$ and some algebra we can show

$$\text{MMD}(\hat{P}, \hat{Q}) = 4\Lambda^2 \|M\|_F = 4\Lambda^2 \|\lambda\|_2$$

where the second equality follows since M is a real symmetric matrix and thus its eigenvalues are equal to its singular values (up to a sign change). \square

Let us first generalize Theorem 12 to any arbitrary dimension d , afterward we extend the result to any kernel K .

Theorem 13 (MMD Computation linear kernel arbitrary dimension) *If $K_{\mathcal{L}}(x_i, x_j) = K(x_i, x_j)^2$ and $\Lambda_{\mathcal{L}} = 4\Lambda^2$. Furthermore, assume K is the linear kernel, $K(x_i, x_j) = x_i^T x_j$ and the dimension of the input space \mathcal{X} is d , then*

$$\text{MMD}(\hat{P}, \hat{Q}) = 4\Lambda^2 \|\lambda\|_2.$$

Proof We can show using (20) that the vector $\mu_{\hat{P}} - \mu_{\hat{Q}}$ becomes

$$(\mu_{\hat{P}} - \mu_{\hat{Q}})^T = (M_{11}, M_{22}, \sqrt{2}M_{12}, M_{33}, \sqrt{2}M_{13}, \sqrt{2}M_{23}, M_{44}, \sqrt{2}M_{14}, \sqrt{2}M_{24}, \sqrt{2}M_{34}, \dots)^T.$$

Observe that the entry M_{ii} appears only once in $\mu_{\hat{P}} - \mu_{\hat{Q}}$, and any entry M_{ij} where $i \neq j$, appears as $\sqrt{2}M_{ij}$. Furthermore, note that the diagonal M_{ii} only occurs once in the matrix M . However, any element M_{ij} appears twice in M (since M is symmetric). Therefore

$$(\mu_{\hat{P}} - \mu_{\hat{Q}})^T (\mu_{\hat{P}} - \mu_{\hat{Q}}) = \sum_i M_{ii}^2 + 2 \sum_{i \neq j} M_{ij}^2,$$

is a sum of all entries (squared) of M , and therefore

$$\|\mu_{\hat{P}} - \mu_{\hat{Q}}\|_{K_{\mathcal{L}}} = \|M\|_F,$$

as before. The rest of the proof is identical to the proof of Theorem 12 ($d = 2$). □

Note that due to our careful ordering of the featuremap of the squared kernel, given in (20), we have that this featuremap is still properly defined even if the dimension of $d \rightarrow \infty$, such as for a Gaussian kernel. Now we are ready to prove Theorem 5.

Proof of Theorem 5 To show the result holds for any arbitrary kernel K , we have to work in the RKHS of K , thus everywhere x needs to be replaced by $z(x)$ as in Appendix A.2, and then we replace M with M_Z . Then Theorem 13 still holds, since the featuremap $\psi(x)_{K'}$ is still given by the featuremap of the squared kernel, (20), however in this case the featuremap is with respect z instead of x . This does not influence the proof. We showed that this holds for any dimension d , and the featuremap still exists if $d \rightarrow \infty$, thus our results hold for a kernel K with arbitrary dimension of the RKHS \mathcal{H} .

For the Gaussian kernel we cannot compute the matrix M . Instead, since M and M_K have the same eigenvalues, see Lemma 3, we can compute the MMD instead using the eigenvalues of M_K , as in Appendix A.2. □

Remark 2 Note that

$$\text{MMD}(\hat{P}, \hat{Q}) \neq 4\Lambda^2 \|M_K\|_F.$$

These are not equal, since the matrix M_K is not symmetric. Therefore, the eigenvalues of M_K are not the same as the singular values of M_K (as was the case for M , which is symmetric).

Proof of Corollary 2 Comparing Eqs. 18 and 6 and noting $\|\lambda\|_\infty \leq \|\lambda\|_2$ shows the result. □

B.4 Probabilistic analysis (Lemma 1, Proposition 2 and Theorem 6)

Proof of Lemma 1 We can show that:

$$\left| L_{\hat{P}}(h, h') - L_{\hat{Q}}(h, h') \right| = \left| \sum_{i=1}^r \tilde{u}_i^2 \lambda_i \right| \leq \sum_{i=1}^r \tilde{u}_i^2 |\lambda_i|$$

where the equality follows from Eq. 13 and the inequality follows from the triangle inequality. Next, bound $L_{\hat{P}}(h, h') - L_{\hat{Q}}(h, h')$ using the bound above and reorder terms. The result is obtained after applying the expectation w.r.t. u on both sides and applying the linearity of the expectation. □

Proof of Proposition 2 Computing $G(2\Lambda e_1, M)$ which will be found to be exactly equal to the Discrepancy. After combining this fact with Lemma 1 will result in the desired equality, the inequality follows from Corollary 2. \square

Proof of Theorem 6 We can show that

$$\mathbb{E}_u G(u, M) = \frac{1}{\sqrt{r}} \text{MMD}(\hat{P}, \hat{Q}) \leq \text{disc}(\hat{P}, \hat{Q}). \tag{27}$$

This equality can be shown by working out the expectation and canceling terms and recognizing the definition of the MMD from (6). The inequality follows from $\|\lambda\|_2 \leq \sqrt{r}\|\lambda\|_\infty$. The final result follows by combining Eq. 27 with Lemma 1. \square

B.5 Proof of nuclear discrepancy bound (Theorem 7)

Before we can show the proof of the Nuclear Discrepancy bound, we need the following lemma:

Lemma 4 *Let $p(u)$ be uniform over all $u \in U$. Then*

$$\mathbb{E}_u \bar{u}_1^2 = \frac{4\Lambda^2}{r+2}.$$

Proof By comparing the volume of a sphere of radius 2Λ and the volume of a sphere of radius $w = \|u\|_2$, we can show that for this distribution $p(u)$ we have that

$$p(w) = \frac{w^{(r-1)r}}{(2\Lambda)^r}.$$

Then it is straightforward to show that

$$\mathbb{E}_u \|u\|_2^2 = \frac{r}{r+2} 4\Lambda^2.$$

by integration of $p(w)$. From the symmetry of $p(u)$ it follows that $\mathbb{E}_u \bar{u}_1^2 = \mathbb{E}_u \bar{u}_i^2$ for all i . From this fact and the linearity of the expectation the result follows. \square

Proof of Theorem 7 We can show that

$$\mathbb{E}_u G(u, M) = \sum_i |\lambda_i| \mathbb{E}_u \bar{u}_i^2 = \|\lambda\|_1 \mathbb{E}_u \bar{u}_1^2 = \frac{4\Lambda^2}{r+2} \|\lambda\|_1.$$

The first equality follows from switching expectation and sum. The second equality follows from symmetry of $p(u)$. The last equality follows from Lemma 4. The bound can be obtained by combining with Theorem 8. The inequalities follow from the vector norm inequalities $\|\lambda\|_1 \leq \sqrt{r}\|\lambda\|_2 \leq r\|\lambda\|_\infty$. \square

C Remark on probabilistic analysis and choice of U_s

The remark in this section will explain why instead of U , we need to take U_s (to be defined below). The problem stems from the fact that if we choose $p(u)$ uniform on U , it may seem unclear what it means for u to be randomly sampled from an infinite dimensional sphere uniformly.

We will use the notation of Appendix A.2, since we will work with a kernel K with a high-dimensional \mathcal{H} , in order to highlight the problem that a lot of eigenvalues may be zero. We are analyzing what happens to

$$\mathbb{E}_u G(u, M_Z) \tag{28}$$

for arbitrary distributions, for example, for the uniform distribution $p(u)$, in case the length of a vector $z(x)$ is infinite such as with a Gaussian kernel (then M has infinite eigenvalues, but only r are non-zero).

The RKHS of K, \mathcal{H} , is then of infinite dimension. We split \mathcal{H} in two parts: $\mathcal{H}_s = \text{span}(Z_{\hat{\rho}})$, and its orthogonal complement \mathcal{H}_s^\perp . Then for any vector in $a \in \mathcal{H}$, $a = a_s + a_s^\perp$, where $a_s \in \mathcal{H}_s$ and $a_s^\perp \in \mathcal{H}_s^\perp$. In particular we have that after training a kernel regularized model, we have $h \in a_s$ due to the regularization term in the training procedure. Furthermore, for any observed f , we have

$$L_{\hat{\rho}}(h, f) = L_{\hat{\rho}}(h, f_s), \tag{29}$$

since $Z_{\hat{\rho}} f_s^\perp = 0$. The same thing holds for $L_{\hat{\rho}}(h, f)$. Therefore, we may consider it redundant to consider f, h , and we may limit our analysis to f_s and h_s . In addition we have that

$$G(u, M_Z) = G(u_s, M_Z), \tag{30}$$

since u is projected on eigenvectors of M_Z , and only eigenvectors in $\text{span}(Z_{\hat{\rho}})$ have non-zero eigenvalue, and thus u_s^\perp only has components that correspond to eigenvalues that are zero. Thus any u has the same objective as the corresponding u_s . Therefore, instead of defining a pdf over U , we define a pdf over $U_s = \{u \in \mathcal{H}_s : \|u\|_K \leq 2A\}$. Then by construction the dimension of u is at most $r = \text{rank}(M_Z) \leq n_{\hat{\rho}}$, which is always finite. Then sampling u is a well defined procedure even in infinite dimensional RKHS.

D Computation of the decomposition of the probabilistic bounds

To compute each term of $G(u, M_Z)$, we can compute the eigendecomposition of M_K to compute the eigenvalues, however we also need to know \bar{u}_i for each i . The computation of \bar{u}_i , the projection of u onto the eigenvector v_i of M_Z is non-trivial to compute in case kernels are used. Observe that here v_i is the i th eigenvector and not a component. Here we assume v_i is not normalized to unit norm (which is why we write it differently from e_i). We give a detailed description in this appendix how to compute \bar{u}_i . In this case the equation for \bar{u}_i is:

$$\bar{u}_i = \frac{u^T v_i}{\sqrt{v_i^T v_i}} \tag{31}$$

The difficulty in this derivation is finding the vector v_i in case kernels are used. Then we need to find v_i expressed in terms of the datamatrix Z . Then we can apply the ‘kernel trick’ to compute (31).

By the eigenvalue equation of M_Z we have:

$$M_Z v_i = \lambda_i v_i \tag{32}$$

In case of the linear kernel it is straightforward to compute v_i . However, to compute v_i for any K , we have to take extra steps. First we show that v_i can be expressed in terms of the datamatrix Z , and afterward we find this expression of v_i in terms of Z . Note that:

$$M_Z v_i = \sum_{j=1}^{n_{\hat{P}}} d_j z_j z_j^T v_i = \sum_{j=1}^{n_{\hat{P}}} (z_j^T v_i) d_j z_j = \lambda_i v_i$$

Thus we have that:

$$\sum_{j=1}^{n_{\hat{P}}} \frac{(z_j^T v_i) d_j}{\lambda_i} z_j = v_i$$

Thus we have that each eigenvector v_i is a linear combination of the vectors z_j . Here the sum is taken over all objects $z(x) : x \in \hat{P}$. Since $\hat{Q} \subseteq \hat{P}$, this includes all data the active learner has access to. Then we can write each eigenvector v_i as:

$$v_i = Z_{\hat{P}}^T \alpha_i \tag{33}$$

Thus we can express each vector v_i using the datamatrix $Z_{\hat{P}}$. Now we will have to find the vector α_i to find v_i . We substitute the equation above in (32) to obtain:

$$M_Z Z_{\hat{P}}^T \alpha_i = \lambda_i Z_{\hat{P}}^T \alpha_i$$

Now we multiply left with $DZ_{\hat{P}}$ on both sides to obtain:

$$DZ_{\hat{P}} M_Z Z_{\hat{P}}^T \alpha_i = \lambda_i DZ_{\hat{P}} Z_{\hat{P}}^T \alpha_i$$

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 (17). Now we define $\beta_i = M_K^T \alpha_i$. Then we find:

$$M_K^T \beta_i = \lambda_i \beta_i \tag{34}$$

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 (34) by $(M_K^T)^{-1}$ on both sides we obtain:

$$\beta_i = \lambda_i (M_K^T)^{-1} \beta_i \tag{35}$$

Now observe that due to the definition of β_i we have that:

$$\beta_i (M_K^T)^{-1} = \alpha_i \tag{36}$$

Combining (35) and (36) we find that:

$$\alpha_i = \frac{\beta_i}{\lambda_i}$$

Substituting this in (33) we find the vector v_i :

$$v_i = Z_{\hat{P}}^T \frac{\beta_i}{\lambda_i} \tag{37}$$

Now we have found v_i . Now we can proceed to compute u_i .

Note that due to the representer theorem we have that the hyperplane of each model can be written as a linear combination of the data:

$$u = f - h = Z_{\hat{D}}^T c' - Z_{\hat{Q}}^T c \equiv Z_{\hat{D}}^T \tilde{c} \tag{38}$$

Here f is given as a linear combination of $Z_{\hat{D}}$, which we define as the complete datamatrix. This datamatrix includes the training and test set, since f in our experiments was obtained by training on the whole dataset where the original binary labels of the dataset are used (in the realizable setting). However note that for any $f \in H$ the model f can be written in this way. Similarly, since h is trained on the dataset \hat{Q} , we can write h as a linear combination of objects $Z_{\hat{Q}}$. Combining (37) and (38) with (31) we find that:

$$\bar{u}_i = \frac{\tilde{c} Z_{\hat{D}} Z_{\hat{P}}^T \frac{\beta_i}{\lambda_i}}{\sqrt{\frac{\beta_i^T}{\lambda_i} Z_{\hat{P}} Z_{\hat{P}}^T \frac{\beta_i}{\lambda_i}}} = \frac{\tilde{c} K_{\hat{D}\hat{P}} \beta_i}{\sqrt{\beta_i^T K_{\hat{P}} \beta_i}}$$

E Experimental settings and dataset characteristics

The active learning methods are evaluated on the datasets shown in Table 4. The datasets marked with * were provided by Cawley and Talbot (2004). Other datasets originate from the UCI Machine Learning repository (Lichman 2013), except the MNIST dataset (LeCun et al. 1998) which is a standalone dataset.

The parameter settings used are displayed in Table 5. To obtain these hyperparameters we repeated the following procedure multiple times. We randomly select 25 examples from the dataset and label these. We train a KRLS model on these samples and evaluate the MSE on all unselected objects. The hyperparameters that result in the best performance after averaging are used in the active learning experiments.

Table 4 Characteristics of evaluation datasets

Dataset	# Objects	# Positive class	Dimensionality
vehicles	435	218	18
heart	297	137	13
sonar	208	97	60
thyroid*	215	65	5
ringnorm*	1000	503	20
ionosphere	351	126	33
diabetes	768	500	8
twonorm*	1000	500	20
banana*	1000	439	2
german	1000	700	20
splice	1000	541	60
breast	699	458	9
mnist 3vs5	1000	484	784
mnist 7vs9	1000	510	784
mnist 5vs8	1000	535	784

Table 5 Table with parameters used for the benchmark datasets

Dataset	σ	$\log_{10}(\mu)$
vehicles	5.270	-3.0
heart	5.906	-1.8
sonar	7.084	-2.6
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
mnist 3vs5	44.215	-6.0
mnist 7vs9	44.215	-3.6
mnist 5vs8	44.215	-8.9

F Results of the agnostic setting

For completeness we discuss the results of the agnostic setting where the original binary labels are used. In this setting $\eta \neq 0$, but η will be small due to our choice of hyperparameters, and therefore we ignore it during active learning (since we also cannot estimate it unless we

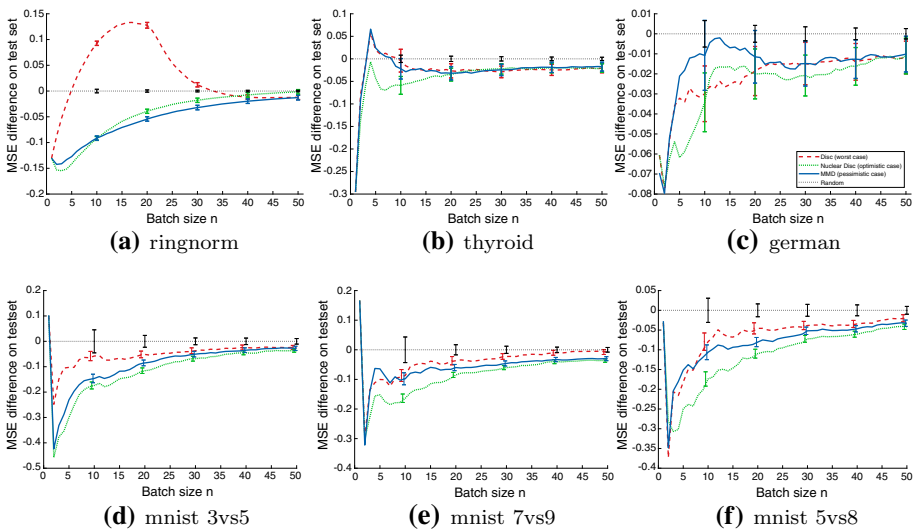


Fig. 3 Learning curves for several datasets for the agnostic setting. Results are averaged over 100 runs. Observe that compared to the realizable setting the variability of the performance increases and therefore performance differences become less significant. Due to unexpected effects of $\eta > 0$ the ranking of the methods may change

Table 6 Area Under the mean squared error Learning Curve (AULC) for the strategies in the agnostic setting, averaged over 100 runs

Dataset	Random	Discrepancy	MMD	Nuclear Discrepancy
vehicles	25.8 (4.7)	22.9 (2.7)	23.9 (3.2)	23.5 (2.6)
heart	34.9 (4.0)	32.7 (3.3)	32.4 (3.8)	32.4 (3.7)
sonar	40.6 (4.3)	39.8 (4.4)	38.3 (3.6)	37.3 (4.2)
thyroid	17.9 (3.4)	16.4 (3.5)	16.3 (3.1)	15.7 (2.9)
ringnorm	35.9 (1.4)	37.5 (0.7)	33.1 (1.0)	33.5 (1.0)
ionosphere	28.9 (3.4)	26.7 (2.5)	26.3 (2.7)	27.3 (3.4)
diabetes	40.5 (3.2)	39.6 (3.2)	39.7 (3.0)	40.2 (2.7)
twonorm	19.3 (2.4)	17.3 (1.6)	17.0 (1.6)	16.2 (1.3)
banana	32.1 (3.4)	28.5 (3.4)	28.6 (2.9)	27.8 (2.5)
german	42.2 (3.2)	40.8 (2.3)	41.1 (2.6)	40.6 (2.4)
splice	45.4 (3.1)	45.2 (3.5)	44.6 (2.8)	43.7 (2.6)
breast	11.7 (2.7)	10.3 (1.7)	10.1 (1.7)	10.1 (1.8)
mnist 3vs5	30.6 (4.5)	28.1 (2.5)	26.1 (2.2)	25.0 (1.8)
mnist 7vs9	27.5 (3.6)	25.5 (2.4)	24.6 (2.0)	23.2 (1.6)
mnist 5vs8	30.2 (3.4)	26.9 (2.7)	26.1 (2.3)	24.5 (2.1)

Bold indicates the best result, or results that are not significantly worse than the best result, according to a paired t-test ($p = 0.05$). Parenthesis indicate standard deviation

have the labels of \hat{P}). Several illustrating learning curves are shown in Fig. 3, all results are summarized in Table 6, all learning curves can be found in Appendix H.3.

The curves are less smooth and have larger standard errors compared to the realizable setting. Therefore the active learning methods are harder to distinguish which is reflected in Table 6 by larger standard deviations and more bold numbers in a single row. Observe that the ranking of the methods can also change, see for example the learning curve on ringnorm: in the realizable setting the ND improved upon the Discrepancy, while in the agnostic setting the reverse is the case for large budgets. In this setting, sometimes the Discrepancy performs the best. From Table 6 we can see that the trends observed in the realizable setting are still observed in the agnostic setting: the ND improves more upon the MMD than the reverse, however, the trend is weaker. This is likely the case because for this setting η_{MMD} and η_{disc} are non-zero, and therefore our theoretical analysis is weakened. Finally, observe that for the MNIST dataset, the learning curves and results as summarized by the AULC for 5vs8 are almost completely identical as in the realizable setting. Similarly, for 3vs5 differences are also quite small. This indicates that MNIST is very close to realizable with these found hyperparameter settings.

G Influence of subsampling on performance

We perform an additional experiment on the splice dataset to see how subsampling affects performance. To this end we measure the performance while we vary the pool size by changing the amount of subsampling. The subsampled pool is used as training set (this is the pool from which active learners can select queries, \hat{P}), all remaining samples are used as testset \hat{T} . Furthermore we use the same experimental protocol as for the other experiments.

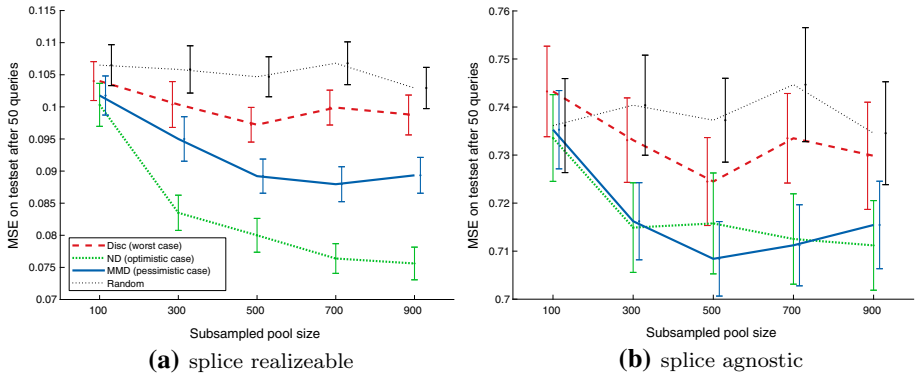


Fig. 4 Dataset size versus performance of the active learners on splice. We observe that for larger dataset sizes, the active learners typically improve with respect to random sampling, but the improvement levels off for large dataset sizes

We display the performance of the active learners in terms of MSE on the testset after 50 queries in Fig. 4 for both the realizeable and agnostic setting. The curve is averaged over 100 runs. Error bars represent the 95% confidence interval computed using the standard error. As expected, the trends in the realizeable setting are more clear, while due to model misspecification and other effects in the agnostic case performance differences are less clear due to larger standard deviations.

For small pool sizes all active learners experience a drop in performance. In this case the probability is large that ‘good’ queries may be missing because of an unlucky draw from the dataset. For larger pool sizes most active learners perform better. For larger pool sizes the performance at some point levels. At this point the pool contains sufficient representative samples for it to contain all possible ‘good’ queries the active learners will be looking for. The experiment provides evidence that if finer subsampling is used, methods typically improve in performance up to a point where performance levels off.

H Additional experimental results

H.1 Learning curves on all datasets for the realizable setting

(See Fig. 5.)

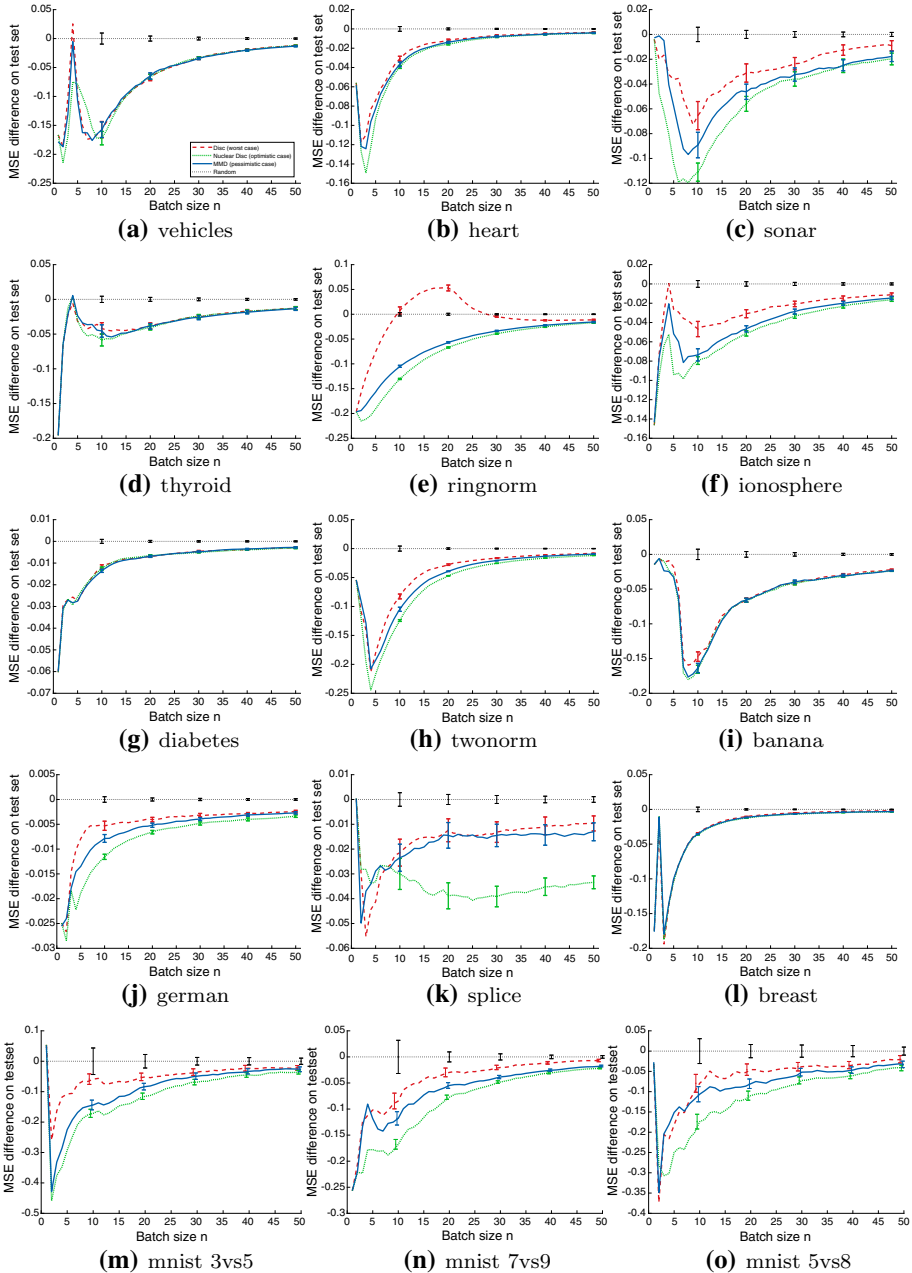


Fig. 5 Results on all benchmark datasets for the realizable setting

H.2 Decomposition of the probabilistic bounds for all datasets

(See Fig. 6.)

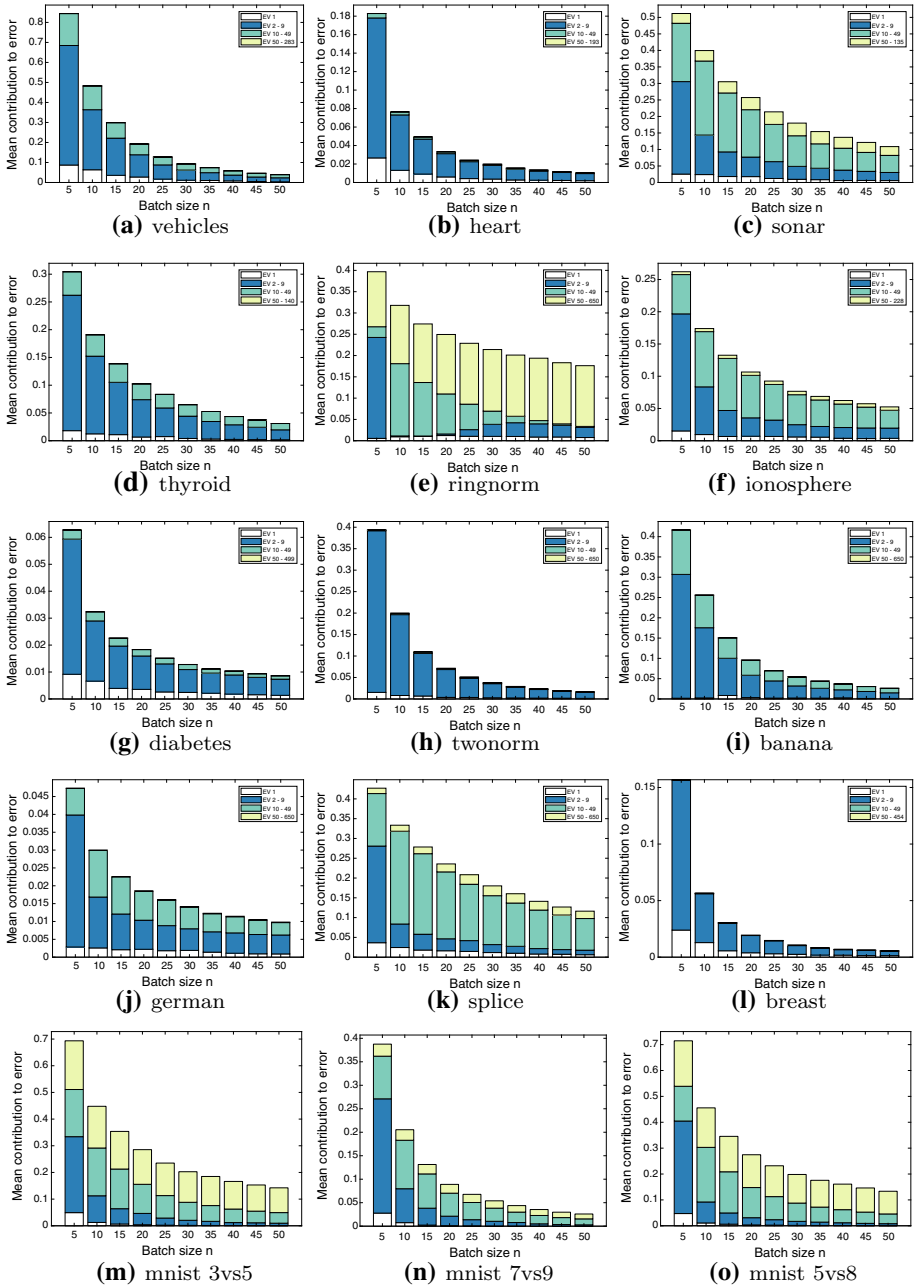


Fig. 6 Decomposition of the sum $G(u, M)$ during active learning for all datasets for the active learner ‘random sampling’

H.3 Learning curves on all datasets for the agnostic setting

(See Fig. 7.)

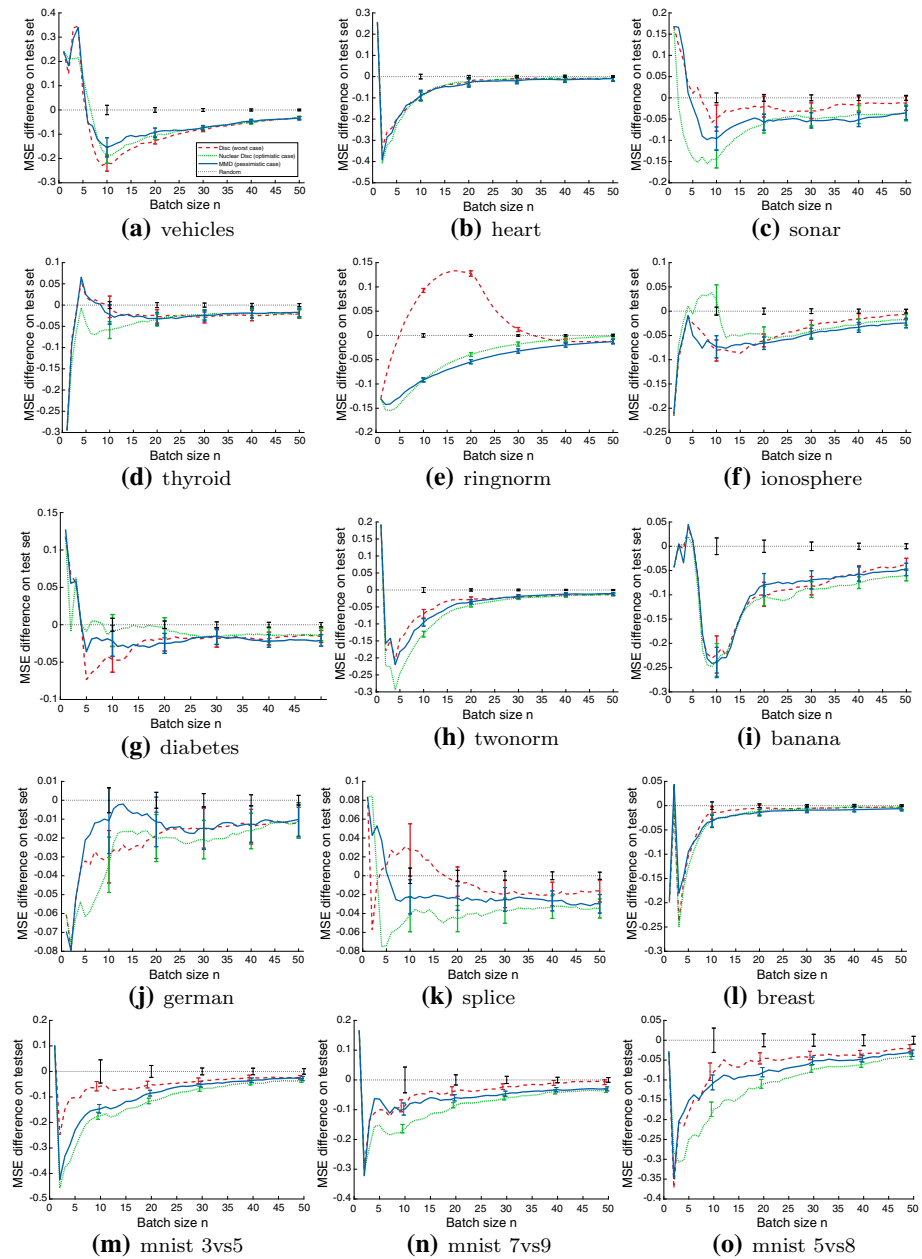


Fig. 7 Results on all benchmark datasets for the agnostic setting

References

- Attenberg, J., & Provost, F. (2011). Inactive learning? Difficulties employing active learning in practice. *ACM SIGKDD Explorations Newsletter*, 12(2), 36–41.
- Balcan, M.-F., & Urner, R. (2016). Active learning-modern learning theory. *Encyclopedia of algorithms*, pp. 8–13.
- Balcan, M.-F., Beygelzimer, A., & Langford, J. (2009). Agnostic active learning. *Journal of Computer and System Sciences*, 75(1), 78–89.
- Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., & Vaughan, J. W. (2010). A theory of learning from different domains. *Machine Learning*, 79(1), 151–175.
- Bod, Z., Minier, Z., & Cs, L. (2011). Active learning with clustering. *Active Learning and Experimental Design workshop in conjunction with AISTATS, 2010*, 127–139.
- Cawley, G. C., & Talbot, N. L. (2004). Fast exact leave-one-out cross-validation of sparse least-squares support vector machines. *Neural Networks*, 17(10), 1467–1475.
- Chattopadhyay, R., Wang, Z., Fan, W., Davidson, I., Panchanathan, S., & Ye, J. (2012). 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)*, pp. 741–749.
- Chattopadhyay, R., Fan, W., Davidson, I., Panchanathan, S., & Ye, J. (2013). Joint transfer and batch-mode active learning. *Proceedings of the 30th international conference on machine learning (ICML)*, pp. 253–261.
- Cohn, D., Atlas, L., & Ladner, R. (1994). Improving generalization with active learning. *Machine Learning*, 15(2), 201–221.
- Contardo, G., Denoyer, L., & Artieres, T. (2017). A meta-learning approach to one-step active learning. *arXiv preprint arXiv:1706.08334*
- Cortes, C., & Mohri, M. (2014). Domain adaptation and sample bias correction theory and algorithm for regression. *Theoretical Computer Science*, 519, 103–126.
- Cortes, C., Mohri, M., & Medina, A. M. (2019). Adaptation based on generalized discrepancy. *Journal of Machine Learning Research*, 20(1), 1–30.
- Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., et al. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639), 115.
- Ganti, R., & Gray, A. (2012). UPAL: Unbiased Pool Based Active Learning. In *Proceedings of the 15th international conference on artificial intelligence and statistics (AISTATS)*, pp. 422–431
- Germain, P., Habrard, A., Laviolette, F., & Morvant, E. (2013). A pac-bayesian approach for domain adaptation with specialization to linear classifiers. In *Proceedings of the 30th international conference on machine learning (ICML)*, pp. 738–746.
- Gretton, A., Borgwardt, K. M., Rasch, M. J., Scholkopf, B., & Smola, A. (2012). A Kernel two-sample Test. *Machine Learning Research*, 13(1), 723–773.
- Gu, Q., & Han, J. (2012). Towards Active Learning on Graphs: An error bound minimization approach. In *Proceedings of the 12th IEEE international conference on data mining (ICDM)*, pp. 882–887
- Gu, Q., Zhang, T., Han, J., & Ding, C. H. (2012). Selective labeling via error bound minimization. In *Proceedings of the 25th conference on advances in neural information processing systems (NIPS)*, pp. 323–331
- Gu, Q., Zhang, T., & Han, J. (2014). Batch-mode active learning via error bound minimization. In *Proceedings of the 30th conference on uncertainty in artificial intelligence (UAI)*.
- Hanneke, S. (2007). A bound on the label complexity of agnostic active learning. In *Proceedings of the 24th international conference on machine learning (ICML)*, pp. 353–360.
- Harpale, A.S., & Yang, Y. (2008). Personalized active learning for collaborative filtering. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 91–98. ACM
- Hoi, S. C., Jin, R., Zhu, J., & Lyu, M. R. (2006). Batch mode active learning and its application to medical image classification. In *Proceedings of the 23rd international conference on machine learning (ICML)*, pp. 417–424.
- Hu, R., Namee, B. Mac & S. J. Delany (2010). Off to a good start: Using clustering to select the initial training set in active learning. In *FLAIRS conference*
- Huang, J., Smola, A. J., Gretton, A., Borgwardt, K. M., & Scholkopf, B. (2007). Correcting sample selection bias by unlabeled data. In *Proceedings of the 19th conference on advances in neural information processing systems (NIPS)*, pp. 601–608.
- Huang, S.-j., Jin, R., & Zhou, Z.-H. (2010). Active learning by querying informative and representative examples. In *Proceedings of the 23th conference on advances in neural information processing systems (NIPS)*, pp. 892–900.

- Huijser, M., & van Gemert, J. C. (2017). Active decision boundary annotation with deep generative models. In *Proceedings of the IEEE international conference on computer vision*, pp. 5286–5295.
- Jaynes, E. T. (1957). Information theory and statistical mechanics. *Physical Review*, 106(4), 620.
- LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324.
- Lichman, M. (2013). UCI Machine Learning Repository. URL <http://archive.ics.uci.edu/ml>.
- Loog, M., & Yang, Y. (2016). An empirical investigation into the inconsistency of sequential active learning. In *2016 23rd international conference on pattern recognition (ICPR)*, pp. 210–215. IEEE.
- Mansour, Y., Mohri, M., & Rostamizadeh, A. (2009). Domain adaptation: Learning bounds and algorithms. In *Proceedings of the 22nd annual conference on learning theory (COLT)*.
- Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2012). *Foundations of Machine Learning*. Cambridge, Massachusetts: MIT press.
- Nguyen, H. T., & Smeulders, A. (2004) Active learning using pre-clustering. In *Proceedings of the 21st international conference on machine learning (ICML)*, p. 79.
- O’Neill, J., Delany, S. J., & MacNamee, B. (2017). Model-free and model-based active learning for regression. In *Advances in computational intelligence systems*, pp. 375–386. Springer
- Rifkin, R., Yeo, G., & Poggio, T. (2003). Regularized least-squares classification. *Advances in Learning Theory: Methods, Model, and Applications*, 190, 131–154.
- Settles, B. (2011). From theories to queries: Active learning in practice. *Active Learning and Experimental Design workshop In conjunction with AISTATS, 2010*, 1–18.
- Settles, B. (2012). Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1), 1–114.
- Settles, B., & Craven, M. (2008). An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the conference on empirical methods in natural language processing*, pp. 1070–1079. Association for Computational Linguistics.
- Shawe-Taylor, J., & Cristianini, N. (2004). *Kernel methods for pattern analysis*. Cambridge: Cambridge University Press.
- Tosh, C., & Dasgupta, S. (2017). Diameter-based active learning. *arXiv preprint arXiv:1702.08553*
- Urner, R., Wulff, S., & Ben-David, S. (2013). Plal: Cluster-based active learning. In *Conference on learning theory (COLT)*, pp. 376–397.
- Wang, Z., & Ye, J. (2013). 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)*, pp. 158–166.
- Xu, Z., Yu, K., Tresp, V., Xu, X., & Wang, J. (2003). Representative Sampling for Text Classification Using Support Vector Machines. In *Advances in Information Retrieval*, pages 393–407. Springer.
- Yang, Y., & Loog, M. (2018). A benchmark and comparison of active learning for logistic regression. *Pattern Recognition*, 83, 401–415.
- Yu, K., Bi, J., & Tresp, V. (2006). Active learning via transductive experimental design. In *Proceedings of the 23rd international conference on machine learning (ICML)*, pp. 1081–1088. <https://doi.org/10.1145/1143844.1143980>.
- Zhu, J., Wang, H., Yao, T., & Tsou, B. K. (2008). Active learning with sampling by uncertainty and density for word sense disambiguation and text classification. In *Proceedings of the 22nd international conference on computational linguistics-volume 1*, pp. 1137–1144.