Active Learning in Non-parametric and Federated Settings

by

Jonathan Richard Goetz

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Statistics)
in the University of Michigan
2020

Doctoral Committee:

Associate Professor Ambuj Tewari, Chair
Associate Professor Long Nguyen
Professor Yaacov Ritov
Associate Professor Paul Zimmerman

Jonathan Richard Goetz

jrgoetz@umich.edu

ORCID iD: 0000-0002-9954-9460

DEDICATION

To everyone who was there for me over these past 5 years

ACKNOWLEDGMENTS

I must start my expressions of gratitude with my adviser Ambuj Tewari. He exposed me to several machine learning topics with connection to computational chemistry, and allowed me to forge my own path through active learning, my topic of choice. Without his guidance, support, patience and understanding I would never have completed a single research project, let alone my thesis. I would also like to thank my other committee members. Long Nguyen taught my first theoretical machine learning class and opened my eyes to the more mathematical side of the subject. Yaacov Ritov continued my theory education, helping distill general understanding of statistics from some of its more obtuse technical areas. And Paul Zimmerman provided a model for interdisciplinary collaboration, drawing on expertise in both machine learning and chemistry to lead substantive discussions on the interplay between the two. I acknowledge the support NSF grant DMS-1646108 for providing three years of funding during my PhD, and thank Liza Levina and everyone else in the Department of Statistics who worked on this grant. The funding helped me focus on my research and gave me the bandwidth to mentor five undergraduate researchers during my PhD, which was an immensely rewarding experience. I also want to thank my close collaborator Josh Kammeraad for the many hours we spent discussing math, computer science, machine learning and chemistry, and for teaching me almost everything I know about the last of these.

I have been fortunate enough to have mentors beyond Michigan as well, and am thankful for their contributions to my growth as a researcher. Daniel Rabinowitz at Columbia helped me transition from undergraduate to graduate student. Rahul Mazumder at MIT took on the arduous task of teaching me how to do research in statistics. Kshitiz Malik at Facebook introduced me to federated learning, and taught me that engineering skills are not just complimentary to, but are in fact necessary for efficient and reliable empirical research.

I am eternally grateful for my parents, who supported my educational pursuits without limit. I was very lucky to pursue my PhD at Michigan with such a talented and wonderful group of Statistics PhD students. The fellowship and collaboration of the department made

these past five years genuinely enjoyable. In particular thanks to my closest friends in my cohort: Roger Fan, Young Hun Jung, Byoungwook Jang and Tim Lycurgus. Thank you for taking this journey with me. Finally I want to thank my love Emma Finder. You believed in me when I did not believe in myself, and without you none of what follows would exist.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

In many real world supervised learning problems, it is easy or cheap to acquire unlabelled data, but challenging or expensive to label it. Active learning aims to take advantage of this abundance of unlabelled data by sequentially selecting data points to label in an attempt to choose the best data points for the underlying prediction problem. In this thesis we present several contributions to the field of active learning.

The first part examines active learning for regression, an under studied topic compared with classification. We consider active learning for non-parametric regression, a particularly challenging problem since it is known that under standard smoothness conditions, the minimax rates for active and passive learning are the same. None-the-less we provide an active learning algorithm with provable improvement over passive learning when our underlying estimator is a purely random decision tree. We experimentally confirm that the gains can be substantial, and provide guidance for practitioners.

The second part returns to classification, but considers all weighted averaging estimators. Here we work to provide an extension of the celebrated Stone's Theorem for consistency under actively sampled data. We provide an augmentation that can be applied to a wide range of active learning algorithms, which allows us to replicate the results of Stone's Theorem in the noiseless case. However this only generalizes to the noisy case for some classical Stone estimators, whereas for others it can catastrophically fail. We explore the cause of this disjunctive behaviour and provide further conditions which exemplify why some estimators remain consistent while others do not.

The final part addresses the emerging area of federated learning. We study the the

problem of user selection during training, and expose the similarities to active learning. We then propose Active Federated Learning, which adapts techniques from active learning to this new setting, and show that the method can lead to reductions in the communication costs of training federated models by 20-70%.

CHAPTER I

Introduction

The intuition behind active learning is deceptively simple and extremely appealing; given a modelling process and some labelled data, certain new data points might be more valuable than other new data points. This difference in value can be the result of both underlying properties of the distribution being modelled, such as having non-uniform noise or complexity of the mean structure. It can also come from inadequacies in the existing sample, such as over/under-exploration of different regions of our covariate space, or even the presence of uncharacteristically high noise in some existing samples. Of course such detailed information of the underlying distribution is usually unknown. In this thesis we initially study active learning under very minimal assumptions on the distribution, and where the models being fit are also non-parametric. We also perform the first study of active learning applied to *federated learning*, an emerging distributed learning setting.

1.1 Motivation from computational chemistry

Our initial motivation for studying active learning came from computational chemistry (although none of our methods or results use any specific structure or assumptions specialized to this application). Historically chemistry research involved developing an understanding of groups of similar reactions through trial and error, working to describe the mechanism by which those reactions occurred, and developing that understanding into mechanistic rules for making predictions about reactions within the group. The development of Quantum Mechanics has in principle given chemists the ability to mathematically calculate most chemical properties of interest. By solving the Schrodinger equations (and various approximations of them) quantum chemists can obtain accurate and detailed descriptions of chemical properties from first principles (*ab initio*), i.e. without any experimental data. But getting solutions to these is highly resource intensive, becoming exponentially more expensive as your molecule size and desired accuracy increases. Not only are individual

simulations computationally expensive, but getting the answers to questions of practical interest can require more simulations than one might initially think. Take as an example one of the most basic questions in chemistry: when I mix two chemicals, what will happen? Even between two simple molecules there can many possible reactions (even just enumerating these potential reactions is non-trivial). The proportion with which these reactions occur is a function of their activation energy; the smaller this activation energy, the more likely it is a chemical reaction will occur at the micro level, meaning that often the reaction or few reactions with the lowest activation energy will be the only ones occurring in any relevant macro proportion. This activation energy is a property which can be calculated by solving (approximately) the Schrodinger wave equation, but using just quantum mechanics you would need to run simulations for all possible reactions. Further compounding this issue is the reality that many chemical reactions of interest are not single event but involve a chain of reactions, where at each stage of the chain there can be many possible reactions. It is easy to see how the exponential number of simulations needed can quickly become unmanageable.

One possible method of alleviating these issues is to use machine learning to avoid having to do all the simulations by building a predictive *metamodel* of quantities of interest. This model can be used to focus on reactions which are most promising. The models uses features of the reaction which chemists consider informative, and which can be easily obtained in an automatic and inexpensive fashion. This model requires completed simulations to use as labelled data, and the generation of a single labelled point can often be parallelized. This makes sequential generation of labelled data reasonable, so we can use active learning to produced the most accurate metamodel given a fixed budget of simulations.

1.2 Active learning overview

There are three main scenarios which dictate how the active learning algorithm can interact with unlabelled data: streaming settings, pool settings and membership query synthesis settings. In the streaming setting our covariates arrive one at a time and we have to choose whether to acquire the label or discard the data point. In the pool setting we are given a fixed set of training data, where we have all the covariates and have to pick data points out of this fixed set to label. Finally the membership query synthesis setting allows the algorithm to choose any point within the covariate space and sample a label for that point. The streaming setting is a well studied and interesting problem in its own right (see Settles (2010) and Fu et al. (2013) for excellent surveys of streaming based active learning and

active learning in general). However we will focus on the pool setting and membership query synthesis setting, as these are the settings which appear in the following chapters.

1.2.1 Pool setting

In the pool setting of active learning there are numerous paradigms for how to choose the next data point to label out of your set of unlabelled data. More formally we begin with n data points $\{X_i\}_1^n \in \mathcal{X}$ which are fixed and always available to the algorithm. For each X_i we have a corresponding $Y_i \in \mathcal{Y}$ and these Y_i are fixed (meaning we cannot sample the same point multiple times and receive multiple samples of the conditional distribution). Initially none or a very small fraction of these Y_i are known to the algorithm. Instead the algorithm has the ability to gain access to any of the Y_i, and the goal of the active learning algorithm is to sequentially select Y_i in order to optimize some objective subject to some constraints. The most common constraint is $|D| \leq k$, where D is the set of data points which are labeled, though generalizations of this which allow different label costs or additional constraints have been proposed (Golovin and Krause, 2011). The objective is usually to model the underlying function relating X_i and Y_i either globally or at the unlabelled data points, but this can be mathematically represented in a variety of ways. We will briefly review different approaches to pool based active learning, though again we refer interested readers to Settles (2010) for a more thorough overview. The majority of the work done in pool based active learning has been done in classification, since one of the archetypal uses for active learning has been to reduce the human labelling needed for common computer science tasks such as NLP (Settles and Craven, 2008), computer vision (Vijayanarasimhan and Grauman, 2009) and speech analysis (Tur et al., 2005).

The most intuitive method is called uncertainty sampling (Lewis and Gale, 1994) and it works exactly how it sounds. The idea is to sample in regions where you are most uncertain about the label on the data. For classification this uncertainty is usually measured using posterior probabilities (Scheffer et al., 2001), entropy (Körner and Wrobel, 2006) and margin distance (Tong and Koller, 2001). For regression this is measured using variance (Cohn et al., 1996), entropy (MacKay, 1992) and mutual information (Krause et al., 2008). This method is particularly popular when using Gaussian Processes, such as in Seo et al. (2000), since the variance at a given point is naturally given by the covariance matrix. Despite being intuitively motivated, it has proven more difficult than expected to analyze these methods theoretically. Only recently has there been successful work to provide guarantees for these types of active learning algorithms. In Golovin and Krause (2011) and Cuong et al. (2014) they use generalizations of submodularity maximization (Nemhauser

et al., 1978) show that specific variants of these methods enjoy near-optimality guarantees. And Mussmann and Liang (2018) showed that uncertainty sampling can be interpreted as performing a pre-conditioned stochastic gradient step on the zero-one loss.

Another similar but more theoretically motivated approach is called Query by Committee (QBC), first proposed in Seung et al. (1992). These methods follow the structure of statistical learning theory and assume there exists a finite set of possible hypotheses $\mathcal{H}$ which represent possible labellings of the data. Algorithms then select data points for which different hypotheses disagree more often, as knowing these data points will help us discriminate between the hypotheses in $\mathcal{H}$. This is studied in both the realizable and non-realizable case (Dasgupta et al., 2008; Nowak, 2008). QBC is much more amiable to theoretical analysis, as seen by the success of IWAL and it's variants (Beygelzimer et al., 2009). Hanneke and Yang (2015) provides general minimax bounds for active learning for a variety of noise conditions using methods based on Cohn et al. (1994). In practice either the full hypothesis space $\mathcal{H}$ is not known, or there are too many hypotheses $h \in \mathcal{H}$ to efficiently evaluate all of them, as many of these algorithms require. Therefore in practice these algorithms often use approximated using a committee of trained models as in Melville and Mooney (2004). QBC is also applied in regression, as in Burbidge et al. (2007), however the theoretical work and justification is not as well developed.

There are also many works that extend the ideas from traditional design of experiment (Santner et al., 2013) to the adaptive world. The goal is to reduce the variance of the parameter estimates when our model is parametric, and is often based on using the Fisher Information matrix. While this had been well studied in the fixed design setting, one of the first works extending these ideas to a non-fixed setting were MacKay (1992) and Cohn et al. (1996). It has been used other works such as Schein and Ungar (2007) where it was used for logistic regression, or Settles and Craven (2008) where it was used for sequence labelling. One benefit is that these methods can be studied theoretically, and recent work such as Chaudhuri et al. (2015) and Sourati et al. (2017) provide convergence rates and asymptotic analysis. Unfortunately these theoretically sound methods are often computationally expensive, and are restricted to parametric models where we have a Fisher Information matrix. However augmenting novel techniques for experimental design such as support points (Mak et al., 2018), core-sets (Sener and Savarese, 2017) and determinantal point processes (Bıyık et al., 2019) to be adaptive consistently produces competitive active learning algorithms.

1.2.2 Membership query synthesis setting

In membership query synthesis we are not restricted to a predefined set of data points. Instead we have some space $\mathcal{X}$ and at any point in that space we can sample a label $Y \in \mathcal{Y}$ at that point. Since we have no predefined set of interest, the objective here is almost always to accurately approximate the underlying function across the covariate space, usually measured by $||f - \hat{f}||^2 = \int_{\mathcal{X}} |f(x) - \hat{f}(x)|dx$. Membership query synthesis has been most widely studied in the service of Sequential Experimental Design for simulations (see Fedorov (1972) and Sacks et al. (1989)). The main simulation task is to build a *metamodel*, or a tractable data driven model (also called a surrogate model or emulator in the literature), of a complicated process which can only be understood through simulations, and which often have various tuning and operational variables. These models are then used to make predictions, guide further simulations, or perform sensitivity analysis. There is need for such models in various fields of physical science and engineering, including Tsunami modelling in Beck and Guillas (2016), industrial chemical engineering optimization in Jin et al. (2016) and borehole flow rate modelling in Liu et al. (2016). The objective of sequential experimental design is to guide the simulation process to sample parameters which would be most informative for modelling. See Liu et al. (2017) for a survey which details advances within this area, as well as discusses many of the key ideas and concepts. In contrast to the pool setting, the query synthesis setting has been almost exclusively used for regression problems, as in many classification tasks (see Baum and Lang (1992)) the algorithms were found to produce examples which could not be classified by humans.

One key difference between regression and classification is is the greater need for exploration in the regression context as opposed to classification. Within classification, data points far away from the classification boundary are considered by many models to be "known" (although with non-linear, non-realizable or noisy models this can be dangerous, see Dasgupta (2011) for a good one dimensional example) and so less exploration is needed away from the current boundary. However in regression it can take several data points to understand how the function behaves in any area, and much less local behaviour can be inferred from global properties of the function. This balance between exploration and exploitation is a common theme in interactive machine learning tasks. Many applied algorithms have explicitly focused on how to deal with that trade-off, and experimental (Singh et al., 2013) and theoretical (Hoang et al., 2014) studies have been done to consider how to optimally balance these conflicting goals.

Variants on many of the paradigms used in classification have been simply adapted for the regression setting with uncertainty based methods (MacKay, 1992; Cohn et al.,

1996), QBC (Viana et al., 2010; Eason and Cremaschi, 2014; Jiang et al., 2015; Golzari et al., 2015; Jin et al., 2016) and variance reduction methods (Krause et al., 2008; Beck and Guillas, 2016) all make an appearance. These methods are all fairly direct adaptations of methods used for classification, except that alterations to ensure there is sufficient exploration are needed. There are also local geometry based approaches which use the fact that our labels are not just class labels but are real numbers. The insight here is that, similar to how in classification the most valuable data points are often the ones very close to the classification boundary, for regression the equivalent ideas are that the most valuable data points are in the regions "most interesting", where this can have different definitions. In Crombecq et al. (2011) and Shahsavani and Grimvall (2009) they seek to find the area where the functions is most locally non-linear, using the local gradient and the quadratic term of a local polynomial regression respectively to measure non-linearity. In Pan et al. (2014) they take a slightly more global approach and sample around the best estimates of local maxima and minima so as to gain an understanding of the shape of the function. As with methods adapted more directly from classification, these methods all had explicit trade off between exploring the space where there is little information, and exploiting what is known about the geometry of the function to further learn about these areas of geometric interest.

1.3 Thesis structure and publications

The core material in this thesis is contained in three chapters. Each of these chapters (Chapter II, Chapter III and Chapter IV) has been adapted from publications and preprints:

- Chapter II is based off Goetz et al. (2018), which was published in the electronic proceedings of the Neural Information Processing Systems Conference. It studies active learning for regression, proposing an algorithm which outperforms random sampling with minimal assumptions.
- Chapter III is based off Goetz and Tewari (2019), a preprint to be submitted to the Electronic Journal of Statistics. It considers consistency of weighted averaging estimators after active learning has been used to sample the training data.
- Chapter IV is based off Goetz et al. (2019), which was an accepted paper at the International Workshop on Federated Learning for User Privacy and Data Confidentiality in Conjunction with the Neural Information Processing Systems Conference. It proposes Active Federated Learning, a framework for non-uniformly selecting (distributed) subsets of data during each federated learning training iteration.

CHAPTER II

Active Learning for Non-Parametric Regression Using Purely Random Trees

In binary classification active learning is known to produce faster rates than passive learning for a broad range of settings. However in regression restrictive structure and tailored methods were previously needed to obtain theoretically superior performance. In this chapter we propose an intuitive tree based active learning algorithm for non-parametric regression with provable improvement over random sampling. When implemented with Mondrian Trees our algorithm is tuning parameter free, consistent and minimax optimal for Lipschitz functions.

2.1 Introduction

In this chapter we study active learning for regression in the pool setting. In our setup we are given a pool of unlabelled data points and want to build the best model with a fixed number of samples, allowing selection of new points to use labels already obtained. Active learning is motivated by scenarios where the experimenter has control over the data labelling process and where unlabelled points are cheap but labels are expensive.

Our primary motivation comes from computational chemistry, where chemical properties of interest can be computed by solving approximations to the Schrödinger equation. One key property to chemists, the rate of chemical reaction, can be quantified via the activation energy, which controls the rate of reaction as a function of temperature Cramer (2013). While calculating the activation energy is expensive, there are a small number of readily available features of the reaction that influence the activation energy. This incentivizes building a metamodel for the activation energy to avoid excessive analysis of undesirable (high activation energy) reactions. Since we are restricted in the number of simulations used to build our metamodel, we want to use the most informative data points.

Because chemical reactions are discrete entities, we are restricted to a finite (but often large) pool of reactions, thus requiring pool setting active learning even though we are selecting simulations.

Active learning methods are usually built on top of existing prediction algorithms. Decision trees and forests are a popular class of such predictors due to their simplicity, expressiveness, state-of-the-art performance and tuning parameter free nature. In this chapter we focus our attention on purely random trees Breiman (2000), decision trees built independently of any data, due to their amenability to theoretical analysis. We use a recently proposed version called Mondrian Trees Lakshminarayanan et al. (2014), which have been shown to produce trees with many attractive properties such as consistency and minimax optimal rate of convergence for Lipschitz functions Mourtada et al. (2017).

As in some previous work Chaudhuri et al. (2017), our active learning algorithm will be developed in two stages. First we introduce a simple and intuitive *oracle* querying algorithm for purely random trees which is optimal among a natural class of sampling schemes which includes random sampling (Theorem 2.4.1). This algorithm is not active but uses statistics of the true joint distribution which are generally unknown. Second we propose an active learning scheme where we first sample passively to estimate the required statistics, and then use those estimates to approximate the oracle algorithm. We show this algorithm is consistent for the oracle algorithm (Theorem 2.5.1) and behaves well when our labels are normally distributed (Theorem 2.5.2). Finally we examine the empirical performance of our active learning algorithm to show that benefits, though sometimes modest, can be significant.

The structure of this chapter is as follows:

1. Introduce a family of sampling algorithms (Algorithm 1) and derive properties of those algorithms.

2. Use these properties to derive the optimal algorithm in this family (Theorem 2.4.1).

3. Propose an active learning algorithm (Algorithm 2) to approximate this optimal sampling algorithm.

4. Analyze the differences between the optimal algorithm and it's active approximation (Sections 2.5.1, 2.5.2, and 2.5.3).

5. Experimentally validate the results of our active learning algorithm (Section 2.6).

6. Provide details of all proofs (Section 2.8).

7. Give additional information with practical recommendations, additional experiments and possible extensions to forests (Section 2.9).

2.2 Setting and background

We begin by describing the pool based active learning setting, as well as introducing purely random and Mondrian trees. We have a pool of m data points $\{X_i\}_1^m$, with $X_i \in [0,1]^d$ (rescaling our X as needed) and $X_i \sim p_X$, which are always available to the algorithm. For each X_i we have a corresponding label $Y_i \in \mathbb{R}$ with the relationship $Y_i = f(X_i) + \sigma(X_i)\epsilon_i$ with $\epsilon_i \sim p_\epsilon$ iid, $\epsilon_i \perp X_j \forall j, E(\epsilon_i) = 0, \text{Var}(\epsilon_i) = 1, \sigma(X_i) : [0,1]^d \to \mathbf{R}_+$, meaning our noise is the product of a function of X with an independent random variable. We assume the $(X_i, Y_i) = D_i$ have been drawn iid from a joint distribution $p_{X,Y}$. We will assume that $f(x)$ and $\sigma(x)$ are bounded.

Initially none of these Y_i are known to the algorithm. Instead we have the ability to gain access to any of the Y_i, and the task is to select $n \ll m$ labels with the goal of building a model with the lowest quadratic risk $E\left[(\hat{f}(X) - f(X))^2\right]$, where the expectation is taken over our test point X, the random process which builds our tree and the labelled data we select. Throughout we will assume that our pool is arbitrarily large; in particular we will assume that the marginal density p_X is known, and that there is enough unlabelled data to implement any sampling scheme for selecting n points. We use *active* sampling (or learning) to describe any sampling scheme which samples in multiple batches and uses both $X_i's$ as well as known $Y_i's$ from previous batches when picking points for the next batch. We use passive sampling to denote any sampling scheme which only uses the X_i to pick points, and we use *random* sampling to denote picking the points uniformly at random from our pool (which is the same as sampling from $p_{X,Y}$).

Our active learning method is for purely random trees Breiman (2000), which are decision trees (or partitions of the space) built using a random process that is independent of the data. We will interchangeably discuss the partition of the space generated by the tree and the leaves of the tree. Let $I_k \in \mathcal{I}$ enumerate the leaves of a tree (partitions of the space), where $k \in \{1...K\}$. We will abuse notation slightly and use the set of partitions $\mathcal{I}$ to denote our tree. These partitions can be used to build regressograms, which make predictions using the average of labelled points within the partition of the test point. With the partitions fixed, the best (in L_2) approximation to f which is piece-wise constant on each partition predicts the conditional mean on that partition Györfi et al. (2006). We will denote true values and estimates of this approximation using "tilde" and "hat" notation as shown below.

True best approximation	Estimate of best approximation
$\tilde{f}_{\mathcal{I}}(x) = \sum_{k=1}^{K} \mathbf{1}(x \in I_k)\tilde{\beta}_k$	$\hat{f}_{\mathcal{I}}(x) = \sum_{k=1}^{K} \mathbf{1}(x \in I_k)\hat{\beta}_k$
$\tilde{\beta}_k = E_{p_{X,Y}}[Y \mid X \in I_k]$	$\hat{\beta}_k = \frac{1}{\sum \mathbf{1}(X_i \in I_k)} \sum_{X_i \in I_k} Y_i$

Our experiments and some results will use particular purely random trees built using the Mondrian Process Lakshminarayanan et al. (2014). The Mondrian Process is a stochastic process for partitioning a hypercube in $\mathbb{R}^d$, a single realization of this process gives a Mondrian Tree. The Mondrian Process iteratively splits existing partitions, and the number of partitions is controlled by a parameter λ which, since the Mondrian Process is a generalization of a Poisson Process, is referred to as the *lifetime* parameter. As this parameter increases the number of partitions increases, and the rate at which the number of partitions increase depends on the dimension and size of the hypercube. We will use Mondrian Trees on a fixed domain $[0, 1]^d$ with varying lifetime as in Mourtada et al. (2017), which describes how these random partitions are built.

2.3 Related work on active learning

The majority of theoretical work in active learning has taken place in binary classification, and there are many approaches which have been studied (see, e.g. Golovin and Krause (2011), Dasgupta et al. (2008), Sourati et al. (2017), Hoang et al. (2014), Balcan et al. (2009), Awasthi et al. (2014)). These algorithms are studied under fairly nonrestrictive assumptions (except occasionally requiring a linear classification boundary). It has been shown that for a variety of realistic noise conditions active learning provides a better minimax learning rate than passive learning (Hanneke and Yang (2015)).

In contrast the theory for active learning in regression is less well developed. A negative result Willett et al. (2006) showed that for a Lipschitz regression function and constant noise variance, the minimax learning rate for active learning was the same as that for passive (up to a constant). Additional assumptions are required to obtain better rates. Such structure includes assumptions of piece-wise constantness of regression function Willett et al. (2006), approximation of a non-linear model by a linear one Sabato and Munos (2014), locally varying smoothness Bull et al. (2013), well-specified parametric model Chaudhuri et al. (2015) or heteroskedasticity Efromovich (2008), Chaudhuri et al. (2017). Even in these cases the results are either discouraging or restrictive. In Arias-Castro et al. (2013) they show that if you are doing compressed sensing with sparsity, adaptively choos-

ing your samples can lead to an order $\log(\frac{n}{k})$ improvement in the signal estimation, where k is the sparsity of the signal (although as the authors point out, this is still only an $O(\log(n))$ improvement). And Efromovich (2008) shows that adaptive sampling can remove the price paid if estimating a function with heteroskedastic noise (assuming the function relating the heterskedasticity to X is suitably regular). However this analysis was restricted to one dimensional covariate space, where in classification you can get an exponential improvement over passive sampling with a generalization of binary search (see Nowak (2008)).

While many of these regression methods are able to provide provably better learning rates in terms of n, d, they are often tailored for their specific assumptions and may perform poorly if the assumptions do not hold. As a recent summary Liu et al. (2017) of numerous flexible but guarantee free methods shows, there is great demand for active learning methods without such stringent conditions. Our active learning algorithm will make very mild assumptions, but the improvement will not be in rates in n, d (since it is known this is not always possible). Rather we will adopt the approach (Golovin and Krause, 2011; Chaudhuri et al., 2017) of comparing the sampling generated by our algorithm to an optimal sampling scheme, as well as to random sampling.

2.4 Oracle label querying algorithm

We first describe a simple family of querying algorithms for a fixed purely random tree $\mathcal{I}$ which are not active. In the first two subsections below, we will be implicitly conditioning on the tree $\mathcal{I}$, but will suppress this in the notation.

2.4.1 Generic algorithm

In our generic algorithm family, the tree is built without using any data. So we build the tree first and query based on the tree's structure. We call it an "oracle" algorithm since it requires a priori

The algorithm is described as picking n_k deterministically for simplification of notation in proofs. However it is clear that if the n_k are random then it is easy (in principle) to discuss the probabilistic properties of the algorithm, and the details of the risk under random versions of Algorithm 1 are discussed in the proof for Corollary 2.4.2. The pool marginal distribution p_X and the proportion in each leaf q_k from the querying algorithm above induce a marginal distribution p'_X, as well as a joint distribution $p'_{X,Y} = p_{Y|X}p'_X$. The scheme is very general, and it is worth noting that random sampling is a (randomized) version of Algorithm 1. But this is enough structure to produce a somewhat obvious but very important property of our sampling distribution restricted to each leaf.

Algorithm 1: Generic "oracle" querying algorithm

Input: Leaves of our tree $\mathcal{I}$, pool of data points $\{X_i\}_{i=1}^m$, label budget n and joint distribution $p_{X,Y}$

Output: The set of points to label

foreach $I_k \in \mathcal{I}$ **do**

  Calculate q_k the proportion of points to select from leaf I_k, using $\mathcal{I}, \{X_i\}_{i=1}^m, n, p_{X,Y}$. ;

  Select $n_k = n \cdot q_k$ points uniformly at random from the pool of unlabelled points in that leaf. ;

end

Proposition 2.4.1. *Fix a tree structure $\mathcal{I}$, pool marginal density p_X and version of Algorithm 1, giving us an induced marginal density p'_X. Let $p'_X(X|I_k) = p'_X(X|X \in I_k)$ denote the induced marginal density conditioned on $X \in I_k$. Then as long as $q_k \neq 0$, $p'_X(X|I_k) = p_X(X|I_k)$ for any version of Algorithm 1.*

One important property this gives us is that $E_{p'_{X,Y}}[\hat{\beta}_k] = \tilde{\beta}_k$ (as long as I_k has at least 1 labelled point to estimate $\hat{\beta}_k$), meaning our sampling scheme produces unbiased estimates of the optimal regressogram for this tree. It also allows for a bias-variance decomposition of the risk of the tree. This decomposition was already known Genuer (2012) under the assumption of independence between tree structure and the data. We relax this assumption slightly as the distribution of the data depends on the structure of the tree, but still permits this decomposition.

Corollary 2.4.1. *For a fixed tree structure $\mathcal{I}$, under any sampling distribution generated by Algorithm 1 we have the following bias-variance decomposition of our risk:*

$$E\Big[(\hat{f}_{\mathcal{I}}(X) - f(X))^2\Big] = E\Big[(\tilde{f}_{\mathcal{I}}(X) - f(X))^2\Big] + E\Big[(\hat{f}_{\mathcal{I}}(X) - \tilde{f}_{\mathcal{I}}(X))^2\Big].$$

We will refer to these as the *risk bias term* and *risk variance term.* The risk bias term depends only on the structure of the tree, which does not depend our sampling scheme. We thus focus on the risk variance term. Again using Proposition 2.4.1 we show this term for a single leaf takes a simple form.

Lemma 2.4.1. *For a fixed tree structure $\mathcal{I}$, under any sampling distribution generated by Algorithm 1 we have that the variance error term on the leaf I_k is:*

$$E\Big[(\hat{f}_{\mathcal{I}}(X) - \tilde{f}_{\mathcal{I}}(X))^2 | X \in I_k\Big] = \frac{1}{n_k}\big(bias_k^2 + \sigma_{\epsilon,k}^2\big) = \frac{1}{n_k} Var(Y|X \in I_k),$$
$$bias_k^2 := E_{p_{X,Y}}\Big[(f(X) - \tilde{\beta}_k)^2 | X \in I_k\Big], \quad \sigma_{\epsilon,k}^2 := E_{p_{X,Y}}\Big[(\sigma(X)\epsilon)^2 | X \in I_k\Big].$$

2.4.2 Optimal algorithm

In the above lemma we have emphasized that the terms $bias_k^2$ and $\sigma_{\epsilon,k}^2$ have expectations taken with respect to the data generating distribution $p_{X,Y}$ and do not depend on the induced distribution $p'_{X,Y}$. Thus the only way our sampling distribution affects the variance term is through n_k. Averaging out over the contribution of each leaf we get that our overall variance error term is.

$$E\Big[(\hat{f}_{\mathcal{I}}(X) - \tilde{f}_{\mathcal{I}}(X))^2\Big] = \sum_k P(X \in I_k)\frac{1}{n_k}\big(bias_k^2 + \sigma_{\epsilon,k}^2\big). \tag{II.1}$$

Let $p_k = P(X \in I_k)$ under the pool marginal distribution and $\sigma_{Y,k}^2 = bias_k^2 + \sigma_{\epsilon,k}^2$. Now we are given a budget of n data points, and we want to minimize our variance error term subject to this budget. This gives us the following optimization problem which can be easily solved:

$$\begin{array}{ll} \underset{n_k}{\text{minimize}} & \sum_k \frac{1}{n_k} p_k \sigma_{Y,k}^2 \\ \text{subject to} & \sum_k n_k = n \end{array} \qquad \rightarrow \qquad n_k^* = n\frac{\sqrt{p_k\sigma_{Y,k}^2}}{\sum_{k'}\sqrt{p_{k'}\sigma_{Y,k'}^2}}$$

The proportions are very intuitive; cells with high bias and/or noise, or high (test) marginal density will get more samples. These results are summarized in the following theorem:

Theorem 2.4.1. *Let $Y_i = f(X_i) + \sigma(X_i)\epsilon_i$ and fix the partitions $\mathcal{I}$ of our tree. The risk minimizing oracle querying algorithm out of the family of algorithms described by Algorithm 1 is the one with the following n_k and error*

$$n_k^* = n\frac{\sqrt{p_k\sigma_{Y,k}^2}}{\sum_{k'}\sqrt{p_{k'}\sigma_{Y,k'}^2}}, \qquad E\Big[(\hat{f}_{\mathcal{I}}(X) - \tilde{f}_{\mathcal{I}}(X))^2\Big] = \frac{1}{n}\Big(\sum_k \sqrt{p_k\sigma_{Y,k}^2}\Big)^2.$$

Definition 2.4.1. The distribution induced by the sampling in Theorem 2.4.1 will be referred to as p_X^*.

Remark. This has a similar flavour to uncertainty sampling methods from classification in that regions with greater variation will get more samples. However whereas in classification sampling can focus locally near the decision boundary, in regression sampling must remain global.

Random sampling is a randomized version of Algorithm 1, so the risk under random sampling is the bias term plus a weighted average of the variance terms for different $(n_1, ..., n_K)$. The sampling scheme from Theorem 2.4.1 has the same bias term, but minimizes the variance term meaning our optimal sampling scheme is better than any randomized version of Algorithm 1 (as long as $m > n$), including random sampling.

Corollary 2.4.2. *For a fixed tree structure $\mathcal{I}$, the risk from any randomized version of Algorithm 1 is greater than the risk from sampling according to p_X^* unless $P(n_1^*, ..., n_K^*) = 1$. In particular sampling according to p_X^* is strictly better than random sampling.*

We can also calculate the excess error if we use the incorrect values of $\sigma^2_{Y,k}$. Let $\tilde{\sigma}^2_{Y,k} = a_k \sigma^2_{Y,k}$, so a_k is a multiplicative error (we will see that our errors will be multiplicative). Given fixed leaf errors $a_1, ..., a_K$ we can calculate the additional risk generated by using $\tilde{\sigma}^2_{Y,k}$ in our optimal algorithm instead of the true $\sigma^2_{Y,k}$

Lemma 2.4.2. *For a fixed tree structure $\mathcal{I}$, if $n_k = n\frac{\sqrt{p_k \tilde{\sigma}^2_{Y,k}}}{\sum_{k'} \sqrt{p_{k'} \tilde{\sigma}^2_{Y,k'}}}$ and the variance error term for each leaf is as in Lemma 2.4.1, then our risk variance is:*

$$E\Big[(\hat{f}_{\mathcal{I}}(X) - \tilde{f}_{\mathcal{I}}(X))^2\Big] = \frac{1}{n}(\sum_k \sqrt{p_k \sigma^2_{Y,k}})^2 + \frac{1}{n}\sum_{k<l}(\frac{\sqrt{a_k}}{\sqrt{a_l}} + \frac{\sqrt{a_l}}{\sqrt{a_k}} - 2)\sqrt{p_k p_l \sigma^2_{Y,k}\sigma^2_{Y,l}}$$
$$:= \text{OPT} + \text{EXCESS}.$$

This also lets us get a sense for the suboptimality of random sampling. If we let $a_k = \frac{p_k}{\sigma^2_{Y,k}}$ then we get $n_k = np_k$ which is the expected number of samples per leaf under random sampling, and so for large n the calculated EXCESS term will be close to the excess risk under random sampling. This gives us the following excess error, which can be small (or even zero) as expected since random sampling can be near-optimal. But if there is varying Y variance across the space this can be large:

Corollary 2.4.3. *For a fixed tree structure $\mathcal{I}$ let $a_k = \frac{p_k}{\sigma^2_{Y,k}}$. Then our excess error is:*

$$\text{EXCESS} = \frac{1}{n}\sum_{k<l}(\sqrt{p_k \sigma^2_{Y,l}} - \sqrt{p_l \sigma^2_{Y,k}})^2 \leq \frac{K}{n}\max_k \sigma^2_{Y,k}.$$

2.4.3 Additional results using Mondrian Trees

The above results hold for any purely random tree. We will now not assume that $\mathcal{I}$ is fixed, but is randomly built using the Mondrian Process and will take expectation over the tree building process as well. Mondrian Trees trained using random sampling are minimax optimal for Lipschitz regression functions when the sequence of lifetime parameters satisfy $\lambda_n \asymp n^{1/(d+2)}$ and $\text{Var}(Y) < \infty$ Mourtada et al. (2017). Additionally Mondrian Trees with random sampling are weakly universally consistent under the same lifetime sequence and variance assumption. Since the optimal oracle algorithm has smaller risk we immediately get minimax optimal rates in terms of n, d under the same assumptions lifetime sequence by Proposition 4 in Mourtada et al. (2017) and Theorem 2.4.1, and weak consistency under Theorem 1 in Mourtada et al. (2018).

Corollary 2.4.4. *Let our purely random trees be Mondrian Trees with lifetime parameters $\lambda_n \asymp n^{1/(d+2)}$, and let $Y = f(X) + \sigma(X)\epsilon$, Var$(Y) < \infty$. If our training data is sampled according to p_X^* then the resulting regressogram has (as $n, m \to \infty$) minimax optimal rates, in terms of n, d, over Lipschitz functions with $E\left[(\hat{f}(X) - f(X))^2\right] = \mathcal{O}(n^{\frac{-2}{2+d}})$ and is weakly consistent.*

2.5 Active learning algorithm

The oracle querying algorithm has many appealing qualities. However it requires knowledge of the $\sigma^2_{Y,k}$ which are never known in practice. In this section we propose a two stage active "oracle estimating" algorithm to remedy this deficiency. In our first stage we sample $n_{(1)}$ points according to Algorithm 1 and use those samples to calculate estimates $\hat{\sigma}^2_{Y,k}$ of $\sigma^2_{Y,k}$, which in turn produce estimates $\hat{n}_k$ of n_k^*. In the second stage we sample $n_{(2)} = n - n_{(1)}$ points such that the total number of samples in each leaf are $\hat{n}_k$. We analyze the consequences of using these estimates, and show that in the case when Y are normal, our trees are Mondrian Trees, and our Stage 1 samples equally in each leaf, our active algorithm is eventually near optimal with high probability. We also show that in general this algorithm's estimates $\hat{n}_k$ are consistent for n_k^*. Below we give the active algorithm. By using this algorithm we have introduced two complications: One is the estimates will have errors from using estimates $\hat{\sigma}^2_{Y,k}$. The other comes from reusing the data from Stage 1 in our estimates of $\hat{\beta}_k$. Since active learning is used exactly when data is difficult to label, to make an algorithm which is practically appealing it is important to make the most out of any labelled data. However this introduces dependency between $\hat{\beta}_k$ and $\hat{n}_k$. These issues will each be addressed separately.

Algorithm 2: Active "oracle estimating" algorithm

Input: Leaves of our tree $\mathcal{I}$, pool of data points $\{X_i\}_{i=1}^m$, and label budgets $n_{(1)}, n_{(2)}, n = n_{(1)} + n_{(2)}$.

Output: The set of labelled points.

Stage 1 ;

Query $n_{(1)}$ data points using a version of Algorithm 1. ;

Use those samples (X_i, Y_i) to estimate $\hat{\sigma}^2_{Y,k} = \frac{1}{n_{(1),k}-1} \sum_{X_i \in I_k} (\hat{\beta}_{(1),k} - Y_i)^2$ for each leaf. ;

Stage 2 ;

foreach $I_k \in \mathcal{I}$ **do**

Calculate $\hat{n}_k = n \frac{\sqrt{p_k \hat{\sigma}^2_{Y,k}}}{\sum_{k'} \sqrt{p_{k'} \hat{\sigma}^2_{Y,k'}}}$ the number of points in the leaf to sample. ;

Select uniformly at random $n_{(2),k}$ points to query from the leaf so the number of points is $\hat{n}_k$. ;

end

2.5.1 Using estimates of n_k^*

First we analyze (as n increases) the effect of using the estimates $\hat{\sigma}^2_{Y,k}$. Let us fix a sequence of trees $\mathcal{I}_{(n)}, |\mathcal{I}_{(n)}| = K_n$. Typically our trees will contain more partitions as we get more data. For a given tree we can estimate the required leaf variances unbiasedly using the standard unbiased sample variance on each leaf $\hat{\sigma}^2_{Y,k}$. Therefore as long as our leaf kurtosis $\kappa_{Y,k} = \frac{\sigma^4_{Y,k}}{(\sigma^2_{Y,k})^2}$ (and thus the variance of our sample variance) are all finite, and asymptotically our sample variances on each leaf are consistent for the true variances on each leaf, our estimates $\hat{n}_k \to n_k^*$. We require strong consistency of our variance estimates as a function of both our partitioning method and Stage 1 sampling method, which gives us $\hat{n}_k \to n_k^*$ almost surely. If our trees are grown according to a random process then this strong consistency may be depend on attributes of the tree which my only be true in probability, and in this case we get $\hat{n}_k \to n_k^*$ in probability. Both cases are covered in the below theorem, where generally the b_n denote statistics of the tree and B is either 0 or ∞.

Theorem 2.5.1. *Assume $\kappa_{Y,k} < \infty \; \forall \; k, n$, and our sequence of trees $\mathcal{I}_{(n)}$ and Stage 1 sampling algorithm is strongly consistent for estimating the conditional variance $E[(Y - f(X))^2 | X = x]$ as some statistic $b_n \to B$. Then if $b_n \to B$ almost surely our estimates $\hat{n}_k \to n_k^*$ almost surely and if $b_n \to B$ in probability our estimates $\hat{n}_k \to n_k^*$ in probability.*

Remark. Note that the condition $\kappa_{Y,k} < \infty \; \forall \; k, n$ is met if $f, \sigma(X)$ are bounded and $\kappa_\epsilon < \infty$.

Now let our sequence of trees be randomly built Mondrian Trees. If we again use

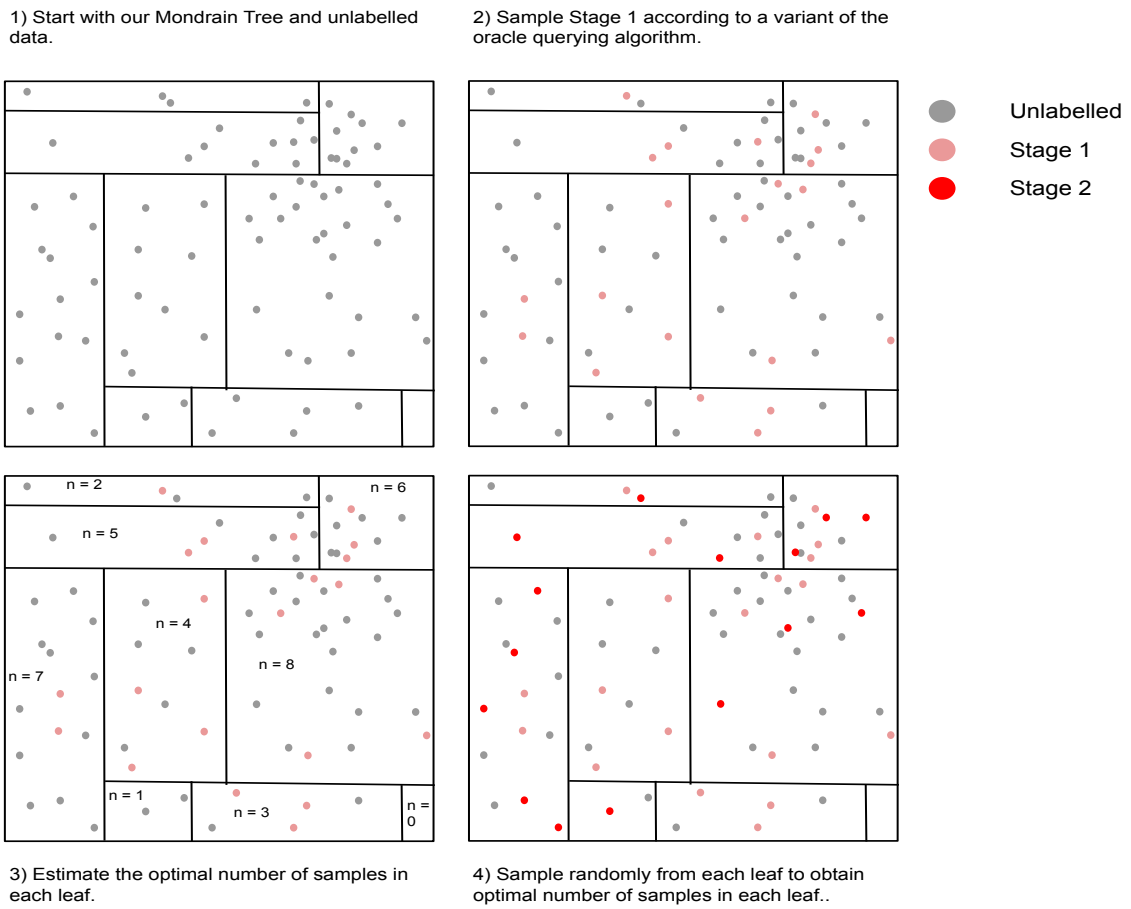

Figure 2.1: Visualization of Algorithm 2.

$\lambda_n \asymp n^{1/(d+2)}$, as long as $n_{(1)}$ increases linearly with n, these conditions are met when our first round of sampling entails sampling equally in each leaf.

Corollary 2.5.1. *Let our purely random trees be Mondrian Trees with lifetime parameter sequence $\lambda_n \asymp n^{1/(d+2)}$ and let $n_{(1)} = cn,\ c \in (0,1)$ a constant. Additionally let Stage 1 query by $n_{(1),k} = \frac{n_{(1)}}{K_n} \forall\, k$. If $\kappa_{Y,k} < \infty \forall\, k, n$ and p_X is bounded away from 0 and ∞ on it's support, so when $p_X > 0$ there exists c, C s.t. $c \leq p_X \leq C$, then our estimates $\hat{n}_k \to n_k^*$ in probability.*

Even with consistency our finite sample estimates will give us some error in $\hat{n}_k$. The variance of our sample variance is $\text{Var}(\hat{\sigma}^2_{Y,k}) = \frac{1}{n_k}(\sigma^4_{Y,k} - (\sigma^2_{Y,k})^2) + \mathcal{O}(\frac{1}{n_k^2}) \approx \frac{1}{n_k}(\kappa_{Y,k} - 1)(\sigma^2_{Y,k})^2$, so our errors will scale multiplicatively with $\sigma^2_{Y,k}$ when our kurtosis $\kappa_{Y,k}$ are bounded. This allows us to use Lemma 2.4.2 to bound our excess error given bounds on the (multiplicative) error $a_k = \hat{\sigma}^2_{Y,k}/\sigma^2_{Y,k}$.

2.5.2 Reusing data

Since we are using the data in Stage 1 both to estimate $\hat{n}_k$ as well as in our estimator $\hat{\beta}_k$, we have introduced dependence between the estimated optimal leaf sample size $\hat{n}_k$ and leaf mean estimate contribution from Stage 1. To understand the effects of this dependence we will break up our estimates of the leaf mean as $\hat{\beta}_k = \frac{n_{(1),k}\hat{\beta}_{(1),k}+n_{(2),k}\hat{\beta}_{(2),k}}{n_{(1),k}+n_{(2),k}}$, where $n_{(i),k}, \hat{\beta}_{(i),k}$ are the number and mean estimate during sampling round $i \in \{1, 2\}$. By writing our final mean estimate in terms of our stage-wise mean estimates we can find an expression for this dependence.

Lemma 2.5.1. *For a fixed tree structure $\mathcal{I}$, under Algorithm 2 the risk variance term becomes:*

$$E[(\hat{\beta}_k - \tilde{\beta}_k)^2] =$$
$$E_{n_{(2),k}}\Big[\frac{n_{(1),k}^2}{(n_{(1),k}+n_{(2),k})^2} E_{D_{1:n_{(1)}}}\big[(\hat{\beta}_{(1),k} - \tilde{\beta}_k)^2 | n_{(2),k}\big] + \frac{n_{(2),k}\sigma_{Y,k}^2}{(n_{(1),k}+n_{(2),k})^2}\Big].$$

The term $E_{D_{1:n_{(1)}}}\big[(\hat{\beta}_{(1),k} - \tilde{\beta}_k)^2 | n_{(2),k}\big]$ quantifies the dependency introduced by reusing the samples from $n_{(1)}$. The dependency is between the variance of part of our mean estimators $(\hat{\beta}_{(1),1}, ..., \hat{\beta}_{(1),k})$ and $(n_{(2),1}, ..., n_{(2),K}) = g(\hat{\sigma}_{Y,1}^2, ..., \hat{\sigma}_{Y,K}^2)$. When $\hat{\beta}_{(1),k} \perp n_{(2),k}$ we get back our risk variance term from Lemma 2.4.1. However when there is dependence we no longer have that the n_k^* from Theorem 2.4.1 are optimal over algorithms with an active stage as in Algorithm 2, since the optimal n_k will depend on the sampling during Stage 1. This dependency can be complex and is generally unknown, though as long as the effect is not too large the n_k^* will still provide a very good solution, and the n_k^* are still better than random sampling. It is worth noting that our active algorithm can take advantage of this dependency in some cases to outperform Algorithm 1, and we informally discuss this in Section 2.9.1.

2.5.3 The Normal case

The complications above depend on the distribution of $a_k = \frac{\hat{\sigma}_{Y,k}^2}{\sigma_{Y,k}^2}$ and the function g, which in general are extremely complicated and hard to analyze for arbitrary f, p_ϵ, p_X. However in the case where Y are normally distributed these become tractable.

Theorem 2.5.2. *Let $Y \sim N(\mu(X), \sigma^2(X))$ and X queried according to Algorithm 2 for a fixed tree $\mathcal{I}$. Then the risk variance term for a leaf is as in Lemma 2.4.1 and we have that*

with probability at least $1 - \sum_{k=1}^{K} e^{-\frac{(n_{(1),k}-1)\alpha^2}{8}}$ *the excess error is bounded by:*

$$\text{EXCESS} \leq \frac{1}{n} \sum_{k<l} \Big[\big(\frac{1+\alpha}{1-\alpha}\big)^{1/4} - \big(\frac{1-\alpha}{1+\alpha}\big)^{1/4} \Big]^2 \sqrt{p_k p_l \sigma^2_{Y,k} \sigma^2_{Y,l}} \, .$$

Additionally if our trees are a sequence of Mondrian Trees with lifetime parameter sequence $\lambda_n \asymp n^{1/(d+2)}$ *and our Stage 1 sampling procedure is to sample equally in each leaf with* $n_{(1)} = cn$, $c \in (0,1)$ *a constant, then the above bound occurs with probability at least* $1 - \delta_1 - \delta_2$ *where*

$$\delta_1 = \frac{(1+n^{1/(d+2)})^d}{n^{(d+1)/(d+2)}} \quad \delta_2 = n^{(d+1)/(d+2)} \exp\big(\frac{-\alpha^2}{8}((cn)^{1/(d+2)}) - 1\big).$$

First, note that a larger n allows us to choose a smaller α and the bound on excess error goes to 0 as $\alpha \to 0$. Second, even for the normal case, d large requires a very large n before we get any control on the error probability δ_2. This is consistent with the empirical observation that Mondrian Trees struggle with large d.

Finally we also note that there are many reasons why in practice it is impossible to use the exact n_k^*. These include the fact that usually n_k^* will be fractional, a leaf may not have n_k^* points in it, or when using the active algorithm $n_{(1),k} > \hat{n}_k$. These issues will be less significant as $n \to \infty$ and we discuss how each is dealt with in Section 2.9.

2.6 Simulations and experiments

We now examine the benefits of active learning on both simulated and real world data. We simulate 2 data sets, one with differing noise variance (our $\sigma^2_{\epsilon,k}$ term), the other with differing function complexity (our $bias^2_k$ term), in different regions of $[0,1]^d$. We also examine performance on the Wine quality data set from UCI and a data set of activation energies of Claisen rearrangement reactions (Cl). We compare the performance of selecting points to label using random sampling, our active algorithm, and a naive uncertainty sampling version of our active algorithm, where each leaf n_k is proportional its variance. In all experiments $n_{(1)} = \frac{n}{2}$ and Mondrian Trees are grown using $\lambda_n = n^{\frac{2}{2+d}} - 1$, which is theoretically motivated, but corrected so when $n = 1, \lambda_n = 0$. We use both Mondrian and Breiman Trees Breiman (2017) as our final regressor. Details of the data sets are in the appendix, which also contains forest versions of these experiments. Additionally all code and experiments (as well as other experiments) are available at `https://github.com/jackrgoetz/Mondrian_Tree_AL`.

When using Mondrian Trees as the final regressor, the active learning method always provides some improvement, and in the simulations this improvement persists when using Breiman Trees. Additionally the uncertainty sampling method sometimes produces worse sampling than random sampling, which is common for direct translations of classification active learning methods. In the real data our benefits are less pronounced, with active learning even being slightly harmful when used with Breiman Trees (although with forests the active learning is beneficial). We believe this performance drop may be due to the inability of the Mondrian Tree to adapt to differing variable importance. It is also possible that our assumptions that Y has changing variance does not hold, and even here the active algorithm is not harmful, where as the naive uncertainty sampling algorithm can be detrimental.

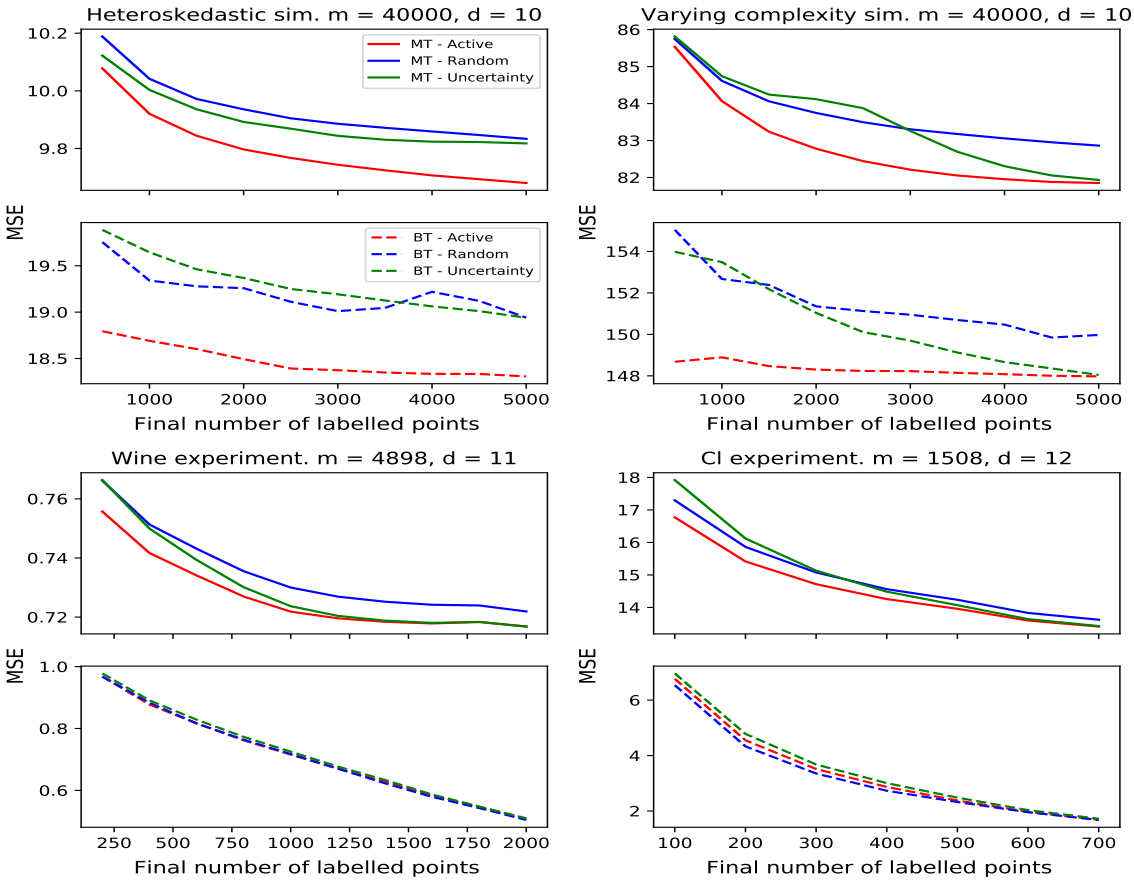

Figure 2.2: Active learning experiments

2.7 Conclusion and further directions

In this chapter we provide a theoretically justified active learning method for non-parametric regression which can take advantage of beneficial structure when present without being

detrimental when such structure is absent. When used with Mondrian Trees the method requires no tuning parameters (which are difficult to tune while actively sampling (Attenberg and Provost, 2011)), is asymptotically minimax optimal for Lipschitz regression functions, and is consistent. Although the improvement for active learning in regression is often restricted to constant factor improvements, these constant improvements are important in real world applications.

Despite technical theoretical arguments needed for the theory, the method itself is simple, leading to many interesting avenues for further exploration. One direction would be extending theory to ensembles of trees, or developing tools to deal with high dimensions. Another possibility is to exploit the online nature of Mondrian Trees to develop a parallel theory for streaming based active learning. Finally it may be possible to extend the ideas here to non tree based active learning for regression.

2.8 Appendix A: Proofs

2.8.1 Proofs for the oracle algorithm

Proof of Proposition 2.4.1. This results is nothing more than the fact that a random subsample of size $n < m$ from an initial sample of sizer m has the same distribution as a sample of size n from that original distribution. The only issue here is if $q_k = 0$, in which case $p'_X(x) = 0 \, \forall \, x \in I_k$, where as $p_X(x)$ may be non-zero on a set of positive measure.

□

Proof of Corollary 2.4.1. We start by confirming that $E_{p'_{X,Y}}[\hat{\beta}_k] = \tilde{\beta}_k$. Let us fix $\mathcal{I}, k$ with n labelled points and let $n_k = \sum_{i=1}^{n} \mathbf{1}(X_i \in I_k)$. By assumption $n_k > 0$ otherwise $\hat{\beta}_k = \frac{1}{\sum \mathbf{1}(X_i \in I_k)} \sum_{X_i \in I_k} Y_i$ is undefined. Since Algorithm 1 is not active we have that $Y|X \in I_k \perp n_k$.

$$
\begin{aligned}
E_{p'_{X,Y}}[\hat{\beta}_k] &= E_{n_k} E_{p'_{X,Y}}\big[\frac{1}{\sum \mathbf{1}(X_i \in I_k)} \sum_{i=1}^{n} Y_i \mathbf{1}(X_i \in I_k) | n_k\big] \\
&= E_{n_k} \frac{1}{n_k} \sum_{i=1}^{n} E_{p'_{X,Y}}\big[Y_i \mathbf{1}(X_i \in I_k) | n_k\big] \\
&= E_{n_k} \frac{1}{n_k} \sum_{i=1}^{n} P(X_i \in I_k | n_k) E_{p'_{X,Y}}\big[Y_i | n_k, X_i \in I_k\big] \\
&= E_{n_k} \frac{1}{n_k} E_{p_{X,Y}}\big[Y | X \in I_k\big] \sum_{i=1}^{n} P(X_i \in I_k | n_k) \\
&= E_{n_k} E_{p_{X,Y}}\big[Y | X \in I_k\big] = E_{p_{X,Y}}\big[Y | X \in I_k\big].
\end{aligned}
$$

Now we use this to derive the decomposition in the standard way.

$$
\begin{aligned}
E\Big[(\hat{f}_{\mathcal{I}}(X) - f(x))^2\Big] &= E\Big[(\hat{f}_{\mathcal{I}}(X) - \tilde{f}_{\mathcal{I}}(X))^2\Big] + E\Big[(\tilde{f}_{\mathcal{I}}(X) - f(X))^2\Big] \\
&\quad + 2E\Big[(\hat{f}_{\mathcal{I}}(X) - \tilde{f}_{\mathcal{I}}(X))(\tilde{f}_{\mathcal{I}}(X) - f(X))\Big].
\end{aligned}
$$

$$
\begin{aligned}
&E\Big[(\hat{f}_{\mathcal{I}}(X) - \tilde{f}_{\mathcal{I}}(X))(\tilde{f}_{\mathcal{I}}(X) - f(X))\Big] = \\
&\quad E[\hat{f}_{\mathcal{I}}(X)]\tilde{f}_{\mathcal{I}}(X) - E[\hat{f}_{\mathcal{I}}(X)]f(X) - \tilde{f}_{\mathcal{I}}(X)^2 + \tilde{f}_{\mathcal{I}}(X)f(X) = 0.
\end{aligned}
$$

□

Proof of Lemma 2.4.1. We fix n_k. Given $X \in I_k$ we know that $\hat{f}_{\mathcal{I}}(X) = \hat{\beta}_k$ and $\tilde{f}_{\mathcal{I}}(X) = \tilde{\beta}_k$. Let us reorder the data $D_{1:n}$ so that the first n_k are in the leaf k for ease of notation. Then use Proposition 2.4.1, where the cross term disappears since $\epsilon_i \perp X_i$ under $p_{X,Y}$ by assumption.

$$
\begin{aligned}
&E_{p'_{X,Y}}\Big[(\hat{f}_{\mathcal{I}}(X) - \tilde{f}_{\mathcal{I}}(X))^2 | X \in I_k\Big] = \\
&\frac{1}{n_k^2}\Big(\sum_{i=1}^{n_k} E_{p'_{X,Y}}\Big[(f(X_i) - \tilde{\beta}_k)^2 | X_i \in I_k\Big] + \sum_{i=1}^{n_k} E_{p'_{X,Y}}\Big[(\sigma(X_i)\epsilon_i)^2 | X_i \in I_k\Big] \\
&+ 2\sum_{i=1}^{n_k} E_{p'_{X,Y}}\Big[(f(X_i) - \tilde{\beta}_k)\sigma(X_i)\epsilon_i | X_i \in I_k\Big]\Big) \\
&= \frac{1}{n_k^2}\Big(\sum_{i=1}^{n_k} E_{p_{X,Y}}\Big[(f(X_i) - \tilde{\beta}_k)^2 | X_i \in I_k\Big] + \sum_{i=1}^{n_k} E_{p_{X,Y}}\Big[(\sigma(X_i)\epsilon_i)^2 | X_i \in I_k\Big] \\
&= \frac{1}{n_k}\big(bias_k^2 + \sigma_{\epsilon,k}^2\big).
\end{aligned}
$$

□

Proof of Corollary 2.4.2. The proof involves looking at the expected risk under a random version of Algorithm 1. Formally allow Algorithm 1 to generate the q_i in a randomized fashion (with the randomness independent from all other sources of randomness), potentially using the other inputs to Algorithm 1 $(\mathcal{I}, \{X_i\}_{i=1}^m, n, p_{X,Y})$ as parameters. Thus $(q_1, ..., q_K)$ are drawn from a distribution, which in turn for all $(n_1, ..., n_K) \in \mathbb{N}^K \; s.t. \; \sum n_k = n$ generates $P(n_1, ..., n_K)$ the probability of the algorithm sampling $(n_1, ..., n_K)$ points from each of the tree leaves. Let $Risk(n_1, ..., n_K)$ denote the risk when our by leaf samples sizes are $n_1, ..., n_k$, with $RiskBias$ and $RiskVar(n_1, ..., n_K)$ being the bias and variance terms of the decomposition. The $RiskBias$ does not depend on $n_1, ..., n_K$ since the risk bias term does not depend on how we sample. Then the risk of the randomized version of Algorithm 1 is

$$
\begin{aligned}
Risk &= \sum_{(n_1,...,n_K)} P(n_1, ..., n_K) Risk(n_1, ..., n_K) \\
&= RiskBias + \sum_{(n_1,...,n_K)} P(n_1, ..., n_K) RiskVar(n_1, ..., n_K).
\end{aligned}
$$

If $n_1^*, ..., n_K^*$ is our optimal solution then by Theorem 2.4.1 $RiskVar(n_1^*, ..., n_K^*) \leq RiskVar(n_1, ..., n_K) \; \forall \; (n_1, ..., n_K) \in \mathbb{N}^K \; s.t. \; \sum n_k = n$. For random sampling, unless $P(n_1^*, ..., n_K^*) = 1$ the Risk will clearly be greater than (or equal to) that of the optimal

since the probability weighted average is greater than (or equal to) the min term of the sum.

□

Proof of Lemma 2.4.2. This is all algebra. By Equation II.1

$$\begin{aligned}
E\Big[(\hat{f}_{\mathcal{I}}(X) - \tilde{f}_{\mathcal{I}}(X))^2\Big] &= \frac{1}{n}\sum_{k=1}^{K}\sqrt{a_k}\sqrt{p_k\sigma^2_{Y,k}} \times \sum_{l=1}^{K}\frac{1}{\sqrt{a_l}}\sqrt{p_l\sigma^2_{Y,l}} \\
&= \frac{1}{n}\Big(\sum_k p_k\sigma^2_{Y,k} + \sum_{k\neq l}\frac{\sqrt{a_k}}{\sqrt{a_l}}\sqrt{p_kp_l\sigma^2_{Y,k}\sigma^2_{Y,l}}\Big) \\
&= \frac{1}{n}\Big(\sum_k p_k\sigma^2_{Y,k} + \sum_{k<l}\Big(\frac{\sqrt{a_k}}{\sqrt{a_l}} + \frac{\sqrt{a_k}}{\sqrt{a_l}}\Big)\sqrt{p_kp_l\sigma^2_{Y,k}\sigma^2_{Y,l}}\Big) \\
&= \frac{1}{n}\Big(\sum_k \sqrt{p_k\sigma^2_{Y,k}}\Big)^2 + \frac{1}{n}\sum_{k<l}\Big(\frac{\sqrt{a_k}}{\sqrt{a_l}} + \frac{\sqrt{a_l}}{\sqrt{a_k}} - 2\Big)\sqrt{p_kp_l\sigma^2_{Y,k}\sigma^2_{Y,l}} \\
&= \text{OPT + ERROR}.
\end{aligned}$$

□

Proof of Corollary 2.4.3. Again, this is just algebra.

$$\begin{aligned}
&\frac{1}{n}\sum_{k<l}\Big(\frac{\sqrt{p_k\sigma^2_{Y,l}}}{\sqrt{p_l\sigma^2_{Y,k}}} + \frac{\sqrt{p_l\sigma^2_{Y,k}}}{\sqrt{p_k\sigma^2_{Y,l}}} - 2\Big)\sqrt{p_kp_l\sigma^2_{Y,k}\sigma^2_{Y,l}} = \frac{1}{n}\sum_{k<l}\Big(\sqrt{p_k\sigma^2_{Y,l}} - \sqrt{p_l\sigma^2_{Y,k}}\Big)^2 \\
&\leq \frac{1}{n}\sum_{k<l}(2p_k\sigma^2_{Y,l} + 2p_l\sigma^2_{Y,k}) \leq \frac{1}{n}\max_k\sigma^2_{Y,k}\sum_{k\neq l}(p_k + p_l) \leq \frac{K}{n}\max_k\sigma^2_{Y,k}.
\end{aligned}$$

□

2.8.2 Proofs for the active algorithm

Proof of Theorem 2.5.1. By the assumption that our sequence of trees $\mathcal{I}_{(n)}$ and Stage 1 sampling algorithm is strongly consistent for estimating the conditional variance $E[(Y - f(X))^2|X = x]$ as some statistic $b_n \to B$ we have that $\hat{\sigma}^2_{1,k} \to \sigma^2_k$ a.s. as $b_n \to B$. To see this let $\hat{\sigma}^2_{1,k}(x) = \hat{\sigma}^2_{1,k}$ for $x \in I_k$, $\sigma^2_k(x) = \sigma^2_k$ for $x \in I_k$ and let $\sigma^2(x) = E[(Y - f(X))^2|X = x]$. Then $|\hat{\sigma}^2_{1,k}(x) - \sigma^2_k(x)| \leq |\hat{\sigma}^2_{1,k}(x) - \sigma^2(x)| + |\sigma^2_k(x) - \sigma^2(x)| \to 0$, where the first term disappears due to the strong consistency, and the second term disappears due to the size of the partitions shrinking.

If $\hat{\sigma}^2_{1,k} \to \sigma^2_k$ a.s. then $\sum_{k=1}^{K_n} \sqrt{p_k \hat{\sigma}^2_{1,k}} \to \sum_{k=1}^{K_n} \sqrt{p_k \sigma^2_k}$ a.s. as $b_n \to B$. So if $b_n \to B$ a.s. then $\hat{n}_k \to n^*_k$ almost surely.

Now assume $b_n \to B$ in probability as $n \to \infty$ and want to show that these implies $\sum_{k=1}^{K_n} \sqrt{p_k \hat{\sigma}^2_{1,k}} \to \sum_{k=1}^{K_n} \sqrt{p_k \sigma^2_k}$ in probability $n \to \infty$. We will use Lemma 6.3.1.b from Resnick (2013) which states:

Lemma 1 (6.3.1.b in Resnick (2013))**.** *$X_n \to X$ in probability iff for each subsequence $\{X_{n_k}\}, n_k \to \infty$ there exists a further subsubsequence $\{X_{n_{k_t}}\}, n_{k_t} \to \infty$ which converges a.s. to X.*

(The n_k here are unrelated to the n_k in our trees).

Let $Y_n = |\sum_{k=1}^{K_n} \sqrt{p_k \hat{\sigma}^2_{1,k}} - \sum_{k=1}^{K_n} \sqrt{p_k \sigma^2_k}|$, so $Y_n \to 0$ a.s. if $b_n \to B$. Thus we have a subset of the overall probability space Ω which is

$$\Omega \supset \Omega^* = \{\omega \in \Omega : \lim b_n(\omega) \neq B \text{ or } Y_n(\omega) \to 0\}$$

where $P(\Omega^*) = 1$. Now take a subsequence $n_k \to \infty$ of n. By $b_n \to B$ in probability $\exists n_{k_t} \to \infty$ such that $b_{n_{k_t}} \to B$ a.s. as $n_{k_t} \to \infty$. This gives us a second subset of Ω

$$\Omega \supset \Omega' = \{\omega \in \Omega : b_{n_{k_t}}(\omega) \to B\}$$

where again $P(\Omega') = 1$. On the intersection of these we get

$$\Omega^* \cap \Omega' \subset \{\omega \in \Omega : Y_{n_{k_t}(\omega)}(\omega) \to 0\}$$

where $P(\Omega^* \cap \Omega') = 1$. n_k was an arbitrary subsequence of n and so by using Lemma 6.3.1.b in the reverse direction we get that $Y_n \to 0$ in probability.

□

Proof of Corollary 2.5.1. Here our $b_n = \frac{K_n}{n^{\frac{d+1}{d+2}}}$ and $B = 0$. Since $E[K_n] = (1 + n^{\frac{1}{d+2}})^d$ by Markov $\frac{K_n}{n^{\frac{d+1}{d+2}}} \to 0$ in probability.

Now we need to show that if we assume $\frac{K_n}{n^{\frac{d+1}{d+2}}} \to 0$ we get strong consistency of our conditional variance function estimation. By Theorem 23.3 in Györfi et al. (2006) we get that our tree is strongly consistent for estimating the mean function, since $\frac{K_n \log(n)}{n} \to 0$ so eventually every partition will have more than $\log(n)$ samples in the leaf, and the augmented estimator in Theorem 23.3 is the same as the usual estimator. (The augmented estimator in Theorem 23.3 is the usual decision tree estimator if there are more than $\log(n)$

data points in the partition and 0 otherwise). Finally we need the p_X bounded since Theorem 23.3 assumes that our test X density is the same as our training one, but since p_X is bounded the Radon Nikodym derivative is bounded and so we get strong consistency even with the different test density.

So our tree and Stage 1 sampling scheme are strongly consistent for estimating the mean function $f(x) = E[Y|X = x]$. Now assume we had access to a new set of random variables $Z_i = (Y_i = f(X_i))^2$. Because of the bounded kurtosis our tree would also be strongly consistent for estimating the mean function of the Z_i which we will call $f_Z(x) = E[(Y - f(X))^2|X = x]$. So if we had access to the Z_i we could use them to estimate our Y conditional variance function using $\hat{f}_Z(x) = \frac{\sum Z_i \mathbf{1}_{X_i \in I(x)}}{\sum \mathbf{1}_{X_i \in I(x)}}$.

We don't have these Z_i but we do have $\tilde{Z}_i = (Y_i - \hat{f}(X_i))^2$, and it's easy to show that $\frac{\sum \tilde{Z}_i \mathbf{1}_{X_i \in I(x)}}{\sum \mathbf{1}_{X_i \in I(x)} - 1} \to \frac{\sum Z_i \mathbf{1}_{X_i \in I(x)}}{\sum \mathbf{1}_{X_i \in I(x)}}$ by adding and subtracting $f(x)$ inside the square. This gives us a strongly consistent estimator of our conditional variance as required.

□

Proof of Lemma 2.5.1. Since the Stage 1 sampling uses Algorithm 1 our $n_{(1),k}$ are fixed (though this could be extended to randomized version of Algorithm 1). The proof is mostly algebra, using the fact that $\hat{\beta}_{(1),k}$ is conditionally independent of $\hat{\beta}_{(2),k}$ given $n_{(2),k}$.

$$
\begin{aligned}
E[(\hat{\beta}_k - \tilde{\beta}_k)^2] &= E\Big[\big(\frac{n_{(1)}(\hat{\beta}_{(1),k} - \tilde{\beta}_k)}{n_{(1),k} + n_{(2),k}} + \frac{n_{(2)}(\hat{\beta}_{(2),k} - \tilde{\beta}_k)}{n_{(1),k} + n_{(2),k}}\big)^2\Big] \\
&= E_{n_{(2),k}}\Big[\frac{n_{(1),k}^2}{(n_{(1),k} + n_{(2),k})^2} E_{D_{1:n_{(1)}}}\big((\hat{\beta}_{(1),k} - \tilde{\beta}_k)^2 | n_{(2),k}\big) \\
&+ 2\frac{n_{(1),k} n_{(2),k}}{(n_{(1),k} + n_{(2),k})^2} E_{D_{1:n_{(1)}}}\big((\hat{\beta}_{(1),k} - \tilde{\beta}_k) | n_{(2),k}\big) \\
&\times E_{D_{n_{(1)}+1:n}}\big((\hat{\beta}_{(2),k} - \tilde{\beta}_k) | n_{(2),k}\big) \\
&+ \frac{n_{(2),k}^2}{(n_{(1),k} + n_{(2),k})^2} E_{D_{n_{(1)}+1:n}}\big((\hat{\beta}_{(2),k} - \tilde{\beta}_k)^2 | n_{(2),k}\big)\Big].
\end{aligned}
$$

We have that

$$
E_{D_{n_{(1)}+1:n}}\big((\hat{\beta}_{(2),k} - \tilde{\beta}_k) | n_{(2),k}\big) = 0,\ E_{D_{n_{(1)+1:n}}}\big((\hat{\beta}_{(2),k} - \tilde{\beta}_k)^2 | n_{(2),k}\big) = \frac{\sigma_{Y,k}^2}{n_{(2),k}}
$$

which gives us the desired result.

□

Proof of Theorem 2.5.2. By assumption we have that Y_i's are Normally distributed. We first deal with the dependence $E_{D_{1:n_1}}\big((\hat{\beta}_{(1),k} - \tilde{\beta}_{(1),k})^2 | n_{(2),k}\big)$. A well known property of

the Normal distribution Sen (2012) is that the estimate of the mean $\hat{\beta}_{(1),k}$ and the estimate of the variance $\hat{\sigma}^2_{Y,k}$ are independent. This immediately gives that $E_{D_{1:n_1}}\big((\hat{\beta}_{(1),k} - \tilde{\beta}_{(1),k})^2 | n_{(2),k}\big) = E_{D_{1:n_1}}\big((\hat{\beta}_{(1),k} - \tilde{\beta}_{(1),k})^2\big) = \frac{\sigma^2_{Y,k}}{n_{(1),k}}$ as there is no dependence between $\hat{\beta}_{(1),k}$ and $n_{(2),k}$. Thus we get that the risk variance for that leaf is just as from Lemma 2.4.1.

Now we want to bound the probability $\hat{n}_k$ above is far away from n^*_k. We will do this by bounding the a_k. Another well known property of the normal distribution is $\frac{(n_k-1)S^2_{Y,k}}{\sigma^2_{Y,k}} = (n_k - 1)a_k \sim \chi^2_{(n_k-1)}$. By characterization of sub-exponential random variables:

$$P((n_k-1)|a_k-(n-1)| > \sqrt{2(n_k-1)t} + 2t) \leq e^{-t}$$

$$P(|a_k - 1| > \frac{\sqrt{2t}}{\sqrt{(n_k-1)}} + \frac{2t}{(n_k-1)}) \leq e^{-t}$$

$$\frac{\sqrt{2t}}{\sqrt{(n_k-1)}} + \frac{2t}{(n_k-1)} \in (0,1) \implies \frac{2t}{(n_k-1)} \leq 1 \implies \frac{2t}{(n_k-1)} < \frac{\sqrt{2t}}{\sqrt{(n_k-1)}}$$

$$\implies P(|a_k - 1| > \frac{2\sqrt{2t}}{\sqrt{(n_k-1)}}) \leq P(|a_k - 1| > \frac{\sqrt{2t}}{\sqrt{(n_k-1)}} + \frac{2t}{(n_k-1)}) \leq e^{-t}$$

$$\forall\, \alpha \in (0,1)$$

$$P(|a_k - 1| > \alpha) \leq e^{-\frac{(n_k-1)\alpha^2}{8}}$$

$$P(\exists k \text{ s.t. } |a_k - 1| > \alpha) \leq \sum_{k=1}^{K} e^{-\frac{(n_k-1)\alpha^2}{8}}.$$

And now we apply Lemma 2.4.2 to bound the excess. Now we assume our purely random tree is a Mondrian Tree with the above assumptions, so $n_k = \frac{cn}{K}$. By Markov inequality and Proposition 2 in Mourtada et al. (2018) we have that:

$$P(K_n - 1 > n^{\frac{d+\epsilon}{d+2}}) \leq \frac{E[K_n]}{n^{\frac{d+\epsilon}{d+2}}} = \frac{(1 + n^{\frac{1}{2+d}})^d}{n^{\frac{d+\epsilon}{d+2}}} = \delta_1$$

$$P(\exists k \text{ s.t. } |a_k - 1| > \alpha | K_n \leq n^{\frac{d+\epsilon}{d+2}}) \leq n^{\frac{d+\epsilon}{d+2}} e^{-\frac{\alpha^2}{8}((cn)^{\frac{2-\epsilon}{d+2}} - 1)} = \delta_2$$

By setting $\epsilon = 1$ and using the union bound we get the result.

Remark. It is worth noting that in the above proof we have only used the property that χ^2 are subexponetial. A slightly stronger (in terms of n, α) inequality is possible using Chernoff Bounds and exploiting the structure of χ^2 random variables.

□

2.9 Appendix B: Additional experiments and discussion

2.9.1 Dependence in non-normal case

We are interested in the question of when is $E_{D_{1:n_{(1)}}}\left[(\hat{\beta}_{(1),k} - \tilde{\beta}_k)^2 | n_{(2),k} < n^*_{(2),k}\right] < E_{D_{1:n_{(1)}}}\left[(\hat{\beta}_{(1),k} - \tilde{\beta}_k)^2\right]$. Unfortunately $n_{(2),k}$ is a function not only of $\hat{\sigma}^2_{(1),k}$ but of all other $\hat{\sigma}^2_{(1),l}$. Let us start with a more simple and general question of when $E\left[(\hat{\mu} - \mu)^2 | \hat{\sigma}^2 < \sigma^2\right] < E\left[(\hat{\mu} - \mu)^2\right]$. We present no formal arguments here but rather share our findings and conjectures which we consider both interesting in their own right as well as excellent candidates for further study. The first observation is that far from this being an unusual property this seems to be a fairly common property. In fact for symmetric distributions the relationship appears to be well behaved. From Sen (2012) the sample mean and sample variance are asymptotically MVN (multivariate normal) with cross correlation equal to the skew, so when our distribution is symmetric the sample mean and sample variance are independent in the limit. For the finite sample case the relationship between $\hat{\sigma}^2_{1,k} - \sigma^2_{1,k}$ and $E\left[(\hat{\mu} - \mu)^2 | \hat{\sigma}^2 - \sigma^2\right] - E\left[(\hat{\mu} - \mu)^2\right]$ appear to be monotonic and to go through the origin (so when the sample variance is the true variance, the conditional variance of the sample mean is the unconditioned variance, which is what we would hope is the case). In fact it appears both the magnitude and parity of this relationship depends on the *excess kurtosis* $\kappa - 3$. If $\kappa - 3 < 0$ this relationship is negative and if $\kappa - 3 > 0$ this relationship is positive, with the magnitude increasing as you move further away from zero.

If these observations are true for all symmetric distributions it would be quite fortuitous, since large values of κ imply that the estimates of our variances will be more noisy, but those are exactly the cases where actively fitting to the sample variance of our first stage is beneficial: If our sample variance is larger than the population variance, then the variance of our $\hat{\beta}_{(1),k}$ is larger than expected, so it is beneficial to use more points in the second stage than the optimal passive sampling would have assigned. Meanwhile when a smaller sample variance implies the variance of our $\hat{\beta}_{(1),k}$ is larger than expected, κ is small and so our sample variance will itself have small variance. We have not yet been able to prove this relationship, and things become much more complicated in the more realistic case where our distribution is skewed. However these results give us confidence that things are unlikely to go too badly wrong when our labels are not normally distributed.

2.9.2 Experimental data set info

For both simulations our marginal X distribution was uniform over the space $[0,1]^{10}$. Heteroskedastic simulation had constant regression function and Gaussian noise, with space split into high variance region (25) and low variance region (1). Varying complexity had sinusodial regression function $f(x) = C\sin(\frac{2\pi}{d*F} * \sum x_i)$ and Gaussian noise with constant (1) variance. It was split into high variation region ($C = 20, F = 0.05$) and low variation region ($C = 5, F = 0.1$). For both sets $[0.1,1]^{10}$ were the high areas, with everything else a low area.

2.9.3 Practical considerations

Here we compile information related to actually using this active learning method in practice.

2.9.3.1 Heuristics to deal with difference between theory n_k^* and possible values

There are many reason why you may not actually be able to sample according to your estimates of the optimal n_k^*. For a start our n_k^* will almost always be fractional. Additionally there may be less than n_k^* points in a leaf. These issues are fairly minor and become less influential as sample sizes increase. However a more consistent issue that occurs when using the approximating algorithm is when a leaf is oversampled during stage 1, so that $n_{(1),k} > n_k^*$. This means that some other leaf will get fewer than it is optimal number of samples. Although this again can be dealt with asymptotically by making our stage 1 a small fraction of the total number of samples, in practice this is a problem which often occurs when our sample size is not large.

In our code we implemented heuristics to deal with these mismatches. We emphasize that these heuristics are subjective and one could easily use or argue for others. After calculating our $\hat{n}_k$ we immediately floor them all. We then set $\hat{n}_k = \max(\min(\hat{n}_k, \eta_k), n_{1,k})$ (where η_k is the total number of points in leaf k). It is possible that $\sum \hat{n}_k \neq n$ after these adjustments. If we have too many points, we reduce the largest $\hat{n}_k$ until we achieve the correct total. If we have too few points we increase the $\hat{n}_k$ by 1 each, starting with the smallest, and starting over once we have increased them all by 1. This asymmetry is because increasing small values can have a large reduction on the variance of the estimate, but decreasing large ones leads to a small increase in variance.

2.9.3.2 Lifetime parameter sequence

We have found that the best general form for the lifetime parameter sequence is $\lambda_n = \frac{1}{\gamma}(n^{\frac{2}{2+d}} - 1)$. The γ can be fairly freely chosen with $\gamma = 1$ a reasonable default (and is what is used in all simulations and experiments in this chapter), but the -1 is very important; it ensures that we do not start with a lifetime $= 1$ for $n = 1, \forall\, d$ as when d is large this can result in a very large number of leaves early on.

2.9.3.3 Sampling method during stage 1

During stage one our theory assumed that $n_1 = cn$ and then each leaf received the same fraction of points, as this gives important asymptotic properties. In practice if c is too large this can result in putting too many samples in certain small leaves during stage 1, so that $n_{1,k} > n_k^*$, meaning that we have oversampled this leaf and will have to reduce other sampling elsewhere. One way of avoiding this is by making c small, but this risks getting bad leaf estimates and suboptimal stage 2 sampling unless n is large, where the n required increases as d increases. Another is to sample passively. We have found that generally if $c = 0.5$ then sampling passively tends to produce pretty good results unless your function has massive amounts of variation. Another option is to use a hybrid sampling scheme in stage 1, where each leaf is given a small number of samples, and then the rest of the samples are distributed randomly, but empirically this seems to be worse than random sampling for small values of n.

2.9.3.4 Final regression model

As shown in our experiments, although most the theory assumes that you are using the same tree for your active learning as you are for your final predictions, you also get good results doing active learning with Mondrian Forests, and then taking that data and fitting your final model with a more data adaptive model, although not always.

2.9.3.5 Using more than 2 stages

It is of course possible to do more than 2 stages, updating your estimates of the leaf variances during each stage to guide sampling during the next stage. We found that in practice the benefits of doing this are generally fairly small. Of course the first stage should still be sufficiently large that you get decent initial estimates for the leaf variances. Much of the theory could be extended to increasing number of stages as long that the number is not increasing with n without much work. Increasing the number of stages as n increases may require additional care and effort.

2.9.4 Forests

Just as with Breiman decision trees you can ensemble purely random trees into forests. These forests show improved performance at the cost of increased computational cost since they average out the random process used to build the trees. We also have an intuitive (though theory free) extension of our active learning method to utilize the power of multiple Mondrian Trees. The idea is each tree determines the optimal number of samples per leaf in the usual way, and then gives data points weights such that the expected number of points sampled from each leaf is the optimal number. These probabilities are then averaged out over all the trees in the forest and the new points are sampled using these averaged probabilities. The formal algorithm is given below:

Algorithm 3: Forest version of oracle approximation algorithm

Input: Leaves of our T trees $\mathcal{I}_1 ... \mathcal{I}_T$, pool of data points $\{X_i\}_{i=1}^m$, and label budgets $n_{(1)}, n_{(2)}, n = n_{(1)} + n_{(2)}$.

Output: The set of labelled points.

Stage 1: ;

Sample $n_{(1)}$ data points (possibly according to the structure of the trees $\mathcal{I}_t$) using a version of algorithm 1. ;

foreach *t* **do**

  Use those samples (X_i, Y_i) to estimate $\hat{\sigma}^2_{Y,k,t}$ for each leaf. ;

end

Stage 2: ;

foreach *t* **do**

  foreach $I_{k,t} \in \mathcal{I}_t$ **do**

    Calculate $\hat{n}_{k,t} = n \frac{\sqrt{p_{k,t} \hat{\sigma}^2_{Y,k,t}}}{\sum_{k'} \sqrt{p_{k',t} \hat{\sigma}^2_{Y,k',t}}}$ the number of points in the leaf to sample. ;

    Count $m_{k,t}$ the number of unlabelled points in leaf $I_{k,t}$;

    foreach *Unlabelled* $X_i \in I_{k,t}$ **do**

      Assign weight $W_{i,t} = \frac{\hat{n}_{k,t} - n_{(1),k,t}}{n_2 * m_{k,t}}$. ;

    end

  end

end

foreach *Unlabelled* X_i **do**

  Final weight $W_i = \frac{1}{T} \sum W_{i,t}$. ;

end

Sample $n_{(2)}$ points with weights W_i.

Below we show the results of using Mondrian Forests for our active learning, and both Mondrian Forests and Random Forests as our final regression model. Here we see some benefit using Mondrian Forest for active learning and then Random Forests for our final

regressor (although in fact the naive uncertainty sampling method outperforms ours). Although the benefit on the real data appears to be a small constant factor, the actively learned models provide similar accuracy with 10s of fewer data points, which can be significant.

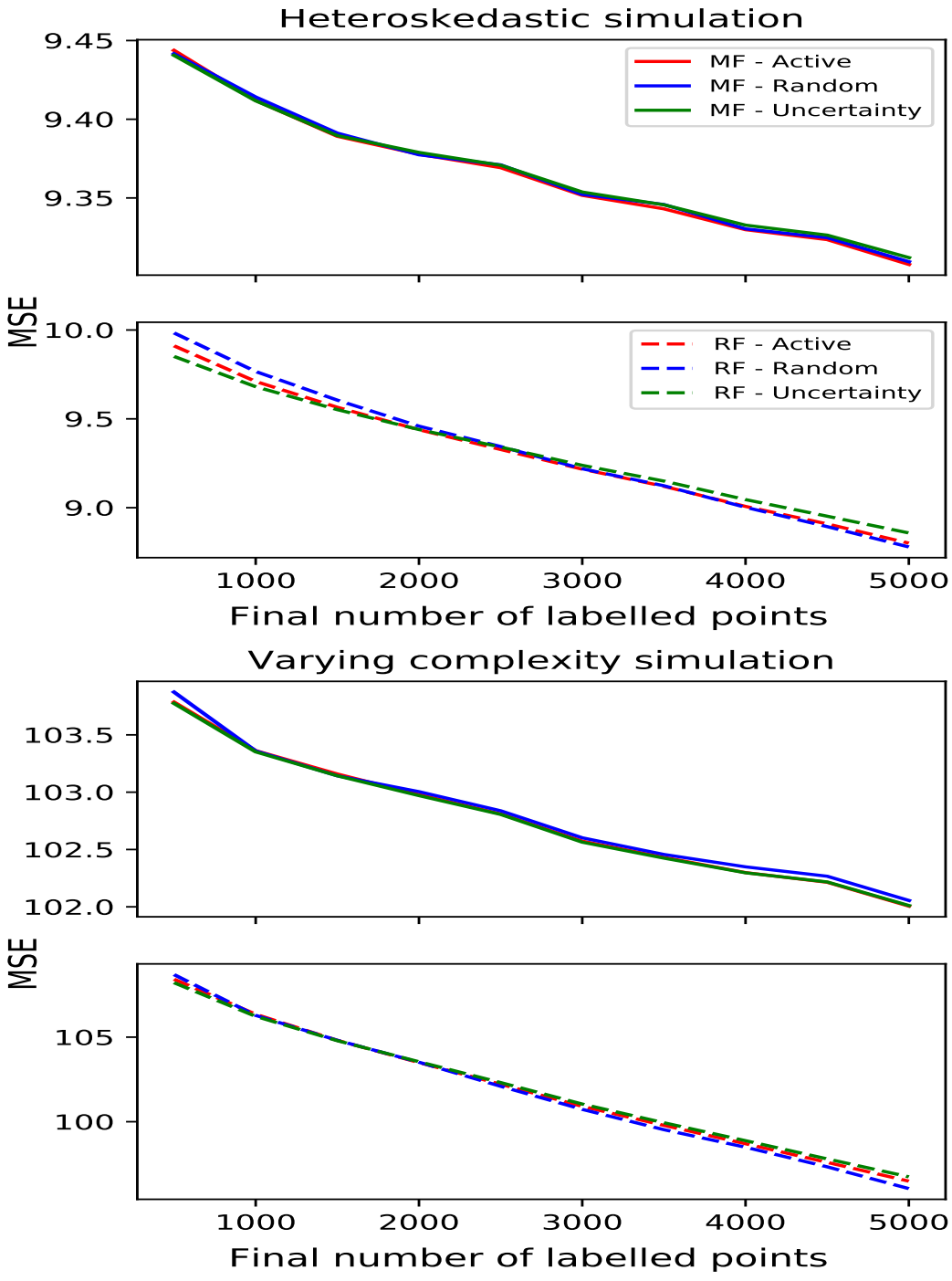

Figure 2.3: Mondrian Forest active learning simulations

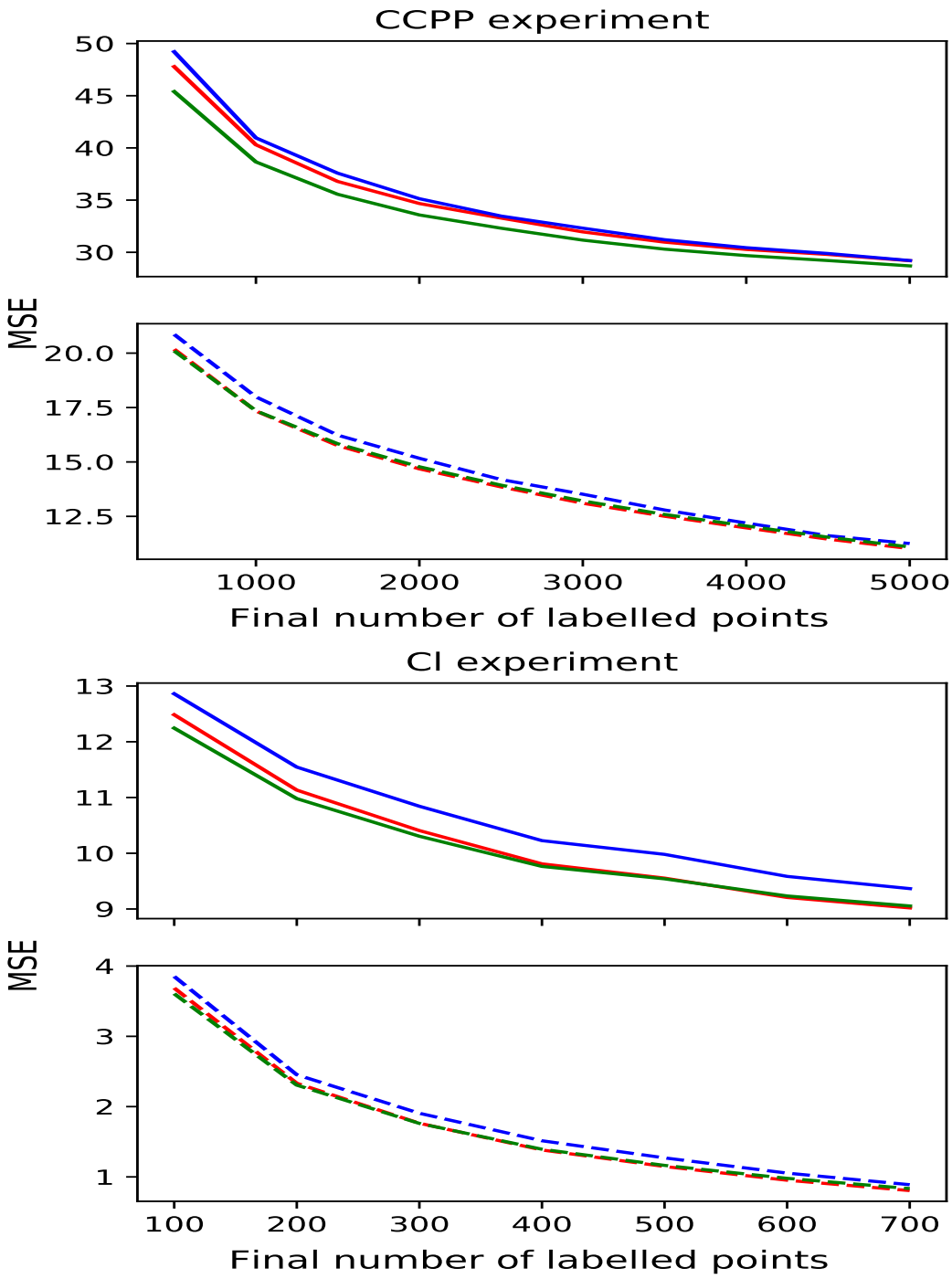

Figure 2.4: Mondrian Forest active learning experiments

2.9.5 Additional experimental results

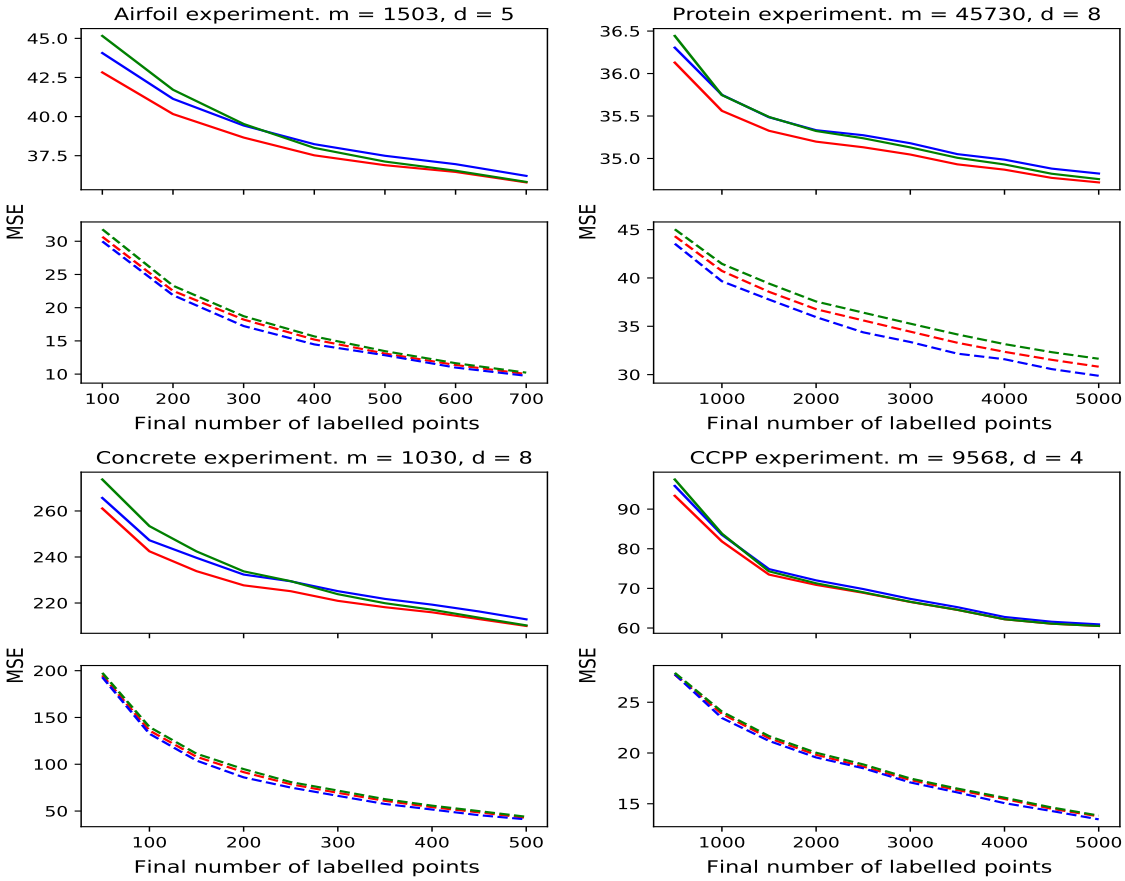

Figure 2.5: Additional active learning experiments on UCI data with Mondrian Trees

CHAPTER III

Consistency of Weighted Averaging Estimators Under Active Learning

Active learning seeks to build the best possible model with a budget of labelled data by sequentially selecting the next point to label. However the training set is no longer *iid*, violating the conditions required by existing consistency results. Inspired by the success of Stone's Theorem (Györfi et al., 2006) we aim to regain consistency for weighted averaging estimators under active learning. Based on ideas in Dasgupta (2012), our approach is to enforce a small amount of random sampling by running an augmented version of the underlying active learning algorithm. We generalize Stone's Theorem in the noise free setting, proving consistency for well known classifiers such as k-NN, histogram and kernel estimators under conditions which mirror classical results. However in the presence of noise we can no longer deal with these estimators in a unified manner; for some satisfying this condition also guarantees sufficiency in the noisy case, while for others we can achieve near perfect inconsistency while this condition holds. Finally we provide conditions for consistency in the presence of noise, which give insight into why these estimators can behave so differently under the combination of noise and active learning.

3.1 Introduction

Active learning results in training data which is neither independent, nor from the same distribution on our covariates as the test data (which we assume we have no control over and which is drawn *iid* from some underlying joint distribution $P_{X,Y}$). Thus even if our classification algorithm is well studied, standard results on consistency of that classifier, arguably the minimal requirement for a good method, no longer apply. The loss of consistency is of practical concern as even popular active learning algorithms can induce inconsistency (Dasgupta, 2011). Can we recover this lost consistency?

We begin to answer this question by focusing on weighted averaging binary classifiers, of which k-NN, histogram and kernel estimators (Devroye et al., 2013) are the classic examples. Under *iid* assumptions consistency of these is largely covered by the celebrated Stone's Theorem (Stone, 1977), and our goal is to generalize these results to an actively selected training set. However it is clear that if our active learning method can be completely arbitrary, there is not much hope of obtaining consistency. Adapting a requirement in Dasgupta (2012), we begin by introducing a method to *augment* any existing active learning algorithm, which only influences the sampling policy a vanishing fraction of the time.

In the noiseless setting this augmentation is sufficient, and consistency of the above classical estimators is proven using a technical condition. However in the presence of noise the behaviour of these classical estimators diverges sharply; for histogram estimators satisfying this condition guarantees consistency even with noise, whereas for k-nn we provide a counterexample where the condition is satisfied, but we achieve maximal risk. Finally we will provide additional conditions which provide consistency under noise, and which illustrate the differences between histogram and k-nn which lead to starkly different behaviour.

The structure of this chapter is as follows:

1. Give a natural augmentation to sequential active learning algorithms (Algorithm 4).

2. Prove that in the noiseless setting and under this augmentation, histogram and k-nn are consistent for **any** such active learning algorithm (Proposition 3.4.1). These are proved by providing a sufficient condition (Condition 1) for consistency for any weighted averaging estimator under any such active learning algorithm (Theorem 3.4.1).

3. Showing the histogram estimator is still consistent under this condition even in the noisy setting (Proposition 3.5.1).

4. Providing a counterexample in the noisy setting where k-nn satisfies our condition, but achieves the largest Risk possible (Theorems 3.5.1 and 3.5.2) under an (augmented) adversarial active learning algorithm.

5. Provide further conditions (Condition 2) which are sufficient for consistency in the noisy setting, which show why histogram is sufficient but k-nn is not (Theorem 3.6.1).

6. Describe the k-nn case (Section 3.8).

7. Give details of all proofs (Section 3.9).

3.2 Setting and background

Our results contain both positive and negative results. Positive results give conditions under which we maintain consistency under any (possibly adversarial) active learning algorithm. Our negative results provide counterexamples showing when such conditions are not sufficient for consistency. Our positive results will be in the query synthesis setting, as this allows for the strongest possible family of adversarial active learning algorithms (as the adversary can select arbitrary points). Conversely our negative results will be in the pool setting, restricting ourselves to a weaker family of active learning algorithms (as the adversary will be restricted to a pool of points generated by nature). Each result is proved in the more challenging setting (and will also be valid for the less challenging setting), confirming that the results are not just artifacts of our choice of active learning setting.

Our setup will be fairly standard for active learning (Settles, 2012). In the query synthesis setting the active learning algorithm can select any point within the support of the marginal test distribution P_X. In the pool setting the algorithm will select n data points to label out of a pool of m_n data points, where the size of our initial pool depends on how many labelled points we will select. Let $D_n = \{(X_i, Y_i)\}_{i=1}^{m_n}$ be our pool with known covariates $X_i \in \mathcal{X} \subset \mathcal{R}^d$ and hidden labels $Y_i \in \{0, 1\}$, where $(X_i, Y_i) \overset{iid}{\sim} P_{X,Y} = P_{Y|X} P_X$, $f(x) = P(Y = 1|X = x)$ and with Bayes classifier $f^*(x) = \mathbf{1}_{f(x)>1/2}$. We will assume that $\mathcal{X}$ is a bounded subset of $\mathcal{R}^d$, however if this does not hold then many of our results can be applied on a sphere centered at the origin with all but an arbitrary ϵ of the probability mass to extend the results beyond bounded $\mathcal{X}$. Additionally let $D_n(X)$ and $D_n(Y)$ denote just the X and Y of the pool respectively. Note that the pool setting is slightly different from the setup in Hanneke (2014); Hanneke et al. (2019), as our setting assumes $m_n < \infty$ while theirs assumes $m_n = \infty \; \forall \; n$.

The algorithm will create a labelled subset S_n with the goal of minimizing the risk $E\left[\mathbf{1}_{f_n(X,S_n)\neq Y}\right]$. The notation $f_n(x, S_n)$ indicates the prediction given at point x when trained on the labelled data S_n (with $S_n(X), S_n(Y)$ as just the covariates and labels). We use lower case letter x to denote non-random quantities and upper case X to denote random ones. We will use *passive sampling* to refer to sampling according to the marginal P_X. In the pool setting given S_n, let S_n^c be the remaining $m_n - n$ unlabelled data points, with $\emptyset^c = D_n(X)$ (so it's not exactly a true complement operator but has a similar flavor). Our (potentially randomized) active learning algorithm selecting the i^{th} point will be $A : S_{i-1} \to supp(P_X)$ in the query synthesis setting and $A : S_{i-1} \times S_{i-1}^c \to S_{i-1}^c$ in the pool setting. Technically S_n is a multiset and so can contain identical 2-tuples (X_i, Y_i). Unless otherwise specified, all references to consistency will be weak (in probability) consistency.

We will focus on weighted averaging estimators for classification (Devroye et al., 2013), where the estimators take the following form (where $W_{ni}(x) = W_{ni}(x, S_n(X))$):

$$f_n(x, S_n) = \begin{cases} 0 & \text{if } \sum_{(X_i,Y_i)\in S_n} Y_i W_{ni}(x) \leq \frac{1}{2} \\ 1 & \text{otherwise} \end{cases}$$

We will make the following assumptions about the structure of our functions $W_{ni}(x)$.

$$W_{ni}(x) \geq 0, \quad \sum W_{ni}(x) \leq 1$$

The inconsistency introduced during active learning is well studied (Beygelzimer et al., 2011), and even in the one dimensional case popular and intuitive algorithms can be inconsistent in non-pathological examples (Dasgupta, 2011). One of the first and best known general augmentations for providing consistency in active learning is importance weighting (Beygelzimer et al., 2009; Sugiyama and Kawanabe, 2012). This technique is very successful, but requires that probabilities of selecting points be non-zero, and therefore cannot be applied to deterministic active learning algorithms. In contrast our results do apply to deterministic and non-deterministic algorithms. A recent study (Loog and Yang, 2016) showed that while most active learning methods examined performed well on many data sets, they also had data sets on which they do not appear to be converging to the performance of random sampling. Our work extends that of Dasgupta (2012), which studied consistent active learning for nearest neighbor estimators in the streaming setting.

3.3 Augmented algorithm

Without any structure on the sampling process it would be impossible to provide conditions on the estimator which guarantee consistency for an arbitrary active learning algorithm A. For example it is clear that an algorithm which samples the same location forever will not (in general) be consistent. At the same time we do not want to constrain our active learning algorithm too much. Our proposal, based on (R1) in Dasgupta (2012), is a simple and intuitive augmentation which is relatively inexpensive. The idea is to occasionally ignore our active learning algorithm and instead sample according to the underlying P_X. In query synthesis this is done directly, and in the pool setting this is done by sampling uniformly from the unlabelled data.

The augmented algorithm is still an active learning algorithm. However we will refer to it as the augmented algorithm to avoid confusion with the active learning algorithm A which it augments. We impose the following requirements on our sequence of p_i:

Algorithm 4: Augmented Algorithm for pool setting

Input: Active learning algorithm A, number of samples n, probability sequence $(p_1, ..., p_n)$, unlabelled data $D_n(X)$

Output: Labelled data set S_n

$S_0 = \emptyset$;

for i *from* 1 *to* n **do**

  Draw an independent Bernoulli random variable Z_i with $P(Z_i = 1) = p_i$;

  if $Z_i = 1$ **then**

    Select X_i uniformly at random from S_{i-1}^c

  else

    Select X_i according to $A(S_{i-1}, S_{i-1}^c)$

  end

  Query selected point and receive Y_i ;

  $S_i = S_{i-1} \cup (X_i, Y_i)$;

end

Remark. In the Query Synthesis setting, if $Z_i = 1$ then our augmented algorithm will simply draw X according to P_X and Y from $P_{Y|X}$, and the full algorithm is in the appendix.

$$p_i \searrow 0, \qquad \sum_{i=1}^{\infty} p_i = \infty$$

The first requirement ensures that asymptotically the fraction of your data set which is sampled randomly goes to 0, and that as you collect more data, you are more likely to exploit the information you have and sample actively. The second requirement ensures we will sample at random infinitely often, even though the fraction of samples chosen randomly is asymptotically negligible. These are very similar to requirements for the ϵ-greedy approach (Sutton and Barto, 1998) with decaying ϵ_n for multi-armed bandits.

3.4 Sufficiency in the noise free case

We first consider the more simple noise free case, where $f(x) = P(Y = 1|X = x) \in \{0, 1\}$. We impose the following Regularity Condition on our underlying distribution: that the boundary between the two classes has $[P_X]-$measure 0:

Regularity Condition 1. Assume we are in the noise free setting, i.e., $Y = f(X)$ almost surely. Use $B_{x,r}$ to denote the ball of radius r centered at x. Let $\mathcal{X}_0 \subset \mathcal{X}$ be $\mathcal{X}_0 = \{x \in \mathcal{X} : \exists\, B = B_{x,r}, r > 0, P_X(B) > 0, f(z) = 0\ \forall\ z \in B\}$ and define $\mathcal{X}_1$ similarly. Then $P_X(\mathcal{X}_0 \cup \mathcal{X}_1) = 1$.

Under this Regularity Condition and using the augmentation in Algorithm 4, classic weighted averaging estimators (Györfi et al., 2006) can all be made consistent for any base active learning algorithm A.

Proposition 3.4.1. *Assume Regularity Condition 1, and sample using Algorithm 4 with any active learning algorithm A. Let $s_n = \sum_{i=1}^{n} p_i$. Then the following estimators are consistent:*

- *The histogram estimator with cell lengths h_n if $h_n \to 0, h_n^d s_n \to \infty$.*
- *k-nn with neighbor cardinality k_n if $\frac{k_n}{s_n} \to 0$.*

Additionally similar results can be proven for many standard bounded support kernel estimators under the condition that the bandwidth $h_n \to 0, h_n^d s_n \to \infty$. These conditions are almost the same as the conditions derived using Stone's Theorem under *iid* sampling, except n the number of samples has been replaced by s_n the (expected) number of random (*iid* from P_X) samples.

The consistency of these is provided by a single unifying condition. The statement of the condition is somewhat technical, and we will discuss why such technicality is needed. Let $\tilde{X}_i = \tilde{X}_i(X_i, \mathbf{1}_{E_i}, V_i) = X_i \mathbf{1}_{E_i} + V_i(1 - \mathbf{1}_{E_i})$. We will define a (family of) function $g_n : \mathcal{X} \times \mathbf{R}_+ \times \mathcal{X}^n \times \{0,1\}^n \to [0,1]$ by:

$$g_n(x, r, \{X_i\}^n, \{\mathbf{1}_{E_i}\}^n) = \inf_{\{V_i\} \in supp(\mathcal{X})} \sum_{i=1}^{n} W_{ni}(x, \{\tilde{X}_i\}^n) \mathbf{1}_{\tilde{X}_i \in B_{x,r}}$$

Note that if $\mathbf{1}_{E_i} = 0$ then the value of X_i does not matter. That is

$$g_n(...x_i = a...\mathbf{1}_{E_i} = 0...) = g_n(...x_i = b...\mathbf{1}_{E_i} = 0...) \; \forall \, a, b, \{x_j\}_{j \neq i} \{\mathbf{1}_{E_j}\}_{j \neq i}$$

Now assume we are sampling (Z_i, X_i) according to our augmented active learning algorithm, and let $E_i = \{Z_i = 1\} \cap \{X_i \in B_{x,r}\}$. Then our Condition is the following:

Condition 1. Let $X, X_i \sim P_X$ and $Z_i \sim B(p_i)$. Assume $\exists \, H_n \; s.t. \frac{H_n}{s_n} \to 0$ and $\forall \, r > 0$:

$$\underset{X}{E} \underset{Z_i}{E} \underset{X_i}{E} \Big[g_n(X, r, \{X_i\}^n, \{\mathbf{1}_{E_i}\}^n) | \sum \mathbf{1}_{E_i} \geq H_n \Big] \to 1$$

Theorem 3.4.1. *Assume Regularity Condition 1, that data is sampled according to Algorithm 4 with any Active Learning algorithm A. If predictions are made with a weighted averaging estimator satisfying Condition 1 then $E\big[\mathbf{1}_{f_n(X, S_n) \neq Y}\big] \to 0$.*

Condition 1 ensures that predictions are eventually made only using data within an arbitrarily small neighborhood, that those small neighborhoods are non empty, and that the weight of all data in these neighborhoods cannot be nullified by adversarial placement of additional points. The families of estimators which satisfy Stone's Theorem but not this are largely pathological. One example would be a version of the histogram estimator where data points which are within a certain distance d_n (decreasing as $n \to \infty$) of another data point are given $W_{ni}(x) = 0 \forall x$. If d_n decreases quickly enough then under random sampling the fraction of data which is nullified will be vanishing and so this estimator would behave the same way as the standard histogram. However an adversarial active learning algorithm can sample within d_n of every randomly sampled data point, giving all randomly sampled data weight of 0 and nullifying the augmentation.

3.5 Examples in the noisy case

We now move beyond the noise free setting and allow for $f(x) = P(Y = 1|X = x) \in [0, 1]$. Following Dasgupta (2012) we will assume a Regularity Condition on $f(x)$:

Regularity Condition 2. If the support of P_X is $\{x \in \mathcal{X} : P_X(B_{x,r}) > 0 \, \forall \, r > 0\}$ then $\forall x$ in the support of P_X x is a continuity point of $f(x)$.

This condition gives us the following property: for all x except on a set of P_X measure 0, and for any $\epsilon > 0$ there is a ball $B_{x,r}$ with $P_X(B_{x,r}) > 0$ such that $|f(x) - f(z)| < \epsilon \forall z \in B_{x,r}$. We will also assume that $P_X(\{x \in \mathcal{X} : f(x) = \frac{1}{2}\}) = 0$ to remove uninteresting qualifications during statements and proofs. Under these assumptions, is Condition 1 still sufficient for consistency?

3.5.1 Histogram Estimators

We begin with the positive case by showing that for the histogram estimator, properties required for Condition 1 also give consistency in the noisy setting. As shown in the proof of Proposition 3.5.1, Condition 1 holds for the histogram iff $h_n \to 0, h_n^d s_n \to \infty$, and the proof shows that if Condition 1 is satisfied, the probability of our test point falling in a partition with only M data points goes to 0 for all $M < \infty$. Under our Regularity Condition 2 this is sufficient for consistency

Proposition 3.5.1. *Under Regularity Condition 2, a histogram classifier with*

$$h_n \to 0, h_n^d s_n \to \infty$$

is consistent for any base active learning algorithm.

Therefore properties of our histogram required to satisfy Condition 1 (and therefore give consistency in the noise free case) also give consistency in the noisy case.

3.5.2 Nearest Neighbor Estimators

We now present an example where you can satisfy Condition 1 but are not consistent in the noisy setting, using nearest neighbors as our underlying estimator. In our counterexample the Bayes Risk will be η for some $\eta > 0$ but arbitrarily small, but the risk of our augmented algorithm will be $1 - \eta$. We will present the example for 1-NN since the intuition is strongest here, but the example generalizes when $k_n \to \infty, \frac{k_n}{s_n} \to 0$, and we will give the corresponding theorem and proof in the appendix. Although 1-NN is not consistent when there is noise present under passive sampling, it achieves within a factor of 2 from the optimal risk R^* of the Bayes classifier (Cover and Hart, 1967) whereas in our counter example it has risk close to 1.

Let $\mathcal{X} = [0, 1]$, $X_i \sim U[0, 1]$ and $Y_i|X_i \sim Bern(\eta), 0 < \eta < \frac{1}{2}$ (so we trivially satisfy Regularity Condition 2). Note here that the Bayes classifier $f^*(x)$ always predicts the class 0 and has risk η. Let $f(x, S_n)$ be the prediction of a 1-NN learner at point x trained on the data set S_n. This example will assume we are in the pool setting (although the translation of the example to the query synthesis setting is clear). Recall that m_n is the size of our unlabelled pool from which we select n points. We assume that acquiring unlabelled data is effectively free compared with the cost of labelling the data. In particular we will assume that $\frac{n}{m_n} \to 0$.

We will again use augmented Algorithm 4. However our base active learning algorithm will be a specific active learning algorithm $A^\dagger$ defined in the next section, which is an adversarial active learning algorithm, developed purely to test the sufficiency claim of Theorem 3.4.1 when we do not assume Regularity Condition 1. We will first describe informally what the algorithm does and how it achieves it's asymptotically near perfect riskiness before presenting the proof.

3.5.2.1 Informal proof outline

During this subsection, we will let X_i be the i^{th} point sampled, and let the ordered random variables $X_{(i)}$ denote ordering of the unlabelled data on the interval $[0, 1]$. We will sample according to Algorithm 4, with a specific active learning Algorithm $A^\dagger$. The active learning algorithm $A^\dagger$ will work in the following way: Given S_t and S_t^c, we can define *open points* as unlabelled data points whose left or right neighbor are labelled as 0:

Definition 3.5.1. Let $L^t(X)$ denote the known label of point X at some time t, with $L^t(X) = ?$ if the point is unlabelled at iteration t. Then a point $X_{(i)}$ is an *open point* at time t if $L^t(X_{(i)}) = ?, L^t(X_{(i+1)}) = 0$ or $L^t(X_{(i)}) = ?, L^t(X_{(i-1)}) = 0$.

Algorithm 5: Adversarial Active Learning algorithm $A^\dagger$

Input: Currently labelled data S_t, unlabelled data S_t^c
Output: The next point to label
if *There is at least one open point* **then**
  Sample the smallest open point.
else
  Sample the unlabelled data point which is furthest from a labelled data point.
end

Notice that whenever an open point is labelled, it is no longer an open point. If the label of that (former) open point is 0 then it (usually) creates another open point adjacent to it, and if it is 1 then it does not create a new open point. The results of this is that we will sample consecutive points in a line, creating *interior points* which are labelled point who's left and right neighbor are both labelled:

Definition 3.5.2. $X_{(i)}$ is an *interior point* at time t if $L^t(X_{(i-1)}), L^t(X_{(i)})$, and $L^t(X_{(i+1)})$ are all labelled at time t.

These interior points (plus the two points at each end) form *intervals*:

Definition 3.5.3. An *interval* is a groups of consecutive labelled points (and we allow singleton points to be intervals of length 1).

Our active learning algorithm $A^\dagger$ samples consecutive points until we get a point who's label is 1, which can be thought of as having 'closed off' that side of the interval. The expected distance between these interior points is $\frac{1}{m_n+1}$. By construction all points with label 0 are interior points, or are adjacent to open points. We will show that eventually almost all points with the label 0 are interior points.

We then define the *coverage* of a point as the area where they are the nearest neighbor:

Definition 3.5.4. The coverage of a point x is $I(x, S_n) = \int \mathbf{1}_{x=\underset{x' \in S_n}{\arg\min} |z-x'|} \, dz$

Note that the expected area covered by our points with label 1 is:

$$E[\mathbf{1}_{f(X,S_n)=1}] = \sum_{x \in S_n} E[I(x, S_n)\mathbf{1}_{L^n(x)=1}]$$

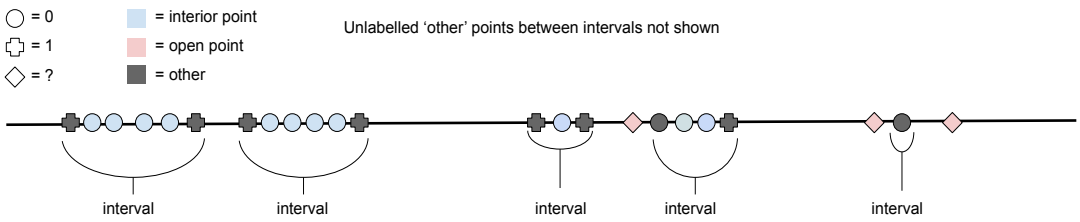

Figure 3.1: Visualization of intervals for 1-nn

.

The coverage of all interior points is $\leq \frac{n}{m_n+1} \to 0$. And we show that the coverage of each open point's labelled neighbor (which has label 0) also $\to 0$. Thus the area covered by points with label 1 goes to 1, and so the risk goes to $1 - \eta$ as the resulting estimator approaches $1 - f^*(x)$.

3.5.2.2 Formal proof

The structure of the proof will be based around corollary 3.5.1 and corollary 3.5.3. Since all points with label 0 are either interior or adjacent to open points, we just need to control the coverage of these two types of points. First we will bound the expected coverage of n interior points and see that it goes to 0. Next we will show that with high probability the number of open points will eventually be bounded. Finally we will show that each point adjacent to an open point has coverage going to 0.

Since $\frac{n}{m_n} \to 0$ the coverage of all interior points decreases faster than the number of interior points can grow.

Proposition 3.5.2. *If $X_{(i)}$ is an interior point, then the expected area covered by that point is $E[I(X_{(i)}, S_n)] = \frac{1}{m_n+1}$.*

Corollary 3.5.1. *The expected area covered by all interior points approaches 0 as $n \to \infty$.*

Now we want to show that asymptotically the probability of there being many open points at time n, when we stop sampling, is small. Let O_n be the number of open points at time n and let U_i be the change in the number of open points at time i So $O_n = \sum_{i=1}^{n} U_i$ and by construction $O_n \geq 0 \; \forall \; n$. Since the behaviour of U_i is different depending on whether O_{i-1} is 0 or not, we analyze U_i by analyzing it's behaviour between times when it returns to 0. We will call these returns to 0 *cycles*. Let τ_j be the j^{th} time that $O_i = 0$, with $\tau_1 = 0$ (since with no labelled points we have no open points). We first want to show that

$\tau_j < \infty \; \forall \; j$ with probability 1, that is that our number of open points returns to 0 infinitely often with probability 1.

To do this we will bound U_i by an 'idealized' process U_i'. This bound will only hold between cycles (since U_i has different behaviour when the number of open points is 0).

$$U_i' = \begin{cases} 2 & \text{if } Z_i = 1 \text{ and } Y_i = 0 \\ -1 & \text{if } Z_i = 0 \text{ and } Y_i = 1 \\ 0 & \text{otherwise} \end{cases}$$

Proposition 3.5.3. *If $O_{i-1} \neq 0$ then $U_i \leq U_i'$ $a.s.$*

Note that for i sufficiently large $E[U_i'] < 0$ and so $\sum_{i=i_0}^{\infty} U_i' \overset{a.s.}{\to} -\infty$. Thus the number of open points will always return to 0 in a finite number of iterations (with probability 1).

Proposition 3.5.4. $P(O_i = 0 \; i.o.) = 1$.

So we know we return to 0 open points infinitely often with probability 1. We want to show that the probability of having a large number of open points any time during cycle j_0 goes to zero as $j_0 \to \infty$.

Proposition 3.5.5. *Let $\tilde{T}_{1,i_0}$ be the first time after i_0 that $\sum_{i=i_0+1}^{\tilde{T}_{1,i_0}} Z_i = 1$ and let $T_{1,i_0} = \tilde{T}_{1,i_0} - i_0$. Let $\tilde{T}_{2,i_0}$ be the first time after i_0 that $\sum_{i=i_0+1}^{\tilde{T}_{2,i_0}} Y_i = 1$ and let $T_{2,i_0} = \tilde{T}_{2,i_0} - i_0$. Then:*

i $P(T_{1,i_0} < T_{2,i_0}) \leq p_{i_0} \frac{1}{\eta}$

ii $P(T_{1,i_0} = T_{2,i_0}) \leq p_{i_0}$

The first result can be generalized to find the probability of getting $\sum Z_{i_0+t} = a$ before $\sum Y_{i_0+t} = b$. Since the Z_i and Y_i are all independent, these can be calculated recursively.

Corollary 3.5.2. *Let $\tilde{T}_{1,i_0}^{(a)}$ be the first time after i_0 that $\sum_{i=i_0+1}^{\tilde{T}_{1,i_0}^{(a)}} Z_i = a$ and let $T_{1,i_0}^{(a)} = \tilde{T}_{1,i_0}^{(a)} - i_0$. Let $\tilde{T}_{2,i_0}^{(b)}$ be the first time after i_0 that $\sum_{i=i_0+1}^{\tilde{T}_{2,i_0}^{(b)}} Y_i = b$ and let $T_{2,i_0}^{(a)} = \tilde{T}_{2,i_0}^{(a)} - i_0$. If we denote $p_{i_0}^{(a,b)} = P(T_{1,i_0}^{(a)} < T_{2,i_0}^{(b)})$. Then we have the following recursive relationship:*

$$
\begin{aligned}
p_{i_0}^{(1,b)} &\leq p_{i_0}^{(1,1)} + (1 - p_{i_0}^{(1,1)}) p_{i_0}^{(1,b-1)} \leq b p_{i_0}^{(1,1)} \\
p_{i_0}^{(a,1)} &\leq p_{i_0}^{(1,1)} p_{i_0}^{(a-1,1)} \leq (p_{i_0}^{(1,1)})^a \\
p_{i_0}^{(a,b)} &= p_{i_0}^{(1,1)} p_{i_0}^{(a-1,b)} + P(T_{1,i_0} = T_{2,i_0}) p_{i_0}^{(a-1,b-1)} \\
&\quad + (1 - p_{i_0}^{(1,1)} - P(T_{1,i_0} = T_{2,i_0})) p_{i_0}^{(a,b-1)} \\
&\leq p_{i_0}^{(1,1)} p_{i_0}^{(a-1,b)} + \eta p_{i_0}^{(1,1)} p_{i_0}^{(a-1,b-1)} + p_{i_0}^{(a,b-1)}
\end{aligned}
$$

In particular we have that $p_{i_0}^{(a,b)} \leq 3^{a+b} (p_{i_0}^{(1,1)})^a$.

This shows that the probability of increasing beyond 4 open points before dropping back down to 0 open points $p_{i_0}^{(2,4)}$ is decreasing to 0.

Lemma 3.5.1. $P(O_n > 4) \to 0$ *as* $n \to \infty$.

We already know that points with label 0 which are not adjacent to open points are interior points. So we just need to show the contribution from the (up to 4) non-interior points with label 0 is shrinking to 0. We will do this by showing that the maximal distance between two intervals goes to 0.

Proposition 3.5.6. *Let* d_t *be the maximum of all distances between consecutive intervals at time* t. *Then* $d_n \overset{a.s.}{\to} 0$.

Corollary 3.5.3. *The coverage of labelled points adjacent to open points* $\overset{a.s.}{\to} 0$.

With corollaries 3.5.1 and 3.5.3 we can now prove Theorem 3.5.1.

Theorem 3.5.1. *Let* $X \overset{iid}{\sim} U(0,1)$ *and let* $f(x) = \eta,\ 0 < \eta < \frac{1}{2}$. *We sample* S_n *using augmented Algorithm 4, and with base active learning algorithm* $A^\dagger$ *described in Algorithm* 5. *If our estimator* $f(x, S_n)$ *is 1-*NN *then* $E\left[\mathbf{1}_{f(X,S_n)=Y}\right] \to 1 - \eta$.

Remark. It is clear that a similar result for regression (with squared loss) could be obtained using the same idea, with $f^*(X) = E[Y|X] = 0, Y_i = \epsilon_i$ (where ϵ_i is our *iid* $E[\epsilon] = 0$ noise) by using the above algorithm. Let a point have a pseudo-label of 0 if $|Y| \leq c \in |supp(\epsilon)|$ and 1 otherwise, and run the above algorithm on the pseudo-labels. You would again get intervals of low value points enclosed by high value points and could get MSE $\geq c^2$.

As stated earlier, this counterexample persists even if you require $k_n \to \infty$ and only stipulate that $\frac{k_n}{s_n} \to 0$, which is required by Condition 1 (and which gives consistency if our data is sampled passively). Although the result for the $k_n \to \infty$ case is slightly less general, and the definitions and techniques are more complex, the main idea behind the proof is the same, and the proof can be found in the appendix.

Theorem 3.5.2. *Let* $X \overset{iid}{\sim} U(0,1)$ *and let* $f(x) = \eta,\ 0 < \eta < \frac{1}{2}$ *and fix* $\epsilon > 0$. *We create our labelled training set* S_n *using augmented Algorithm 4, with* $P(Z_i = 1) = \frac{1}{i}$, *and with base active learning algorithm* $A^\dagger$ *described in Algorithm* 5. *If our estimator is* k*-NN then* $\exists\ \{k_n\}_{n=1}^{\infty}$ *which satisfies Condition 1 and* $\liminf E\left[\mathbf{1}_{f_n(x,S_n)=Y}\right] \geq 1 - \eta - \epsilon$.

3.6 Sufficiency for bounded support estimators

We now aim to extract the properties of the histogram estimator which make it immune to the type of attack used in the nearest neighbor counterexample. Our conditions will assume that the weight functions $W_{ni}(x, S_n(X))$ take a simplified form, where which training points have non-zero weight only depends on x, X_i and n. Similar to Condition 1, these conditions will be complex to state mathematically, but will have interpretable effects.

Condition 2.

1. $W_{ni}(x, S_n(X)) = \frac{w_n(x,X_i)}{\sum_j w_n(x,X_j)}$.

2. if $supp_n(x) = \{y \in \mathcal{X} : w_n(x,y) > 0\}$ then $diam(supp_n(x)) \to 0$.

3. $w_n(x,y) \leq K\ \forall\ n, x, y$.

4. $\sum_{i=1}^{n} w_n(X, X_i) Z_i \xrightarrow{\mathrm{P}} \infty \quad (X, X_i \sim P_X)$.

By enforcing this structure on $W_{ni}(x)$, we allow the unnormalized weight of each point to depend only on the location of the training point X_i and the test point x, preventing the relative weight of a point from being affected after the label has been observed. By forcing the support to shrink in size we ensure that the method is sufficiently local (Zakai and Ritov, 2009). Finally by bounding the maximum relative weight of any single point and requiring that the relative weights of our randomly sampled points is unbounded (in probability), we ensure that no finite amount of actively sampled data can overwhelm our passively sampled data. Although this generalization only includes certain partition estimators and bounded support regular kernel estimators who's kernel function is also bounded away from 0 on their support, the proof itself illuminating.

Theorem 3.6.1. *Assume our classifier and augmented algorithm satisfy Condition 2. Then under Regularity Condition 2 our estimator is consistent for any active learning algorithm* A.

3.7 Conclusions and further directions

We have seen that in the noiseless setting under mild conditions the classical weighted averaging estimators, specifically those based on partitions, smoothing kernels and nearest neighbors, are consistent with a small amount of data sampled randomly. However once even a little noise is introduced there is a bifurcation, where some estimators such as the histogram retain this consistency while others such as k-nn can be made highly inconsistent even if they are consistent in the noiseless case. The structure of the counterexample in Section 3.5 and the Condition in Section 3.6 suggests this divergence stems from how dramatically the relative weight of a data point can be affected after its label has been observed, and how few data points determine the final prediction. This explains why both adversarial sampling and label noise were needed to highlight the differences in behaviour. As seen in the 1-NN counterexample (the structure of which can also give counterexamples for unbounded kernel estimators with sufficiently quickly shrinking h_n), if the influence of one data point can be too easily manipulated (after observing it's label) by the placement of other data points, we can get inconsistency even with our randomly sampled data. Condition 2 strongly protect against this, and less strenuous conditions can likely be found for local averaging estimators. However more interestingly the intuition behind these properties may provide guidance when using more modern estimators, and exploring and formalizing this is the subject of future work.

One direction would be to explore whether this disjunction in the vulnerability of different estimators is mirrored for more advanced methods. Under passive sampling SVMs and Random Forests are both competitive classifiers (Caruana and Niculescu-Mizil, 2006), but given the similarities between SVM and Nearest Neighbors, and Random Forests and Histograms, their guarantees may be very different under active sampling. Another potential avenue would be finding ways to adapt complex methods to maintain consistency under Algorithm 4 or similar schemes. For example the soft-margin SVM dual form optimization variables α_i encode the influence of a data point on the prediction of nearby points. The high level ideas in Condition 2 suggest additional constraints (such as $\max_i \alpha_i - B_n \sum \alpha_i \leq 0, B_n \to 0$) may result in a version of the SVM which is more robust under a similar augmented active learning algorithm.

Orthogonally it would be interesting to see how these Conditions change if we put constraints on the underlying active learning algorithm being augmented. The counter examples explored were extreme, and it is unlikely that adversarial behaviour would be induced by most active learning algorithms used in practice. It is an interesting open question whether constraints can be place on the underlying active learning algorithm in Algorithm

4 which are not too constricting, but which allow for an extension of Theorem 3.4.1 to the noisy case.

3.8 Appendix A: Counterexample for nearest neighbors with $k_n \to \infty$

In order to more accurately mirror the consistency conditions under passive sampling we now add the requirement that $k_n \to \infty$.

Property 1. $k_n \to \infty, \frac{k_n}{s_n} \to 0$

Our counterexample will be similar to in the 1-NN case, but we will work with $p_i = \frac{1}{i}$ instead of a generic p_i. The only difference will be in the definition of an open point, which will need to be generalized to depend on k_n. If $k_n = k$ then an open point will be an unlabelled point with at least one labelled neighbour, and without $\lfloor \frac{k}{2} \rfloor + 1 = k'$ 1 labels in a row to the left or right.

Definition 3.8.1. A point $X_{(i)}$ is an *open point* at time t if $L^t(X_{(i)}) = ?, L^t(X_{(i+1)}) \in \{0, 1\}$ and $(L^t(X_{(i+1)}, ..., L^t(X_{(i+k')}) \neq \mathbf{1}_{k'}$ or $L^t(X_{(i)}) = ?, L^t(X_{(i-1)}) \in \{0, 1\}$ and $(L^t(X_{(i-1)}, ..., L^t(X_{(i-k')}) \neq \mathbf{1}_{k'}$, where $\mathbf{1}_{k'}$ is a k'-vector of 1's.

Note that when $k = 1$ this is the same as our previous definition, and that intervals will have the same effect as before, where two consecutive intervals without any open points between them will cause any test points between them to be predicted 1. Similarly we will extend our definition of coverage to be the area where a set of size k are the k closest labelled points.

Definition 3.8.2. Let $C_k(x, S_n) = \underset{C \subset S_n, |C|=k}{\arg\min} \sum_{x' \in C} |x' - x|$. Then the *k-coverage* of a set C is $I_k(C, S_n) = \int \mathbf{1}_{C = C_k(z, S_n)}\, dz$

Note that the only sets with non-zero coverage are sets of consecutive (within S_n) labelled points. This again partitions the real line and we get a decomposition of our expected coverage by 1.

$$E[\mathbf{1}_{f_n(X,S_n)=1}] = \sum_{C \in \mathcal{C}_k} E\Big[I_k(C, S_n)\mathbf{1}_{\sum_{x \in C} L^n(x) \geq k'}\Big]$$

(where $1+? = 1, 0+? = 0$)

$\mathcal{C}_k = \{C \subset S_n : C = \{X_{(i)}, ..., X_{(i+k)}\}\}$

Where this ordering is only over $X \in S_n$

For each k fixed the proof will follow largely the same structure; although getting k' 1's in a row is a much lower probability event than just getting a single 1, the probability is

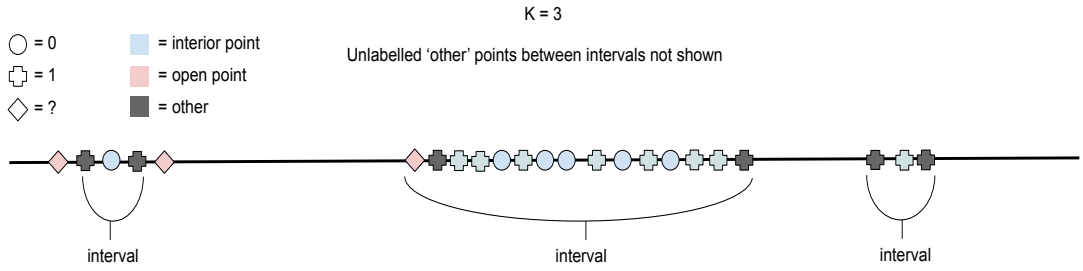

Figure 3.2: Visualization of intervals for 3-nn

still constant (for fixed k), where as the probability of sampling randomly is shrinking, and so eventually the number of open points will be small.

Our strategy will be very similar to in the 1-NN case, which was to show that the expected area covered by point with label 1 $E\left[\mathbf{1}_{f_n(X,S_n)=1}\right] \to 1$, as this gives us a risk of $1-\eta$.

We again use U_i to denote the change in the number of open points, and will again use an idealized version U_i' which dominates U_i to simplify analysis.

$$U_i' = \begin{cases} 2 & \text{if } Z_i = 1 \text{ and } Y_i = 0 \\ -1 & \text{if } Z_j = 0 \,\forall\, j \in \{i, ..., i-k'\} \text{ and} \\ & Y_j = 1 \,\forall\, j \in \{i, ..., i-k'\} \text{ and} \\ & Y_{i-k'-1} = 0 \\ 0 & \text{otherwise} \end{cases}$$

The following propositions all have the same proofs as in the 1-NN case, since $P(U_i' = 2) \to 0$ and $P(U_i' = -1) \to \eta^{k'}(1-\eta)$. Our U_i' are no longer independent, but they do have finite range independence, and so we still have a SLLN for them.

Proposition 3.8.1. *If $O_l \neq 0$ for $l \in \{i-k'-2, ..., i-1\}$ then $U_i \leq U_i'$ a.s.*

Proposition 3.8.2. $P(O_i = 0 \; i.o.) = 1$.

Now we will get the equivalent to proposition 3.5.5.

Proposition 3.8.3. *Assume $P(Z_i = 1) = \frac{1}{i}$, $P(Y_i = 1) = \eta$ and k fixed. Let $\tilde{T}_{1,i_0}$ be the first time after i_0 that $\sum_{i=i_0+1}^{\tilde{T}_{1,i_0}} Z_i = 1$ and let $T_{1,i_0} = \tilde{T}_{1,i_0} - i_0$. Let $\tilde{T}_{2,i_0}$ be the first time after i_0 that $\sum_{i=i_0+\tilde{T}_{2,i_0}-k'}^{\tilde{T}_{2,i_0}} Y_i = k'$ and let $T_{2,i_0} = \tilde{T}_{2,i_0} - i_0$. Then:*

i $P(T_{1,i_0} < T_{2,i_0}) \leq \frac{c}{2\sqrt{i_0}}\left[\frac{1}{1-\eta}\left(\frac{1}{\eta^{k'}} - 1\right)\right]$

ii $P(T_{1,i_0} = T_{2,i_0}) \leq \frac{1}{i_0}$

And if we generalize we have the same recursive relationships.

Corollary 3.8.1. *Let* $\tilde{T}^{(a)}_{1,i_0}$ *be the first time after* i_0 *that* $\sum_{i=i_0+1}^{\tilde{T}^{(a)}_{1,i_0}} Z_i = a$ *and let* $T^{(a)}_{1,i_0} = \tilde{T}^{(a)}_{1,i_0} - i_0$. *Let* $\tilde{T}^{(b)}_{2,i_0}$ *be the first time after* i_0 *that we've had* k' *out of the last* k *queries be error terms on* b *disjoint occasions (so starting over each time) and let* $T^{(a)}_{2,i_0} = \tilde{T}^{(a)}_{2,i_0} - i_0$. *If we denote* $p^{(a,b)}_{i_0} = P(T^{(a)}_{1,i_0} < T^{(b)}_{2,i_0})$. *Then we have the following recursive relationship:*

$$
\begin{aligned}
p^{(1,b)}_{i_0} &\leq p^{(1,1)}_{i_0} + (1 - p^{(1,1)}_{i_0})p^{(1,b-1)}_{i_0} \leq b p^{(1,1)}_{i_0} \\
p^{(a,1)}_{i_0} &\leq p^{(1,1)}_{i_0} p^{(a-1,1)}_{i_0} \leq (p^{(1,1)}_{i_0})^a \\
p^{(a,b)}_{i_0} &\leq p^{(1,1)}_{i_0} p^{(a-1,b)}_{i_0} + \frac{1}{i_0} p^{(a-1,b-1)}_{i_0} + p^{(a,b-1)}_{i_0}
\end{aligned}
$$

In particular we have that $p^{(a,b)}_{i_0} \leq 3^{a+b}(p^{(1,1)}_{i_0})^a$

Now we have a (slightly stronger) equivalent to lemma 3.5.1.

Lemma 3.8.1. *For any* k, $P(O_n > 6 \; i.o.) = 0$.

This means $\mathbf{1}_{O_n>6} \overset{a.s.}{\to} 0$. Therefore by an equivalent definition of almost sure convergence (Chung, 2001) $\exists\, n_{k,\epsilon}\; s.t.\; P(\mathbf{1}_{O_n>6} \neq 0\; \forall n \geq n_{k,\epsilon}) \leq \epsilon$. Of course we have no way of knowing what $n_{k,\epsilon}$ is for each values of k, ϵ, but we know they exist. Therefore we will allow k_n to increase in the following manner (which we denote $k(n, \epsilon)$):

- $k_n = 1$ for $n < n_{2,\epsilon}$
- $k_n = 2$ for $n \in [n_{2,\epsilon}, n_{3,\epsilon}]$
- ...
- $k_n = k$ for $n \in [n_{k,\epsilon}, n_{k+1,\epsilon}]$

Of course we also need to satisfy $\frac{k_n}{s_n} \to 0$ and so we can just take $k_n = \min(k(n, \epsilon), \log\log(n))$. The rest of the proof follows as in the 1-NN case.

Proof of theorem 3.5.2. Let the sequence $\{k_n\}_{n=1}^{\infty}$ be as described above. Then for $n \geq n_2$ we know that when we have finished taking our n samples, $P(\mathbf{1}_{O_n>6} \neq 0\; \forall n \geq n_{k,\epsilon}) \leq \epsilon$. We therefore split our expected coverage with 1 into

$E[\mathbf{1}_{f_n(X,S_n)=1}] = E[\mathbf{1}_{f_n(X,S_n)=1}\mathbf{1}_{O_n>6}] + E[\mathbf{1}_{f_n(X,S_n)=1}\mathbf{1}_{O_n\leq 6}]$.

Trivially $E[\mathbf{1}_{f_n(X,S_n)=1}\mathbf{1}_{O_n>6}] \geq 0$. So we focus on $E[\mathbf{1}_{f_n(X,S_n)=1}\mathbf{1}_{O_n\leq 6}]$.

Again all area which may be predicted as 0 are on the interior of intervals, or are covered by the points next to open points. The expected k-coverage of a set $C \in \mathcal{C}_{k_n}$ of all interior points is again $\frac{1}{m_n+1}$ and there are fewer than n such sets. This leaves up to 6 C that are not all interior points, and which could have $\sum_{x\in C} L^n(x) < k'$. These are all on the edges of intervals, and the lengths between intervals are still approaching 0 with probability 1 by proposition 3.5.6. Thus we have $E[\mathbf{1}_{f_n(X,S_n)=1}\mathbf{1}_{O_n\leq 6}] \to 1-\epsilon$, and so $E[\mathbf{1}_{f_n(X,S_n)=1}] \geq 1-\epsilon$, which gives us $E[\mathbf{1}_{f_n(X,S_n)\neq Y}] \geq (1-2\eta)(1-\epsilon)+\eta \geq 1-\eta-\epsilon$. □

3.9 Appendix B: Proofs

Algorithm 6: Augmented Algorithm for query synthesis

Input: Active learning algorithm A, number of samples n, probability sequence $(p_1, ..., p_n)$, underlying marginal distribution P_X

Output: Labelled data set S_n

$S_0 = \emptyset$;

for i *from* 1 *to* n **do**

  Draw an independent Bernoulli random variable Z_i with $P(Z_i = 1) = p_i$;

  if $Z_i = 1$ **then**

    Draw X_i from P_X

  else

    Select X_i according to $A(S_{i-1})$

  end

  Query selected point and receive Y_i ;

  $S_i = S_{i-1} \cup (X_i, Y_i)$;

end

3.9.1 Sufficiency in the noise free case

Why is Condition 1 our requirement? Fix $X = x$ and let ϕ be the distribution on $(X_1, ..., X_n)$ induced by our augmented AL algorithm. By the definition of E_i we have that $\mathbf{1}_{E_i} = 1 \implies Z_i = 1$ and so:

$$\begin{aligned}&\underset{Z_i\sim B(p_i)}{E}\Big(\underset{X\sim\phi}{E}\big[g_n(x, r, \{X_i\}^n, \{\mathbf{1}_{E_i}\}^n)|\{Z_i\}^n\big]\Big)\\ &= \underset{Z_i\sim B(p_i)}{E}\underset{X_i\sim\mu}{E}\big[g_n(x, r, \{X_i\}^n, \{\mathbf{1}_{E_i}\}^n)\big]\end{aligned}$$

And from this and the definition of g_n we have that $\forall\, k$:

$$\underset{Z_i\sim B(p_i)}{E}\underset{X\sim\phi}{E}\big[\sum_{i=1}^{n} W_{ni}(x,\{X_i\}^n)\mathbf{1}_{X_i\in B_{x,r}}|\sum \mathbf{1}_{E_i}=k\big]$$
$$\geq \underset{Z_i\sim B(p_i)}{E}\underset{X_i\sim\mu}{E}\big[g_n(x,r,\{X_i\}^n,\{\mathbf{1}_{E_i}\}^n)|\sum \mathbf{1}_{E_i}=k\big]$$

Proof of Theorem 3.4.1.
We want to show that $\int E_{S_n}\big[(f_n(x,S_n)-f(x))^2\big]P_X(dx)\to 0$.

$$\int E_{S_n}\big[(f_n(x,S_n)-f(x))^2\big]P_X(dx)=\int E_{S_n}\big[(\sum W_{ni}(x)f(x_i)-f(x))^2\big]P_X(dx)$$
$$\leq 2\int E_{S_n}\big[(\sum W_{ni}(x)f(x_i)-\sum W_{ni}(x)f(x))^2\big]P_X(dx)$$
$$+2\int E_{S_n}\big[(f(x)[\sum W_{ni}(x)-1])^2\big]P_X(dx)$$

We will work on bounding the first term since the second term trivially goes to 0 due to Condition 1.

$$\int E_{S_n}\big[(\sum W_{ni}(x)f(x_i)-\sum W_{ni}(x)f(x))^2\big]P_X(dx)$$
$$\leq \int E_{S_n}\big[\sum W_{ni}(x)(f(x_i)-f(x))^2\big]P_X(dx)$$
Define $\mathcal{X}^{(\delta)}=\{x:|f(B_{x,\delta})|=1\},\quad \delta_\epsilon=\sup\delta\ s.t.\ P_X(\mathcal{X}^{(\delta)})\geq 1-\epsilon.$
$$\leq \int_{\mathcal{X}^{(\delta_\epsilon)}} E_{S_n}\big[\sum W_{ni}(x)(f(x_i)-f(x))^2\big]P_X(dx)+\epsilon$$
Let $S_n=S_n^{(a)}\cup S_n^{(r)}$ where the first is actively selected data and the second is the randomly selected.
$|B_{x,\delta_\epsilon}\cap S_n^{(r)}|\to P_X(B_{x,\delta_\epsilon})\sum^n p_i.$
For each $x\ \exists\, n_0(x)\ s.t.\ P_X(B_{x,\delta_\epsilon})\sum^n p_i\geq H_n\ \forall n\geq n_0(x).$
$\exists n_0\ s.t.\ P_X(\{x:P_X(B_{x,\delta_\epsilon})\sum^n p_i\leq H_n\})\leq\epsilon.$

Let n be sufficiently large and denote the intersection of the complement of the above set with $\mathcal{X}^{(\delta_\epsilon)}$ by $\tilde{\mathcal{X}}$.

$$\leq \int_{\tilde{\mathcal{X}}} E_{S_n}\big[\sum W_{ni}(x)(f(x_i)-f(x))^2\big]P_X(dx)+2\epsilon.$$

Let F_n be the event that $|B_{x,\delta_\epsilon}\cap S_n^{(r)}|\geq H_n$.

$$= \int_{\tilde{\mathcal{X}}} P(F_n)E_{S_n}\big[\sum W_{ni}(x)(f(x_i)-f(x))^2|F_n\big]$$
$$+ P(F_n^c)E_{S_n}\big[\sum W_{ni}(x)(f(x_i)-f(x))^2|F_n^c\big]P_X(dx)+2\epsilon$$
$$\leq \int_{\tilde{\mathcal{X}}} E_{S_n}\big[\sum W_{ni}(x)(f(x_i)-f(x))^2|F_n\big]P_X(dx)+3\epsilon$$

For $n\geq n_1$ since $P_X(B_{x,\delta_\epsilon})$ bounded away from 0).

$$\leq \int_{\tilde{\mathcal{X}}} E_{S_n^{(r)}}\big[\sup_{S_n^{(a)}}\sum W_{ni}(x)\mathbf{1}_{||X_i-x||\geq\delta_\epsilon}|F_n\big]P_X(dx)+3\epsilon$$
$$\to 3\epsilon.$$

□

Proof of Proposition 3.4.1. In the proof of part i) we will actually prove that the condition is if-and-only-if since this will be needed in section 4.

Let $N_n^{(R)}=\sum Z_i$ be the number of labelled points selected randomly. Let $A_n(x)$ denote the cell containing the point x and let $N_n(x)=\sum \mathbf{1}_{X_i\in A_n(x)}$ be the number of labelled points in the same cell as x, and let $N_n^{(R)}(x)=\sum \mathbf{1}_{X_i\in A_n(x)}Z_i$ be the number of labelled points in the same cell as x which were selected randomly.

We first prove the forward direction by showing we satisfy Condition 1. Let $H_n=\left\lfloor\frac{\sqrt{s_n}}{\sqrt{h_n^d}}\right\rfloor$, noting that $\frac{H_n}{s_n}=\frac{1}{\sqrt{h_n^d s_n}}\to 0$ and $H_n h_n^d=\sqrt{h_n s_n}\to\infty$. Since $h_n\to 0$, for any $r>0$ eventually the entire cell a data point is in will be within r of the point. Then repeat the proof of Theorem 6.2 in Devroye et al. (2013), replacing n with H_n, to show that $P(N_n^{(R)}(X)\leq M)\to 0\,\forall\, M<\infty$. This completes the proof since a non-empty histogram has $\sum W_{ni}(x)=1$, and for n sufficiently large all the training points with non-zero weight will be within r.

If $h_n\not\to 0$ then clearly the Condition cannot hold for r sufficiently small as the ball $B_{x,r}$ can be made arbitrarily small compared to the minimum size of the cell.

If $h_n^d s_n\to 0$, then the number of cells is growing at a faster rate than the number of randomly sampled data points, and if our active algorithm just samples the nearest neighbor to the point last sampled, then the majority of cells would end up with no data and would thus have $\sum W_{ni}(x)=0$.

This leaves us with the case where $h_n^d s_n\in[\alpha_1,\alpha_2],\ 0<\alpha_1\leq\alpha_2<\infty$. We can study this using the theory of Random Allocations Kolchin et al. (1978), which characterizes the properties of counts of urns with k balls after n balls are placed iid into urns. If we have a

uniform distribution on $\mathcal{X}$ then we are in the Central Domain with equiprobable allocation, and from Theorem 1 (p.18) of Kolchin et al. (1978), we have that for any $\epsilon > 0$, for n sufficiently large $P(N_n^{(R)}(X) = 0) \geq e^{-h_n^d(1+\epsilon)s_n}$ almost surely. This is because the number of cells with no randomly sampled points is normally distributed around $\frac{1}{h_n}e^{-h_n^d(1+\epsilon)s_n}$ with variance that is $O(\frac{1}{h_n})$. Thus as above satisfying the Condition is impossible if, for example, our active algorithm just samples the nearest neighbor to the point last sampled.

For part $ii)$ Condition 1 is satisfied with $H_n = k_n$ as long as for any fixed $r > 0$ and any x, random sampling puts more than k_n data points $B_{x,r}$, and this is proved in Lemma 1 in Dasgupta (2012).

□

3.9.2 Examples in the noisy case

Proof of Proposition 3.5.1. By Regularity Condition 2, for n large enough all but an $\epsilon > 0$ P_X-measure of cells will be such that $f^*(x_1) = f^*(x_2) \; \forall \; x_1, x_2 \in A_{nj}$, where A_{nj} is an arbitrary cell in our histogram. Therefore we need to show that $P(N_n^{(R)}(X) \leq M) \to 0 \forall M$. But if we fix $N_n^{(R)}$, this is exactly the result in Theorem 6.2 of Devroye et al. (2013), with n replaced by $N_n^{(R)}$. And by Levy's extension to Borel-Cantelli Williams (1991) we know that for any $\delta > 0$, for n sufficiently large $N_n^{(R)} \in [(1-\delta)s_n, (1+\delta)s_n]$ with probability 1. Thus with probability 1 we have that $h_n^d N_n^{(R)} \to \infty$, so the conditions of Theorem 6.2 in Devroye et al. (2013) hold with probability 1, ensuring that $P(N_n^{(R)}(X) \leq M) \to 0 \; \forall \; M$ thereby completing the proof.

□

3.9.3 Nearest Neighbor counterexample

Proof of proposition 3.5.2. Since $X_{(i)}$ is an interior point, both of these neighbors are labelled, and so $X_{(i)}$ will only be the closest point on an area half of the distance between its neighbors on either side. Since the $X_i \sim U(0,1)$ the expected distance between $X_{(i)}$ and its neighbor on either side is $\frac{1}{m_n+1}$. Therefore the expected coverage is $\frac{1}{m_n+1}$. □

Proof of corollary 3.5.1. Each interior point covers $\frac{1}{m_n+1}$ and the number of interior points is trivially bounded by n, and by our assumptions $\frac{n}{m_n+1} \to 0$. □

Proof of proposition 3.5.3. Note that the only way to increase the number of open points is to query a point which is not open, and for that point to have label 0. In this case we increase the number of open points by at most 2. This is the event $\{Z_i = 1, Y_i = 0\}$. Conversely if we query an open point and it's label is 1 then we decrease the number of

open points by at least 1. This is the event $\{Z_i = 0, Y_i = 1\}$. And even when neither of these happens the number of open points can still decrease, but cannot increase. Thus we have that $U_i' = \max\{supp(U_i|Y_i = y, Z_i = z)\}$. □

Proof of proposition 3.5.4. We will prove by induction that $\tau_j < \infty \; \forall \; j$ with probability 1. Note for our base case that $\tau_1 = 0 < \infty$. Now assume $\tau_{j-1} = i_0 < \infty$. If $U_{i_0+1} = 0$ (which can happen if for example our new data point has label 1) then $\tau_j = i_0 + 1 < \infty$. Now assume $U_{i_0+1} > 0$. Thus we know that $O_{i_0+1} > 0$, and will remain above 0 until τ_j giving us that $\sum_{i=i_0+2}^{\tau_j} U_i \leq \sum_{i=i_0+2}^{\tau_j} U_i'$. But for some i_1 $E[U_i'] < 0 \; \forall \; i \geq i_1$. Therefore by SLLN $\sum_{i=i_0}^{\infty} U_i' \overset{a.s.}{\to} -\infty$ and so there exists some $T < \infty$ s.t. $\sum_{i=i_0}^{\infty} U_i' \leq 0$ with probability 1. Therefore $\tau_j \leq T < \infty$ with probability 1, and so $P(O_i = 0 \; i.o.) = 1$. □

Proof of proposition 3.5.5. i

$$P(T_{1,i_0} < T_{2,i_0}) = \sum_{t=1}^{\infty} P(T_{1,i_0} = t) P(T_{1,i_0} < T_{2,i_0} | T_{1,i_0} = t)$$
$$= \sum_{t=1}^{\infty} p_{i_0+t} \prod_{j=1}^{t-1} (1 - p_{i_0+j})[(1-\eta)^{t-1}] \leq p_{i_0} \sum_{t=1}^{\infty} [(1-\eta)^{t-1}] = p_{i_0} \frac{1}{\eta}$$

ii

$$P(T_{1,i_0} = T_{2,i_0}) = \sum_{t=1}^{\infty} P(T_{1,i_0} = t) P(T_{2,i_0} = t)$$
$$= \sum_{t=1}^{\infty} p_{i_0+t} \prod_{j=1}^{t-1} (1 - p_{i_0+j})[P(T_{2,i_0} = i_0 + t)] \leq p_{i_0} \sum_{t=1}^{\infty} [P(T_{2,i_0} = i_0 + t)] = p_{i_0}$$

□

Proof of corollary 3.5.2. The three inequality relationships come straight from the independence of our random variables Y_i, Z_i. The final statement can be shown by induction. It is clearly true for the case $a = 1, b = 1$. Assume true for all $a \leq a_0 - 1, b \leq b_0$.

$$\begin{aligned} p_{i_0}^{(a_0,b_0)} \leq & p_{i_0}^{(1,1)} \times 3^{a_0-1+b_0} (p_{i_0}^{(1,1)})^{a_0-1} + \eta p_{i_0}^{(1,1)} \times 3^{a_0-1+b_0-1} (p_{i_0}^{(1,1)})^{a_0-1} + \\ & 3^{a_0+b_0-1} (p_{i_0}^{(1,1)})^{a_0} \\ \leq & 3^{a_0+b_0} (p_{i_0}^{(1,1)})^{a_0} \end{aligned}$$

And finally note that in the above there is symmetry between the roles of a and b so the same calculations show that if it's true for all $a \leq a_0, b \leq b_0 - 1$ then it's true for $a \leq a_0, b \leq b_0$.

□

Proof of lemma 3.5.1. Let E_j be the event that during the j^{th} cycle we have more than 4 open points. So if the j^{th} cycle starts at time τ_j then $E_j = \{\max_{\tau j \leq j \leq \tau_{j+1}} O_j > 4\}$. Also $\forall\, t \in [\tau j \leq j \leq \tau_{j+1}], \{O_t > 4\} \subset E_j$. Note that $P(E_j) \leq p_{\tau_j}^{(2,4)} \leq cp_{\tau_j}^2 \leq cp_{j-1}^2$ since each cycle must have length at least 1. Thus if we hit n during the j^{th} cycle then $P(O_n > 4) \leq cp_j^2 \to 0$ as $j \to \infty$. And by proposition 3.5.4 we have that if j_0 is the cycle we are in at time n then $j_0 \to \infty\ a.s.$ as $n \to \infty$. Thus $P(E_{j_0}) \to 0$. □

Proof of proposition 3.5.6. Fix $\epsilon > 0$ and $\delta < \frac{\epsilon}{2}$. Define two events:

1. $\Omega_1 = \{$we return to 0 infinitely often$\}$
2. $\Omega_2 = \{\forall\, i\, X_{(i+1)} - X_{(i)} \leq \delta$ and $X_{(1)}, 1 - X_{(n)} \leq \delta\}$

then $\{d_n > \epsilon\, \forall\, n\} \subset (\Omega_1 \cap \Omega_2)^c$. This is because returning to 0 infinitely often means that infinitely often we act according to $A^\dagger$ when the number of open points is 0. This action samples the unlabelled point which is furthest from any labelled point. We will show that just these actions are enough to prevent $d_n \geq \epsilon\, \forall\, n$ when (2) is also true. We will also ignore the fact that our labelled intervals take up length as this length is negligible and only forces the *empty interval* (the interval of consecutive unlabelled points) to be smaller.

By (2) if the empty interval containing the unlabelled point which is furthest from any labelled point is of size l then the point which is newly labelled must be within $\frac{\delta}{2}$ of the center of the interval, and so the maximum size of the two new empty intervals created is $\frac{l+\delta}{2}$. If $l \geq \epsilon$ then we get that the new empty intervals have length $\leq \frac{3}{4}l$, so we're guaranteed to produce empty intervals of length no more than $\frac{3}{4}$ of the original intervals length. Additionally since $\epsilon > 2\delta$ there are no empty intervals which cannot be cut to size smaller than ϵ due to there not being two consecutive points with distance greater than ϵ. So any interval of finite size $> \epsilon$ can be split into intervals all of size less than ϵ in a finite number of cuts. Thus if at any time t we have $N < \infty$ empty intervals of size $> \epsilon$ (which must be the case since the sum of our interval lengths is bounded by 1) they will all be reduced to intervals of size $< \epsilon$ in a finite number of cuts.

By proposition 3.5.4 $P(\Omega_1) = 1$. By Glivenko-Cantelli $P(\Omega_2) = 1$, since otherwise $\exists x\, s.t. F_n(x) = F_n(x+\delta) \forall n$, where $F_n(x)$ is the usual empirical cdf. But $F(x) \neq F(x+\delta)$ and so Glivenko-Cantelli would be violated, which happens with probability 0. Therefore with probability 1 we cannot have that $d_n > \epsilon\, \forall\, n$ and so $d_n \overset{a.s.}{\to} 0$. □

Proof of corollary 3.5.3. The coverage of each labelled point adjacent to an open point is half the distance to the next interval. However by 3.5.6 this distance $\overset{a.s.}{\to} 0$. □

Proof of theorem 3.5.1. Let $\mathcal{I}_n$ be the set of all interior points and let $\mathcal{A}_n$ be the set of all labelled points adjacent to open points.

$$E[\mathbf{1}_{f(X,S_n)=1}] = E\Big[\sum_{x\in S_n} I(x,S_n)\mathbf{1}_{L(x)=1}\Big] =$$
$$E\Big[\sum_{x\in\mathcal{I}_n} I(x,S_n)\mathbf{1}_{L(x)=1} + \sum_{x\in\mathcal{A}_n} I(x,S_n)\mathbf{1}_{L(x)=1} + \sum_{x\in S_n\setminus\mathcal{I}_n\cup\mathcal{A}_n} I(x,S_n)\mathbf{1}_{L(x)=1}\Big]$$

We know that all points with label 0 are either in $\mathcal{I}_n$ or $\mathcal{A}_n$. By corollary 3.5.1 we have that $E\sum\limits_{x\in\mathcal{I}_n} I(x,S_n) \to 0$, and by corollary 3.5.3 $E\sum\limits_{x\in\mathcal{A}_n} I(x,S_n) \to 0$. Thus since $E\sum\limits_{x\in S_n} I(x,S_n) = 1$ we have that $E\sum\limits_{x\in S_n\setminus\mathcal{I}_n\cup\mathcal{A}_n} I(x,S_n) \to 1$, and since $\mathbf{1}_{L(x)=1} = 1\ \forall\ x \in S_n \setminus \mathcal{I}_n \cup \mathcal{A}_n$ we have that $E[\mathbf{1}_{f(X,S_n)=1}] \to 1$, and so $E[\mathbf{1}_{f(X,S_n)\neq Y}] \to 1-\eta$. □

Proof of proposition 3.8.3. i

$$P(T_{1,i_0} < T_{2,i_0}) = \sum_{t=1}^{\infty} P(T_{1,i_0}=t)P(T_{1,i_0} < T_{2,i_0}|T_{1,i_0}=t)$$
$$P(T_{1,i_0}=t) \le \frac{1}{i_0+t}$$

By Markov

$$P(T_{2,i_0} > t) \le \frac{E[T_{2,i_0}]}{t+1} \le \frac{E[T_{2,i_0}]}{t}E[T_{2,i_0}] = \frac{1}{1-\eta}\big(\frac{1}{\eta^{k'}}-1\big)$$

By AM-GM inequality

$$P(T_{1,i_0} < T_{2,i_0}) \le \sum_{t=1}^{\infty}\frac{1}{i_0+t}\frac{1}{t}\frac{1}{1-\eta}\big(\frac{1}{\eta^{k'}}-1\big) \le \frac{1}{2\sqrt{i_0}}\frac{1}{1-\eta}\big(\frac{1}{\eta^{k'}}-1\big)\sum_{t=1}^{\infty}\frac{1}{t^{\frac{3}{2}}}$$
$$= \frac{c}{2\sqrt{i_0}}\frac{1}{1-\eta}\big(\frac{1}{\eta^{k'}}-1\big)$$

ii Proof is same as for proposition 3.5.5

□

Proof of lemma 3.8.1. Let E_j be the event that during the j^{th} cycle we have more than 6 open points. So if the j^{th} cycle starts at time τ_j then $E_j = \{\max\limits_{\tau j\le j\le\tau_{j+1}} O_k > 6\}$. Also $\forall\ t \in [\tau j \le j \le \tau_{j+1}], \{O_t > 6\} \subset E_j$. Note that $P(E_j) \le p_{\tau_j}^{(3,6)} \le (cp_{\tau_j})^3 \le (cp_{j-1})^3 = c^3\frac{1}{j^{\frac{3}{2}}}$. By proposition 3.5.4 we have that if j_0 is the cycle we are in at time n then $j_0 \to \infty\ a.s.$ as $n\to\infty$. And by Borel-Cantelli we have that $P(E_j\ i.o.) = 0$. □

3.9.4 Sufficiency for bounded support estimators

Proof of Theorem 3.6.1. For convenience of notation, we will let $Y_i \in \{1, -1\}$, using the usual transformation from our current $Y_i \in \{0, 1\}$ setting. Under this transformation, and by the assumptions on the structure of our $W_{ni}(x, S_n(X))$,

$$\begin{aligned} f_n(x, S_n) &= sign(\sum W_{ni}(x, S_n(X))Y_i) \\ &= sign(\sum w_n(x, X_i)Y_i) \end{aligned}$$

Therefore for consistency we want to show $P(f^*(X)f_n(X, S_n) = -1) \to 0$. For x fixed this occurs iff $\sum w_n(x, X_i)Y_i f^*(x) < 0$. Define $\gamma_n = \{x \in \mathcal{X} : \sup_{z \in supp_n(x)} |f(z) - f(x)| \leq \frac{|0.5 - f(x)|}{2}\}$. By the assumption that $diam(supp_n(x)) \to 0$ and Regularity Condition 2, $P_X(\gamma_n) \to 1$, and so for some $\epsilon > 0$, for n sufficiently large

$$P(f^*(X)f_n(X, S_n) = -1) \leq P(f^*(X)f_n(X, S_n) = -1 | X \in \gamma_n) + \epsilon.$$

Also define the following:

$$\begin{aligned} S_n(x) &= \sum w_n(x, X_i)Y_i f^*(x) = S_n^{(R)}(x) + S_n^{(A)}(x) \\ S_n^{(R)}(x) &= \sum w_n(x, X_i)Z_i Y_i f^*(x) \\ S_n^{(A)}(x) &= \sum w_n(x, X_i)(1 - Z_i)Y_i f^*(x) \end{aligned}$$

We want to show that $P(S_n^{(R)}(X) \leq M | X \in \gamma_n) \to 0 \; \forall \; M < \infty$. To do this we will lower bound $S_n^{(R)}(x)$ by a sum which is easier to analyze, and prove that sum diverges in probability if $x \in \gamma_n$. Let

$$\begin{aligned} &\tilde{S}_n^{(R)}(x) = \sum w_n(x, X_i)Z_i \tilde{Y}_i(x, X_i, Y_i)f^*(x) \\ &\tilde{Y}_i(x, X_i, Y_i) = Y_i \mathbf{1}_{Y_i \neq f^*(x)} + Y_i'(x, X_i)\mathbf{1}_{Y_i = f^*(x)} \\ &Y_n'(x, X_i) \in \{1, -1\} \\ &P(Y_n'(x, X_i) = f^*(x)) = \\ &\begin{cases} \frac{\inf_{z \in supp_n(x)} f(z)}{f(X_i)} & \text{if } f^*(x) = 1, X_i \in supp_n(x) \\ \frac{\inf_{z \in supp_n(x)} 1 - f(z)}{1 - f(X_i)} & \text{if } f^*(x) = -1, X_i \in supp_n(x) \\ 0 \quad \text{otherwise} \end{cases} \end{aligned}$$

Where the randomness in Y_i' is independent of everything

So by construction we have that $P(\tilde{Y}_i = f^*(x) | X_i \in supp_n(x)) = \inf_{z \in supp_n(x)} P(\tilde{Y}_i =$

$f^*(z)|X_i \in supp_n(x))$, $\tilde{Y}_i|X_i \in supp_n(x) \perp X_i$ and $Y_i f^*(x) \geq \tilde{Y}_i f^*(x)$. Since $x \in \gamma_n, P(\tilde{Y}_i f^*(x) = 1|X_i \in supp_n(x)) > \frac{1}{2} + \frac{|0.5-f(x)|}{2}$. By Condition 2 we have that $\sum_{i=1}^{n} \mathbf{1}_{X_i \in supp_n(X), Z_i=1} \xrightarrow{\mathrm{P}} \infty$ and $w_n(x, X_i)|Z_i = 1 \stackrel{d}{=} w_n(x, X_j)|Z_j = 1$, which gives us that $P(\tilde{S}_n^{(R)}(X) \leq M|X \in \gamma_n) \to 0 \,\forall\, M < \infty$ which in turn gives us that $P(S_n^{(R)}(X) \leq M|X \in \gamma_n) \to 0 \,\forall\, M < \infty$. And since ϵ was arbitrary this gives us $P(S_n^{(R)}(X) \leq M) \to 0 \,\forall\, M < \infty$.

Now in order for $S_n(x) < 0$ we need that $S_n^{(A)}(x) \to -\infty$. By defining $\tilde{S}_n^{(A)}(x)$ similarly, the same argument shows that this cannot happen. Since $w_n(x, y) \leq K$ we would require an infinite number of active samples in $supp_n(x)$. For each of these we would have $P(Y_i f^*(x) = 1|X_i \in supp_n(x)) > \frac{1}{2} + \frac{|0.5-f(x)|}{2}$, and so even though we can stop as soon as we are smaller than M, $P(\tilde{S}_n^{(A)}(x) \leq M) \to 0$ as $M \to -\infty$. Therefore $P(S_n(X) > 0) \to 1$ and $P(f^*(X) f_n(X, S_n) = -1) \to 0$.

□

CHAPTER IV

Active Federated Learning

Federated Learning allows for population level models to be trained without centralizing client data by transmitting the global model to clients, calculating gradients locally, then averaging the gradients. Downloading models and uploading gradients uses the client's bandwidth, so minimizing these transmission costs is important. The data on each client is highly variable, so the benefit of training on different clients may differ dramatically. To exploit this we propose *Active Federated Learning*, where in each round clients are selected not uniformly at random, but with a probability conditioned on the current model and the data on the client to maximize efficiency. We propose a cheap, simple and intuitive sampling scheme which reduces the number of required training iterations by 20-70% while maintaining the same model accuracy, and which mimics well known resampling techniques under certain conditions.

The structure of this chapter is as follows:

1. Briefly introduce federated learning (Section 4.1).
2. Propose the Active Federated Learning framework (Section 4.5).
3. Provide experimental comparisons of Active Federated Learning with standard federated learning (Section 4.6).

4.1 Federated learning overview

As machine learning models are deployed in the real world, the assumptions under which they were developed are often shown to be incompatible with user requirements. One such assumptions is unrestricted access to the training data, either on a single machine or distributed over many researcher controlled machines. Over the past few years there has been widespread backlash against indiscriminate acquisition of personal data. Due to privacy

concerns users may not want to transmit data from their own devices, making standard centralized training impossible. Federated learning (McMahan et al., 2016) enables the training of a single model on a central *server* with contributions from data on many users (also called *clients*) without any transmission of data. The basic methodology for training gradient optimized empirical risk minimization models (a class which includes many common deep neural networks) is quite simple:

1. At iteration t, broadcast the server model parameters $\mathbf{w}^{(t)}$ to each client selected for that training round.

2. Each client makes a small update to the parameters using their own data to produce local updated parameters $\mathbf{w}_k^{(t+1)}$ (where k is the client index). This could be as simple as a single step of gradient descent, but in practice it is often more complicated.

3. Each client transmits their locally updated model parameters to the server, which aggregates them to produce the next iteration of the server model parameters $\mathbf{w}^{(t+1)}$.

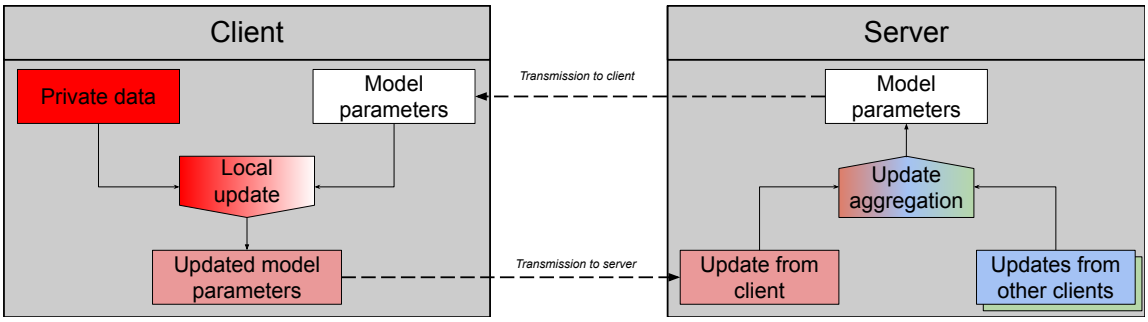

Figure 4.1: Federated learning schematic. Color (abstractly) represents the private information of the data at different stages of the update procedure.

Since being introduced in McMahan et al. (2016) federated learning has been identified as an important tool in privacy-preserving machine learning, with applications in tech (Yang et al., 2019), finance (Li et al., 2019a) and healthcare (Sheller et al., 2018). For excellent overviews we refer readers to Kairouz et al. (2019); Li et al. (2019b); Yang et al. (2019).

4.2 Introduction to Active Federated Learning

Federated Learning enables the training of models on this private data. However the procedure requires broadcasting all the model parameters to each client taking part in that

training round. These models can contain millions of parameters, making transmission costs between the server and the client are high. Additionally for many applications, the users own the broadband connecting their device to the internet, so broadcasting model parameters to clients utilizes the clients private resources. This makes reducing communication costs vitally important. In this chapter we introduce *Active Federated Learning* (AFL) to preferentially train on users which are more beneficial to the model during that training iteration. Motivated by ideas from Active Learning, we propose using a value function which can be evaluated on the user's device and returns a valuation to the server indicating the likely utility of training on that user. The server collects these valuations and converts them to probabilities with which the next cohort of users is selected for training. By using simple a value function related to the loss the user's data suffers under the current model, we can reduce the number of training rounds required.

4.3 Related work

Since its introduction (McMahan et al., 2016; Yang et al., 2019), reducing the communication costs of Federated Learning has been an important goal (Konečnỳ et al., 2016; Caldas et al., 2018). However as discussed in Li et al. (2019b) there are few existing techniques which change the method of selecting users. In Hartmann (2018) the author suggests stratification based on contextual information about the users, and in Nishio and Yonetani (2019) the authors group users based on hardware characteristics. In contrast this work is closer to active learning (Settles, 2009) where the selection policy is dependant on the current state of the model and the data on each user. It is also similar in spirit to non-uniform minibatching for SGD in Zhang et al. (2019). However the methods proposed in that paper rely on the selector having full access to the data itself, which is not possible in the federated setting.

Active learning and AFL share many similar structures, as in both the algorithm for selecting training data must act under imperfect information; in active learning the covariates are fully known, but the label of candidate data points is unknown, whereas in AFL both labels and covariates are fully known on each client, but only a summary is returned to the server. Additionally, in standard active learning individual data points may be selected in an unconstrained manner, whereas in AFL we train on all data points on each selected user, creating predetermined subsets of data.

4.4 Background and notation

Assume we have labelled data (x, y) and a model for predicting $y \in \mathcal{Y}$ given $x \in \mathcal{X}$ which we denote by $\hat{y} = f(x; \mathbf{w})$, where $\mathbf{w} \in \mathbb{R}^d$ are our model parameters. These model parameters will be learned by minimizing some loss function $l(x, y; \mathbf{w})$. Assume our training data is distributed over multiple clients (or users) $\mathcal{U} = \{U_1, ..., U_K\}$, where we denote the data of client U_k by $(\mathbf{x}_k, \mathbf{y}_k) \in \mathcal{X}^{n_k} \times \mathcal{Y}^{n_k}$. Our model parameters will be learned during training iterations, so we will let $\mathbf{w}^{(t)}$ denote the value of our parameters at training iteration t. During each training iteration we select a subset of users $\mathcal{S}^{(t)} \subset \mathcal{U}, |\mathcal{S}^{(t)}| = m$ and send $\mathbf{w}^{(t)}$ to each user in the set. Each user then performs some training $\mathcal{T}$ using their local data and produce updated model parameter values $\mathbf{w}_k^{(t+1)} = \mathcal{T}(\mathbf{x}_k, \mathbf{y}_k; \mathbf{w}^{(t)})$. In its most simple form this training could be a single step of gradient descent, though in practice it is often more complicated, such as multiple passes of SGD. In traditional Federated Learning the subsets $\mathcal{S}^{(t)}$ are selected uniformly at random and independently at each iteration. Our goal in AFL is to select our subsets $\mathcal{S}^{(t)}$ such that fewer training iterations are required to obtain a good model.

4.5 Active Federated Learning (AFL)

Inspired by the structure of classical AL methods, we propose the AFL framework which aims to select an optimized subset of users based on a *value* function that reflects how useful the data on that user is during each training round. Formally, we define a function $\mathcal{V} : \mathcal{X}^{n_k} \times \mathcal{Y}^{n_k} \times \mathbb{R}^d \to \mathbb{R}$ which is evaluated on each user. Once evaluated, each user U_k returns a corresponding *valuation* $v_k \in \mathbb{R}$ to the server, which is used to calculate the sampling distribution for the next training iteration. The valuations are a function of $\mathbf{w}^{(t)}$, but since transmitting the model is expensive we only get fresh valuations of users during an iteration in which we train on them, meaning that

$$v_k^{(t+1)} = \begin{cases} \mathcal{V}(\mathbf{x}_k, \mathbf{y}_k; \mathbf{w}^{(t)}) & \text{if } U_k \in S_t \\ v_k^{(t)} & \text{otherwise.} \end{cases}$$

Ideally the computation of the value function should require minimal additional computation, since the computations are done using the clients hardware, and should not reveal too much about the data on each client. Once the server has all valuations it converts them into a sampling distribution.

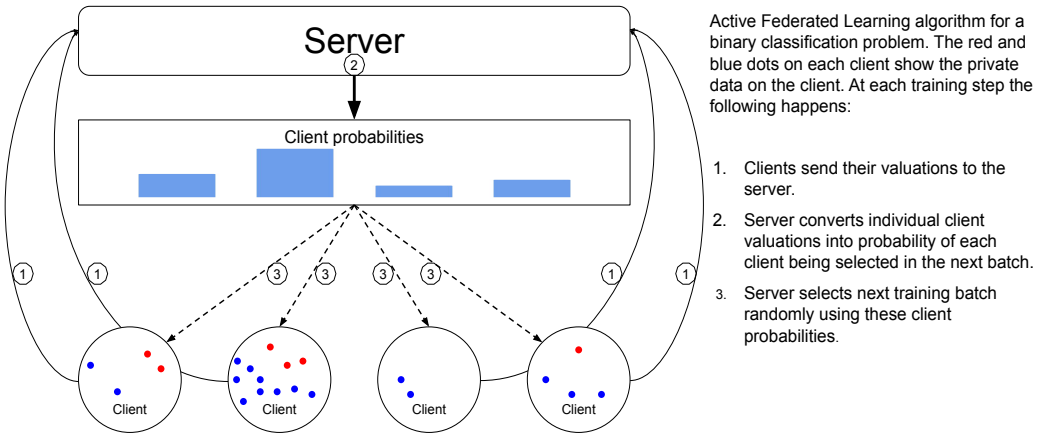

Figure 4.2: Active Federated Learning framework for a binary classification problem.

4.5.1 Loss valuation

One very natural value function is to use the loss of the users data

$$v_k = \frac{1}{\sqrt{n_k}} l(\mathbf{x}_k, \mathbf{y}_k; \mathbf{w})$$

.

It is already calculated during model training and is increasing with how poorly the model performs on the clients data. Additionally it mimics common resampling techniques when the required structure is present in the data. If there is extreme class imbalance and weak separation of the classes, data points of the minority class will have significantly higher loss than majority class data points. Therefore we will prefer users with more minority data, mimicking resampling the minority class data. Similarly if the noise depends on the distance from the classification boundary such as in (Blaschzyk et al., 2018), using the loss replicates margin based resampling techniques. Finally if all data points are equally valuable then users with more data will be given higher valuations. Most importantly these adaptations to the data do not require the practitioner to know the specific structure being exploited. This is particularly important in the Federated setting, where information about the data is limited.

4.5.2 Differential Privacy

Even summarizing the client data with a single float may reveal too much information. To properly protect users the value function should be reported using a Differentially Private mechanism Dwork et al. (2014). The noise introduced to maintain Differential Privacy may mislead the server into selecting sub-optimal clients. However there is structure

which might be exploited to reduce the corruption while still maintaining privacy. One is that many value functions, such as the loss, are not expected to change dramatically within a small number of training rounds. Thus we may be able to query whether a valuation has changed dramatically before querying the new value, similar to the Sparse Vector technique, to reduce the number of queries. We may also be able to adapt our value function to be more amenable to Differential Privacy. For example the loss value function has unbounded sensitivity and requires clipping to provide Differential Privacy. However returning a count of high loss data points has sensitivity 1 and may be less affected by the privacy providing noise. Adding privacy guarantees is an important challenge in AFL and is the subject of much future work.

4.6 Experimental results

We compared AFL to the standard uniform selection on two datasets; one on the Reddit dataset, the other on the Sticker Intent dataset. The Reddit dataset is a publicly available Baumgartner (2019) dataset consisting of comments from users on *reddit.com*. The authors were not involved in collecting this dataset. For the Reddit dataset we predicted the binary label 'controversially' based on the comment text, and selected 8K users at random from the November 2017 data set, similar to Bagdasaryan et al. (2018) but only excluding users with +100K messages. We removed comments being responded to from the messages, and empty messages. The Reddit dataset has many users who post few comments, but a long tail of power users. The Sticker Intent dataset has randomly selected, anonymized messages from a popular messaging app. The task was binary classification - predict whether a message was replied to using a sticker. Messages in this set were collected, de-identified, and annotated automatically; the messages were not read or labeled by human annotators.

Algorithm 7 for converting the valuations into a sampling distribution has 3 tuning parameters: The α_1 proportion of users with the smallest valuations will have their valuations set to $-\infty$. They can still be selected by random sampling. α_2 is our softmax temperature. α_3 is the proportion of users which are selected uniformly at random. In our experiments we used $\alpha_1 = 0.75, \alpha_2 = 0.01, \alpha_3 = 0.1$. We chose α_2 to ensure that the softmax did not produce $p_k = 0$ from underflow errors, and α_1, α_3 were both chosen based on initial experiments on Sticker Intent dataset. The underlying model trained with Federated Learning used a 64 dimensional character level embedding, a 32 dimensional BLSTM, and an MLP with one 64 dimensional hidden layer. The number of users in each Federated round was 200, and on each user 2 passes of SGD was performed with a batch size of 128. The

Algorithm 7: Sampling algorithm

Input: Client Valuations $\{v_1, ..., v_K\}$, tuning parameters $\alpha_1, ..., \alpha_3$, number of clients per round m

Output: Client indices $\{k_1, ..., k_m\}$

Sort users by v_k

For the $\alpha_1 K$ users with smallest v_k, $v_k = -\infty$

for k *from* 1 *to* K **do**

  $p_k \propto e^{\alpha_2 v_k}$

end

Sample $(1 - \alpha_3)m$ users according to their p_k, producing set $\mathcal{S}'$

Sample $\alpha_3 m$ from the remaining users uniformly at random, producing set $\mathcal{S}''$

return $\mathcal{S} = \mathcal{S}' \cup \mathcal{S}''$

	messages	users	% label 1	mean messages/user	median messages/user
Train	124638	7527	0.021	16.6	3
Test	15568	3440	0.021	4.5	2

Table 4.1: Reddit dataset statistics

updated model parameter values are returned to the server and aggregated to produce the next model parameters using Federated ADAM Leroy et al. (2019). The learning rates for both local SGD and Federated ADAM were tuned separately for Random Sampling and AFL and the optimal learning rates were used for each.

Figure 4.3 shows the AUC after each Epoch under uniform random selection of users, and with AFL selection, showing mean and standard errors from 10 repetitions on test data. AFL trains models of the same performance using 20-70% fewer Epochs (where one Epoch is enough training rounds to train on each client once in expectation under random sampling).

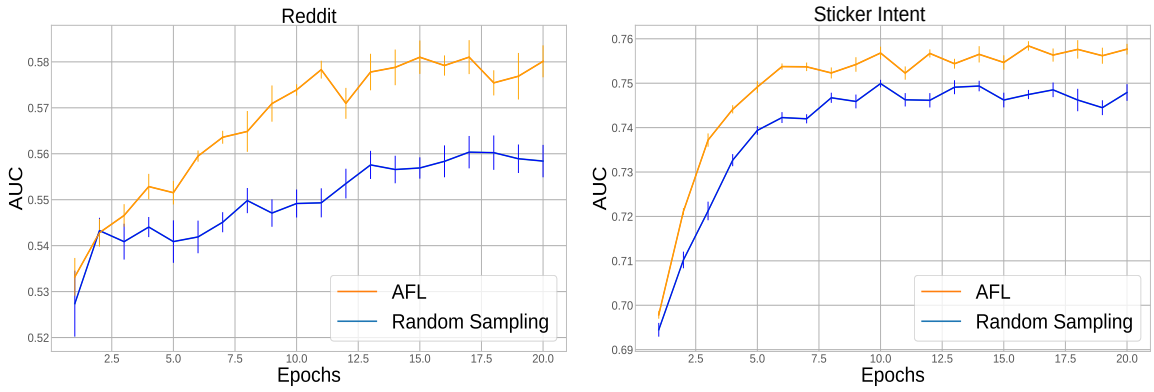

Figure 4.3: Comparison of AUC increase on Reddit and Sticker Intent datasets

4.6.1 Comparison with Resampling of minority class

One difference between AFL and server-side resampling techniques is that AFL selects data points by user, whereas server-side resampling can select arbitrary subsets. To explore the significance of this restriction we compared the gains from oversampling of label 1 data (He and Garcia, 2008) and server-side learning against AFL using the value function $v_k = \sum \mathbf{1}_{y_{i,k}=1}$ and Federated training, using the Reddit dataset. The level of resampling and learning rates were tuned for server training, as were the temperature α_2 and the learning rates for Federated training, and all other tuning parameters were kept the same. The difference between Random Sampling and Active Sampling is much larger for server-side learning with data point selection, compared to federated learning with user selection. Our results suggest that being restricted to sampling pre-defined subsets of data, as opposed to being able to select arbitrary sets of points, is a significant hindrance.

	Random Sampling	Active Sampling
Server selection of data points	0.559	0.615
Federated selection of clients	0.552	0.578

Table 4.2: Comparison of AFL and server side resampling

4.7 Conclusion and further directions

In this chapter we proposed Active Federated Learning (AFL), the first user cohort selection technique for FL which actively adapts to the state of the model and the data on each client. This adaptation allows us to train models with 20-70% fewer iterations for the same performance. Giving formal privacy guarantees is vital future work, but there are many other interesting extensions as well. These experiments were done under simplifying conditions which do not take into account many problems Federated Learning faces in practice, and which AFL may be able to help alleviate. For example clients may have different rates of availability for training. This availability may be correlated with the data on the client, resulting in bias in our model if not corrected. AFL which also takes reliability into account may be used to reduce this bias by increasing the rate at which we try to train on unreliable users. Another challenge is that clients are constantly gathering (and potentially forgetting) data, and in many cases the distribution may be non-stationary. Maintaining the benefits of AFL may require a principled way of ensuring no user goes too long without having their valuation refreshed. Finally our experiments and analyses focused on the classification setting, but the loss value function can be used for any supervised problem, and understanding

AFL with more complex models would be an interesting research direction.

CHAPTER V

Concluding remarks

5.1 Future work

There are a few general directions of active learning research which are more than direct extensions of the work in the above chapters of this thesis, but which these works suggest may be particularly interesting in the future.

5.1.1 Codifying the difference between population driven active learning, and sample driven active learning

As alluded to at the beginning of this thesis, there are multiple strategies active learning algorithms can take. Some of these mechanisms can be seen as *population driven*, attempting to replicate an optimal experimental design, the details of which depend on unknown population structure of the distribution. Algorithm 2 is an example of this, attempting to approximate (a specific example of) Algorithm 1, as are most margin based active learning methods. Other mechanisms can be seen as *sample driven*, adapting to the details of the existing samples. An example of this is Algorithm 5, and similar ideas come up in active teaching (Zhu et al., 2018). This distinction is not excluding as some methods, such as uncertainty sampling, implicitly do both. Such categorization can be useful for studying properties of active learning methods. In particular for sample based methods it is unlikely that the algorithm will produce the desired effect for all underlying distributions (see Section 2.9.1), and closer study of the conditions under which the algorithm behaves as intended would help practitioners select between available methods.

5.1.2 Generalizing methods for more diversity in information content

As active learning (and related ideas like AFL) are applied to increasingly complex real world situations, using methods for selecting a single point from a relatively simple $\mathcal{X} \times \mathcal{Y}$

may miss some valuable structure. In particular for some applications different points might intuitively contain different amounts of information. In AFL we see this when different clients have different numbers of samples, but similar structure appears in NLP (Siddhant and Lipton, 2018) where different documents will contain different numbers of words, and multi label classification (Reyes et al., 2017) where different samples will have different numbers of labels. Universal methods which can capture this difference in information content and effectively use it will be important for these applications.

5.1.3 Active learning under more complex constraints and structure

The majority of active learning work still assumes a classical statistical setup with data (X, Y) and unrestricted access to training data, with the exception of the labels which must be queried. However this is increasingly not the case. As is the case for AFL, there may be more complicated restrictions on what information you have access to about your training data. While there has been some work on active learning under privacy concerns Florina Balcan and Feldman (2013), this is likely to be an increasingly important consideration in machine learning as a whole, and so much be addressed in active learning. There may also be more complex structure in what part of the training data is being actively selected. An interesting situation is one where it is not the label, but a subset of the covariates which must be requested.

5.2 Conclusion

In this thesis we studied active learning in a variety of settings, and the types of results achieved in the different settings illustrates the persistent gulf between theory and application in active learning. Theoretical results were restricted to classical non-parametric estimators, whereas work using more modern methods such as neural nets and federated optimization contained no theoretical guarantees. This divide exists throughout the active learning field, and bridging that divide, though extremely challenging, is an important task as machine learning is applied to a wider range of domains.

Despite the the intuitive appeal of active learning, applying it in reality is a challenging and sometimes risky endeavour. Seemingly reasonable (and even theoretically guaranteed) methods can produce worse results than random sampling, so incorporating active learning into an existing modelling workflow often requires consideration and effort from the practitioner. Although the results in the thesis largely make minimal assumptions, producing truly significant gains from active learning almost always requires the existence (and

sometimes the prior knowledge of) special structure that can be exploited. Active learning should not be treated as a 'go to' or 'off the shelf' technique for improving the efficiency of any modelling pipeline, but instead as a situational technique to be used with care.

BIBLIOGRAPHY

Arias-Castro, E., Candes, E. J., and Davenport, M. A. (2013). On the fundamental limits of adaptive sensing. *IEEE Transactions on Information Theory*, 59(1):472–481.

Attenberg, J. and Provost, F. (2011). Inactive learning?: difficulties employing active learning in practice. *ACM SIGKDD Explorations Newsletter*, 12(2):36–41.

Awasthi, P., Balcan, M. F., and Long, P. M. (2014). The power of localization for efficiently learning linear separators with noise. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 449–458. ACM.

Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., and Shmatikov, V. (2018). How to backdoor federated learning. *arXiv preprint arXiv:1807.00459*.

Balcan, M.-F., Beygelzimer, A., and Langford, J. (2009). Agnostic active learning. *Journal of Computer and System Sciences*, 75(1):78–89.

Baum, E. B. and Lang, K. (1992). Query learning can work poorly when a human oracle is used. In *International joint conference on neural networks*, volume 8, page 8.

Baumgartner, J. (2019). Reddit Comments Dumps. https://files.pushshift.io/reddit/comments/. Accessed: 2019-09-03.

Beck, J. and Guillas, S. (2016). Sequential design with mutual information for computer experiments (mice): Emulation of a tsunami model. *SIAM/ASA Journal on Uncertainty Quantification*, 4(1):739–766.

Beygelzimer, A., Dasgupta, S., and Langford, J. (2009). Importance weighted active learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 49–56.

Beygelzimer, A., Hsu, D., Karampatziakis, N., Langford, J., and Zhang, T. (2011). Efficient active learning. In *ICML 2011 Workshop on On-line Trading of Exploration and Exploitation*.

Bıyık, E., Wang, K., Anari, N., and Sadigh, D. (2019). Batch active learning using determinantal point processes. *arXiv preprint arXiv:1906.07975*.

Blaschzyk, I., Steinwart, I., et al. (2018). Improved classification rates under refined margin conditions. *Electronic Journal of Statistics*, 12(1):793–823.

Breiman, L. (2000). Some infinity theory for predictor ensembles. Technical report, Technical Report 579, Statistics Dept. UCB.

Breiman, L. (2017). *Classification and regression trees*. Routledge.

Bull, A. D. et al. (2013). Spatially-adaptive sensing in nonparametric regression. *The Annals of Statistics*, 41(1):41–62.

Burbidge, R., Rowland, J. J., and King, R. D. (2007). Active learning for regression based on query by committee. In *International Conference on Intelligent Data Engineering and Automated Learning*, pages 209–218. Springer.

Caldas, S., Konečny, J., McMahan, H. B., and Talwalkar, A. (2018). Expanding the reach of federated learning by reducing client resource requirements. *arXiv preprint arXiv:1812.07210*.

Caruana, R. and Niculescu-Mizil, A. (2006). An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning*, pages 161–168.

Chaudhuri, K., Jain, P., and Natarajan, N. (2017). Active heteroscedastic regression. In *International Conference on Machine Learning*, pages 694–702.

Chaudhuri, K., Kakade, S. M., Netrapalli, P., and Sanghavi, S. (2015). Convergence rates of active learning for maximum likelihood estimation. In *Advances in Neural Information Processing Systems*, pages 1090–1098.

Chung, K. L. (2001). *A course in probability theory*. Academic press.

Cohn, D., Atlas, L., and Ladner, R. (1994). Improving generalization with active learning. *Machine learning*, 15(2):201–221.

Cohn, D. A., Ghahramani, Z., and Jordan, M. I. (1996). Active learning with statistical models. *Journal of artificial intelligence research*.

Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27.

Cramer, C. J. (2013). *Essentials of computational chemistry: theories and models*. John Wiley & Sons.

Crombecq, K., Gorissen, D., Deschrijver, D., and Dhaene, T. (2011). A novel hybrid sequential design strategy for global surrogate modeling of computer experiments. *SIAM Journal on Scientific Computing*, 33(4):1948–1974.

Cuong, N. V., Lee, W. S., and Ye, N. (2014). Near-optimal adaptive pool-based active learning with general loss. In *UAI*, pages 122–131.

Dasgupta, S. (2011). Two faces of active learning. *Theoretical computer science*, 412(19):1767–1781.

Dasgupta, S. (2012). Consistency of nearest neighbor classification under selective sampling. In *Conference on Learning Theory*, pages 18–1.

Dasgupta, S., Hsu, D. J., and Monteleoni, C. (2008). A general agnostic active learning algorithm. In *Advances in neural information processing systems*, pages 353–360.

Devroye, L., Györfi, L., and Lugosi, G. (2013). *A probabilistic theory of pattern recognition*, volume 31. Springer Science & Business Media.

Dwork, C., Roth, A., et al. (2014). The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407.

Eason, J. and Cremaschi, S. (2014). Adaptive sequential sampling for surrogate model generation with artificial neural networks. *Computers & Chemical Engineering*, 68:220–232.

Efromovich, S. (2008). Optimal sequential design in a controlled non-parametric regression. *Scandinavian Journal of Statistics*, 35(2):266–285.

Fedorov, V. V. (1972). *Theory of optimal experiments*. Elsevier.

Florina Balcan, M. and Feldman, V. (2013). Statistical active learning algorithms for noise tolerance and differential privacy. *arXiv preprint arXiv:1307.3102*.

Fu, Y., Zhu, X., and Li, B. (2013). A survey on instance selection for active learning. *Knowledge and information systems*, pages 1–35.

Genuer, R. (2012). Variance reduction in purely random forests. *Journal of Nonparametric Statistics*, 24(3):543–562.

Goetz, J., Malik, K., Bui, D., Moon, S., Liu, H., and Kumar, A. (2019). Active federated learning. *arXiv preprint arXiv:1909.12641*.

Goetz, J. and Tewari, A. (2019). Not all are made equal: Consistency of weighted averaging estimators under active learning. *arXiv preprint arXiv:1910.05321*.

Goetz, J., Tewari, A., and Zimmerman, P. (2018). Active learning for non-parametric regression using purely random trees. In *Advances in Neural Information Processing Systems*, pages 2537–2546.

Golovin, D. and Krause, A. (2011). Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *Journal of Artificial Intelligence Research*, 42:427–486.

Golzari, A., Sefat, M. H., and Jamshidi, S. (2015). Development of an adaptive surrogate model for production optimization. *Journal of Petroleum Science and Engineering*, 133:677–688.

Györfi, L., Kohler, M., Krzyzak, A., and Walk, H. (2006). *A distribution-free theory of nonparametric regression*. Springer Science & Business Media.

Hanneke, S. (2014). Theory of disagreement-based active learning. *Foundations and Trends® in Machine Learning*, 7(2-3):131–309.

Hanneke, S. and Yang, L. (2015). Minimax analysis of active learning. *Journal of Machine Learning Research*, 16(12):3487–3602.

Hanneke, S., Yang, L., et al. (2019). Surrogate losses in passive and active learning. *Electronic Journal of Statistics*, 13(2):4646–4708.

Hartmann, F. (2018). Federated learning.

He, H. and Garcia, E. A. (2008). Learning from imbalanced data. *IEEE Transactions on Knowledge & Data Engineering*, (9):1263–1284.

Hoang, T. N., Low, B. K. H., Jaillet, P., and Kankanhalli, M. (2014). Nonmyopic epsilon-bayes-optimal active learning of gaussian processes. *Proceedings of the 31st international conference on Machine learning*.

Jiang, P., Shu, L., Zhou, Q., Zhou, H., Shao, X., and Xu, J. (2015). A novel sequential exploration-exploitation sampling strategy for global metamodeling. *IFAC-PapersOnLine*, 48(28):532–537.

Jin, Y., Li, J., Du, W., and Qian, F. (2016). Adaptive sampling for surrogate modelling with artificial neural network and its application in an industrial cracking furnace. *The Canadian Journal of Chemical Engineering*, 94(2):262–272.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al. (2019). Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*.

Kolchin, V. F., Sevastyanov, B. A., and Chistyakov, V. P. (1978). *Random allocations*. Scripta series in mathematics. V. H. Winston. Translation of Sluchainye razmeshcheniia.

Konečnỳ, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T., and Bacon, D. (2016). Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*.

Körner, C. and Wrobel, S. (2006). Multi-class ensemble-based active learning. In *ECML*, volume 6, pages 687–694. Springer.

Krause, A., Singh, A., and Guestrin, C. (2008). Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9(Feb):235–284.

Lakshminarayanan, B., Roy, D. M., and Teh, Y. W. (2014). Mondrian forests: Efficient online random forests. In *Advances in neural information processing systems*, pages 3140–3148.

Leroy, D., Coucke, A., Lavril, T., Gisselbrecht, T., and Dureau, J. (2019). Federated learning for keyword spotting. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6341–6345. IEEE.

Lewis, D. D. and Gale, W. A. (1994). A sequential algorithm for training text classifiers. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–12. Springer-Verlag New York, Inc.

Li, S., Cheng, Y., Liu, Y., Wang, W., and Chen, T. (2019a). Abnormal client behavior detection in federated learning. *arXiv preprint arXiv:1910.09933*.

Li, T., Sahu, A. K., Talwalkar, A., and Smith, V. (2019b). Federated learning: Challenges, methods, and future directions. *arXiv preprint arXiv:1908.07873*.

Liu, H., Ong, Y.-S., and Cai, J. (2017). A survey of adaptive sampling for global metamodeling in support of simulation-based complex engineering design. *Structural and Multidisciplinary Optimization*, pages 1–24.

Liu, H., Xu, S., Ma, Y., Chen, X., and Wang, X. (2016). An adaptive bayesian sequential sampling approach for global metamodeling. *Journal of Mechanical Design*, 138(1):011404.

Loog, M. and Yang, Y. (2016). An empirical investigation into the inconsistency of sequential active learning. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pages 210–215. IEEE.

MacKay, D. J. (1992). Information-based objective functions for active data selection. *Neural computation*, 4(4):590–604.

Mak, S., Joseph, V. R., et al. (2018). Support points. *The Annals of Statistics*, 46(6A):2562–2592.

McMahan, H. B., Moore, E., Ramage, D., Hampson, S., et al. (2016). Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*.

Melville, P. and Mooney, R. J. (2004). Diverse ensembles for active learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 74. ACM.

Mourtada, J., Gaïffas, S., and Scornet, E. (2017). Universal consistency and minimax rates for online mondrian forests. In *Advances in Neural Information Processing Systems*, pages 3761–3770.

Mourtada, J., Gaïffas, S., and Scornet, E. (2018). Minimax optimal rates for mondrian trees and forests. *arXiv preprint arXiv:1803.05784*.

Mussmann, S. and Liang, P. S. (2018). Uncertainty sampling is preconditioned stochastic gradient descent on zero-one loss. In *Advances in Neural Information Processing Systems*, pages 6955–6964.

Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L. (1978). An analysis of approximations for maximizing submodular set functions—i. *Mathematical Programming*, 14(1):265–294.

Nishio, T. and Yonetani, R. (2019). Client selection for federated learning with heterogeneous resources in mobile edge. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pages 1–7. IEEE.

Nowak, R. (2008). Generalized binary search. In *Communication, Control, and Computing, 2008 46th Annual Allerton Conference on*, pages 568–574. IEEE.

Pan, G., Ye, P., Wang, P., and Yang, Z. (2014). A sequential optimization sampling method for metamodels with radial basis functions. *The Scientific World Journal*, 2014.

Resnick, S. I. (2013). *A probability path*. Springer Science & Business Media.

Reyes, O., Morell, C., and Ventura, S. (2017). Effective active learning strategy for multi-label learning. *Neurocomputing*.

Sabato, S. and Munos, R. (2014). Active regression by stratification. In *Advances in Neural Information Processing Systems*, pages 469–477.

Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989). Design and analysis of computer experiments. *Statistical science*, pages 409–423.

Santner, T. J., Williams, B. J., and Notz, W. I. (2013). *The design and analysis of computer experiments*. Springer Science & Business Media.

Scheffer, T., Decomain, C., and Wrobel, S. (2001). Active hidden markov models for information extraction. In *International Symposium on Intelligent Data Analysis*, pages 309–318. Springer.

Schein, A. I. and Ungar, L. H. (2007). Active learning for logistic regression: an evaluation. *Machine Learning*, 68(3):235–265.

Sen, A. (2012). On the interrelation between the sample mean and the sample variance. *The American Statistician*, 66(2):112–117.

Sener, O. and Savarese, S. (2017). Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*.

Seo, S., Wallat, M., Graepel, T., and Obermayer, K. (2000). Gaussian process regression: Active data selection and test point rejection. In *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on*, volume 3, pages 241–246. IEEE.

Settles, B. (2009). Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences.

Settles, B. (2010). Active learning literature survey. *University of Wisconsin, Madison*, 52(55-66):11.

Settles, B. (2012). Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114.

Settles, B. and 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*, pages 1070–1079. Association for Computational Linguistics.

Seung, H. S., Opper, M., and Sompolinsky, H. (1992). Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 287–294. ACM.

Shahsavani, D. and Grimvall, A. (2009). An adaptive design and interpolation technique for extracting highly nonlinear response surfaces from deterministic models. *Reliability Engineering & System Safety*, 94(7):1173–1182.

Sheller, M. J., Reina, G. A., Edwards, B., Martin, J., and Bakas, S. (2018). Multi-institutional deep learning modeling without sharing patient data: A feasibility study on brain tumor segmentation. In *International MICCAI Brainlesion Workshop*, pages 92–104. Springer.

Siddhant, A. and Lipton, Z. C. (2018). Deep bayesian active learning for natural language processing: Results of a large-scale empirical study. *arXiv preprint arXiv:1808.05697*.

Singh, P., Deschrijver, D., and Dhaene, T. (2013). A balanced sequential design strategy for global surrogate modeling. In *Simulation Conference (WSC), 2013 Winter*, pages 2172–2179. IEEE.

Sourati, J., Akcakaya, M., Leen, T. K., Erdogmus, D., and Dy, J. G. (2017). Asymptotic analysis of objectives based on fisher information in active learning. *Journal of Machine Learning Research*, 18(34):1–41.

Stone, C. J. (1977). Consistent nonparametric regression. *The annals of statistics*, pages 595–620.

Sugiyama, M. and Kawanabe, M. (2012). *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*. MIT press.

Sutton, R. S. and Barto, A. G. (1998). *Introduction to reinforcement learning*, volume 135. MIT press Cambridge.

Tong, S. and Koller, D. (2001). Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66.

Tur, G., Hakkani-Tür, D., and Schapire, R. E. (2005). Combining active and semi-supervised learning for spoken language understanding. *Speech Communication*, 45(2):171–186.

Viana, F. A., Picheny, V., and Haftka, R. T. (2010). Using cross validation to design conservative surrogates. *Aiaa Journal*, 48(10):2286.

Vijayanarasimhan, S. and Grauman, K. (2009). What's it going to cost you?: Predicting effort vs. informativeness for multi-label image annotations. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2262–2269. IEEE.

Willett, R., Nowak, R., and Castro, R. M. (2006). Faster rates in regression via active learning. In *Advances in Neural Information Processing Systems*, pages 179–186.

Williams, D. (1991). *Probability with martingales*. Cambridge university press.

Yang, Q., Liu, Y., Chen, T., and Tong, Y. (2019). Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):12.

Zakai, A. and Ritov, Y. (2009). Consistency and localizability. *Journal of Machine Learning Research*, 10(Apr):827–856.

Zhang, C., Öztireli, C., Mandt, S., and Salvi, G. (2019). Active mini-batch sampling using repulsive point processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5741–5748.

Zhu, X., Singla, A., Zilles, S., and Rafferty, A. N. (2018). An overview of machine teaching. *arXiv preprint arXiv:1801.05927*.