TEL AVIV UNIVERSITY אוניברסיטת תל-אביב

THE IBY AND ALADAR FLEISCHMAN
FACULTY OF ENGINEERING

הפקולטה להנדסה
על שם איבי ואלדר פליישמן

THE IBY AND ALADAR FLEISCHMAN FACULTY OF ENGINEERING
The Zandman-Slaner Graduate School of Engineering

Information Theoretic Active Learning

by

Shachar Shayovitz

THESIS SUBMITTED TO THE SENATE OF TEL-AVIV UNIVERSITY
in partial fulfillment of the requirements for the degree of
“DOCTOR OF PHILOSOPHY”

June, 2024

TEL AVIV UNIVERSITY אוניברסיטת תל-אביב

THE IBY AND ALADAR FLEISCHMAN
FACULTY OF ENGINEERING

הפקולטה להנדסה
על שם איבי ואלדר פליישמן

THE IBY AND ALADAR FLEISCHMAN FACULTY OF ENGINEERING
The Zandman-Slaner Graduate School of Engineering

Information Theoretic Active Learning

by

Shachar Shayovitz

THESIS SUBMITTED TO THE SENATE OF TEL-AVIV UNIVERSITY
in partial fulfillment of the requirements for the degree of
"DOCTOR OF PHILOSOPHY"

Under the Supervision of Prof. Meir Feder

June, 2024

This work was carried out under the supervision of
Prof. Meir Feder

Acknowledgments

I would like to thank my parents Channa and Sorin Shayovitz, who from a young age, encouraged me to explore, ask questions and nurtured a strong sense of curiosity. They instilled in me a joy of learning, which made me into the person I am today.

I would like to thank my wife Zohar, who without her encouragement and belief in me, I would have never considered leaving my comfort zone and embarking on this journey. Without her constant support, I cannot imagine having made it to the finishing line.

I would like to thank my supervisor, Prof. Meir Feder, for the wonderful opportunity of working together. His devoted and joyful guidance have tremendously enriched my professional capabilities. Working together, not only directed me to solving difficult problems in Information Theory, but more importantly taught me how to choose a worthwhile problem and distill it to its essence, and in addition how to read, write, and speak science.

Last but not least, I would like to acknowledge the Yitzhak and Chaya Weinstein Research Institute for Signal Processing at Tel Aviv University, for the very helpful financial assistance.

To my children,
Ori and Iftach

Abstract

Active learning is a learning paradigm where the training data is actively and purposely chosen. One of its main goals is to optimize a model's performance by minimizing the number of annotated samples. Recent active learning leading strategies are based on the assumption that the training pool has the same distribution as the test set, which may not be the case in privacy-sensitive applications where user data cannot be annotated.

In the first part of the research, the stochastic setting for data is considered. A new information theoretic active learning criterion is proposed based on a Redundancy-Capacity theorem of universal source coding. This criterion naturally induces an exploration - exploitation trade-off in feature selection and generalizes previously proposed heuristic criteria. The new criterion is compared analytically and empirically to other commonly used active learning criteria.

Next, the linear hyper-plane hypotheses class with asymmetric label noise is considered. We propose a low complexity algorithm which learns the optimal hyperplane. The algorithm is inspired by the Posterior Matching scheme for communication with feedback with an adaptation to high dimensions. We utilize the previous Capacity - Redundancy theorem to show that for general label noise and bounded feature distribution, the minimax redundancy of this algorithm decays exponentially fast to zero.

In the second part of the research, we consider the individual setting, which does not assume a probabilistic relationship between the training and test data. Motivated by universal source coding, we propose a criterion that chooses to label data points that minimize the min-max regret on the test set. It is shown that for binary classification and linear regression, the resulting criterion coincides with well known active learning criteria and thus represents a unified information theoretic active learning approach for general hypothesis classes. Finally, it is shown, using real data that the proposed criterion outperforms other active learning criteria in terms of sample complexity. By applying an approximate version of our individual criterion to neural networks, we show that in the presence of out-of-distribution data, the proposed criterion reduces the required training set size by up to 15.4%, 10.4%, and 12% for CIFAR10, EMNIST, and MNIST datasets respectively.

Contents

List of Figures

List of Tables

Nomenclature

AWGN	Additive White Gaussian Noise
BSC	Binary Symmetric Channel
BAC	Binary Asymmetric Channel
pNML	Predictive Normalized Maximum Likelihood
DMC	Discrete Memory-less Channel
GPC	Gaussian Process Classification
MC	Monte Carlo
UAL	Universal Active Learning
PM	Posterior Matching
DNN	Deep Neural Network
IAL	Individual Active Learning
DIAL	Deep Individual Active Learning
MU	Maximum Uncertainty
BALD	Bayesian Active Learning by Disagreement
$\mathbb{1}\left(\cdot\right)$	the indicator function, equals one if the condition holds and zero otherwise

Publications

Shachar Shayovitz, Koby Bibas, and Meir Feder. Deep individual active learning: Safeguarding against out-of-distribution challenges in neural networks. *Entropy*, 26(2):129, 2024.

Shachar Shayovitz and Meir Feder. Minimax active learning via minimal model capacity. In *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2019.

Shachar Shayovitz and Meir Feder. Universal active learning via conditional mutual information minimization. *IEEE Journal on Selected Areas in Information Theory*, 2(2):720–734, 2021.

Shachar Shayovitz and Meir Feder. Active learning for individual data via minimal stochastic complexity. In *2022 58th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1–5. IEEE, 2022.

Shachar Shayovitz and Meir Feder. Active learning via predictive normalized maximum likelihood minimization. *IEEE Transactions on Information Theory*, 70(8):5799–5810, 2024.

Chapter 1

Introduction

Many machine learning applications today rely heavily on large labeled data-sets and on the assumption that humans can annotate all the available data for training. The evolution of vision data-sets has been greatly influenced by MNIST and ImageNet from 1998 to 2016. As shown in Figure 1.1 there was a growth of 0.11 OOMs/year (orders of magnitude per year) in the size of the data sets used for training vision models. Datasets for language models have grown by 0.23 OOMs/year since 1990, for a total growth of 7 OOMs between 1990 and 2022 as shown in Figure 1.2. The data suggests that this growth has been faster since 2014. It's remarkable that up until 2014 there were notable models trained on very little data, such as Deep Belief Networks (trained on less than 200 thousand words), but after 2016 virtually all notable models have been trained on more than 100 million words. This might reflect the adoption of more efficient architectures such as Transformers that allow training on much more data.

Moreover, data storage became cheap enough that companies started hoarding data without even knowing quite what to do with it. Data collection became ubiquitous, due to the internet of things, which allowed for entire new streams of valuable data. Data processing benefited immensely from the emerging power of GPUs and TPUs to train more robust deep learning models. Having a preponderance of data to power business operations is, generally speaking positive.

Since ML models use supervised learning, the success of these projects is hugely dependent on a company's ability to label its data accurately and efficiently. Data is annotated by experts (Oracles) tasked with generating what is called "Ground Truth". These annotations are denoted as labels and represent the class or value that we want the trained models to predict for a specific data point. For example, an annotator might look at an image and label an object from a pre-existing ontology of classes. The label is fed to the model for the process of learning via training. In a sense, labeling is the injection of human knowledge into the model. This is a critically important step in developing high-performance ML models. At the risk of being reductive, good labels drive good models. This is also the main bottleneck in an ML project since the experts are typically expensive and their work is time consuming.

The data collection, annotation and learning process is graphically presented in Figure 1.3. We assume a large unlabelled pool of data points is available for labelling. Since we cannot ask an oracle to label all the data points in this pool, a subset is randomly collected and labelled. Consequently, only a small sub-set is labeled which

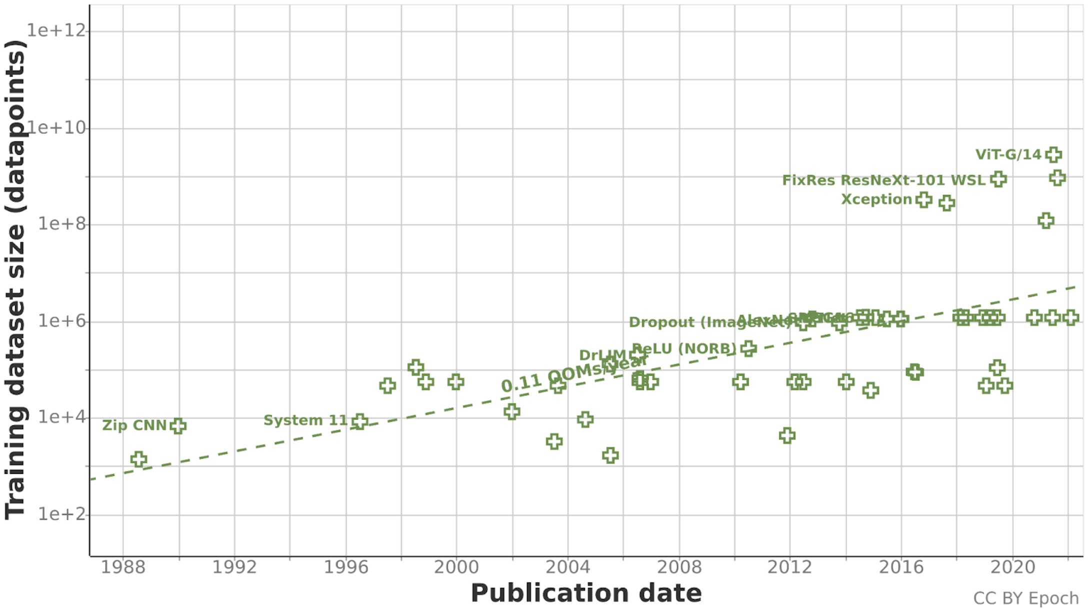


Figure 1.1: Size of Data-sets for Vision

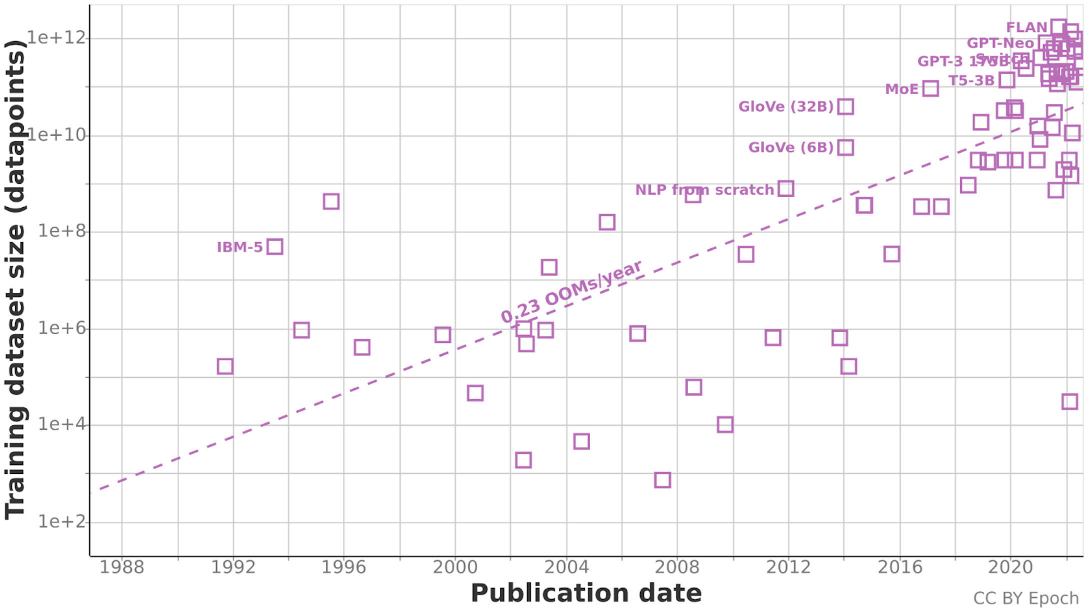


Figure 1.2: Size of Data-sets for Language

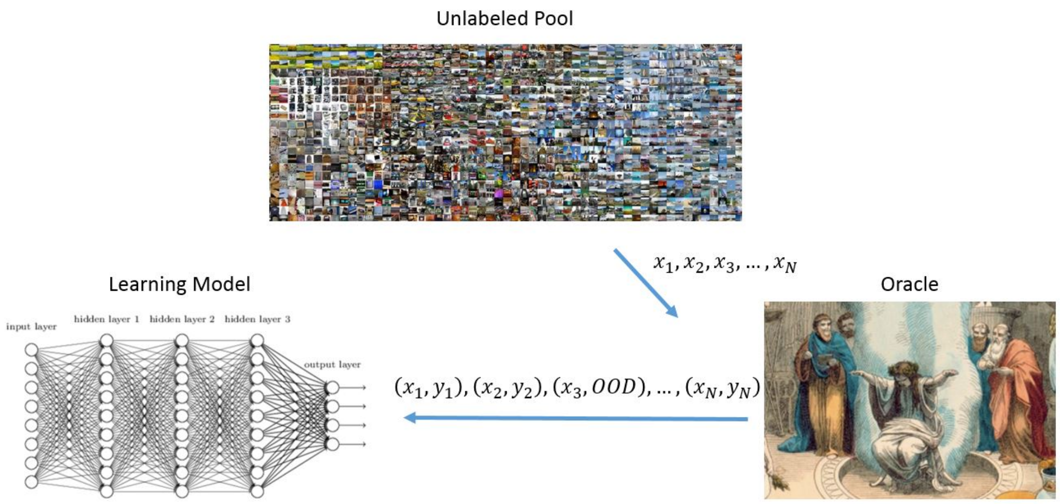


Figure 1.3: Passive Learning

may be un-representative of the true underlying model between features and labels, thus large generalization errors might occur. To avoid this, the training set is redundant and usually larger than required. Therefore, generalization bounds for passive learning error probability do not decay exponentially fast.

Due to the fact that data is ubiquitous and collected from multiple sources, the unlabelled pool may contain out-of-distribution (OOD) data which is irrelevant for the learning task. For example, a learning model classifies cats versus dogs and some images in the unlabelled pool also contain frogs. If we input the frog images to our learning model then what should it output? In this case, the oracle will label these images as OOD and they will not be included in the training phase. However, since the oracle took time to label them as OOD, it will still be included in the labelling budget. For large scale data sets, a pre-processing removal of OOD samples is a very costly process and if it can be embedded into the data selection process it would be extremely beneficial.

Active learning is a framework in which the learner can interact with a labeling expert by sequentially selecting the most informative samples for the expert to label based on previously observed labeled data. Therefore, reducing the number of examples needed to achieve a given accuracy level [1]. The traditional active learning setting is shown in Figure 1.4. In the last decade there has been significant progress in active learning research. Most rigorous results and bounds are, however, for binary linear classification or regression problems. Most papers deal with proposing a heuristic for feature selection, analyzing its performance and comparing to different lower bounds [2], [3] and [4]. Some of the algorithms and heuristics which have been proposed for active learning include: [5], [6], [7], [2], [8], [9], [10], [11] and [12].

One well studied approach is based on the disagreement region introduced by Hanneke in [4]. This region contains all the features for which at least two candidate learners do not agree on. Thus, querying the label of such a feature may be helpful to reduce the candidate pool. The general algorithmic framework of disagreement based active learning in the presence of noise was introduced with the A^2 algorithm by Balcan in [6] and other related work in [13], [10] and [14].

Another approach which has proven effective is margin based active learning which has better label and computational complexity than disagreement based approaches. The idea is not to sample features in all the disagreement region but at carefully selected regions inside, specifically near the edges of this region. This approach was introduced in [7] and continued in [15] and [16]. While this approach has much better computational complexity than the disagreement based approach, it is not robust to noise. Also, since this algorithm samples points based on some known prior distribution on the features, the exponential decay will only work for log-concave distributions.

In addition, several approaches consider information-theoretic criteria for selecting features [17], [18] and [19]. The most common method is uncertainty sampling or Maximum Uncertainty (MU), where the feature with the highest label entropy given the training is selected. In some sense this approach is very similar to the margin based approach in [15]. However, this aggressive, essentially greedy, scheme may lead to large generalization errors since noise might produce very high entropy and corrupt the training set. Suppose a very noisy feature is presented to the learner, then the probability assigned to all the labels will be very low (essentially uniform), causing the label entropy to be very high. The learner will thus learn the noise modalities instead of useful information.

In [19] an information theoretic criterion is proposed which is based on maximizing the mutual information between the model and the selected features and provides good performance. The criterion is based on reducing the number of possible hypotheses maximally fast, i.e. to minimize the uncertainty about parameters using Shannon's entropy. This criterion also appears as an upper bound on information based complexity of stochastic optimization in [20] and also for experimental design of experiments in [21] and [22]. This criterion represents the average reduction in uncertainty on the model θ after observing the label Y_t of feature X_t based on the available training. Since this maximization is generally very difficult, a greedy algorithm is proposed, which seeks the data point X_t that maximizes the decrease in expected posterior entropy. This approach was empirically investigated in [23], where a Bayesian method to perform deep learning was proposed and several heuristic active learning acquisition functions were explored within this framework. It was shown that the performance of this criterion, denoted as BALD , was the best. However, this criterion does not to take into account the test distribution $p(x)$ and thus may select examples which are not informative for the test case at hand.

The underlying assumption for most of the schemes discussed above is that the distributions of the unlabeled pool and the test set are the same. However, this may not always be true, particularly in privacy-sensitive applications where real user data cannot be annotated [24] and the unlabeled pool may contain irrelevant information. In such cases, choosing samples from the unlabeled pool may not necessarily improve model performance on the test set. In order to mitigate this distribution shift, we propose a new setting for active learning as presented in Figure 1.5. The idea is to extend the classical active learning framework in Figure 1.4 by making the learner aware of the test set (access to an **unlabeled** subset of the test set). Test samples may not be annotated due to privacy or other consideration but we assume a very small sample of unlabeled test points can be collected and provided to the active learner. We will show using different

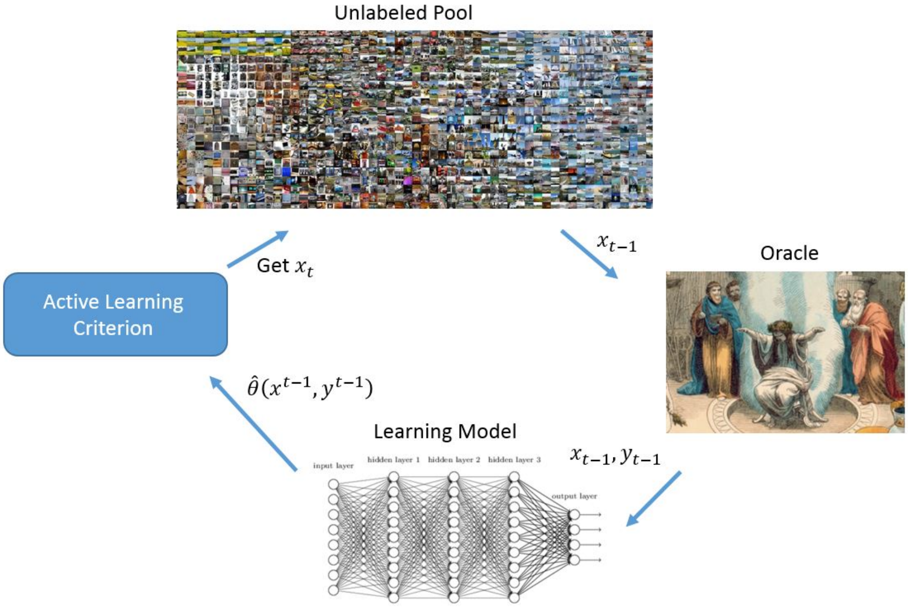


Figure 1.4: Standard Active Learning Setting

assumptions on the data, that this setup allows the derivation of active learning criteria which outperform current state of the art schemes on real data sets.

Following the foundational work on universal prediction done by Feder and Merhav in [25], we divide active learning into two settings: stochastic and individual. The difference between the two is whether the data follows some distribution (stochastic) or not (individual). In the first part of the research, comprising Chapter 2 and Chapter 3 and based on our contributions in [26, 27], we study the stochastic setting and propose an active learning criterion which is based on a Redundancy-Capacity theorem. We analyze its performance for different hypothesis classes and compare it to other information theoretic active learning criteria. We also design a low complexity, noise robust algorithm for active learning in the multi dimensional linear separator hypothesis class with label noise.

In the second part of the research, comprising Chapter 4 and Chapter 5 and based on our contributions in [28, 29, 30], the individual setting is considered. An active learning criterion is proposed based on predictive Normalized Maximum Likelihood (pNML) [31]. The idea is to find a training set for which the minimax regret (with respect to a genie learner) is minimized over the unlabeled test sample. We show that this criterion can be viewed as a generalization of other well known criteria for different hypothesis classes. Also, a low complexity selection algorithm is derived for Deep Neural Networks (DNN) based on MC-Dropout [32]. It is shown that this algorithm outperforms current state of the art active learning schemes on real datasets with distribution shift. Chapter 6 concludes this study, giving concluding remarks and presenting open questions.

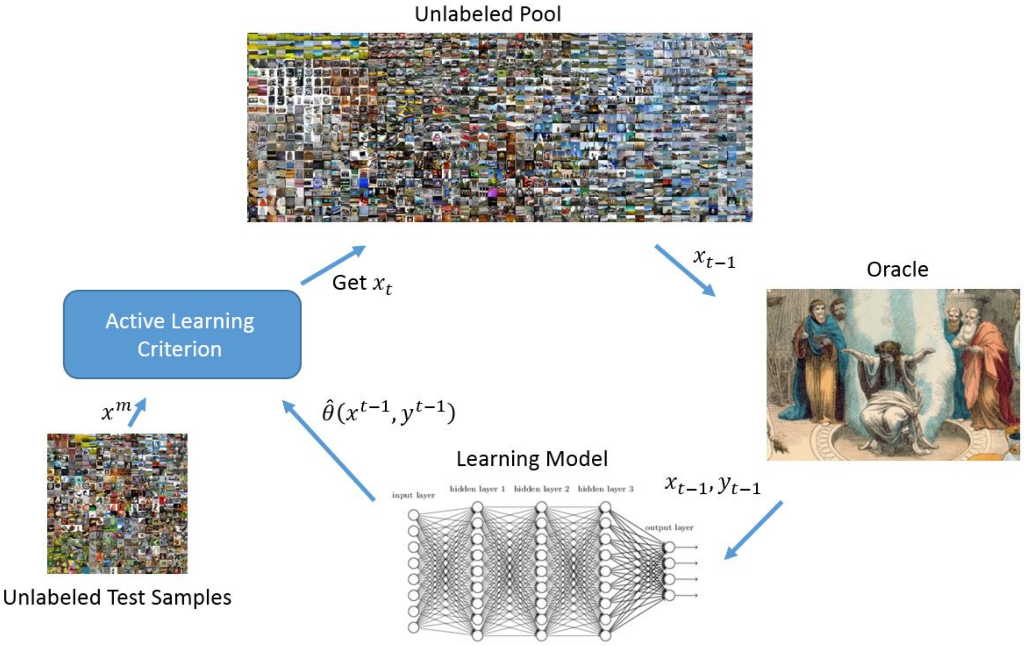


Figure 1.5: Test Aware Active Learning Setting

Part I

Active Learning in the Stochastic Setting

Chapter 2

Universal Active Learning via Conditional Mutual Information Minimization

In this part of the research, comprising Chapter 2 and Chapter 3, active learning in the stochastic setting is discussed and is based on the work in [26, 27]. In Chapter 2, a new active learning criterion is proposed which is based on a Capacity - Redundancy theorem. This criterion is analyzed and its advantages over existing criteria are discussed. The main advantage of the proposed criterion is for the distribution shift scenario. If the test distribution, $p_{test}(x)$, differs from the training set distribution, $p_{training}(x)$, traditional active learning criteria fail to provide good test time error. What is maximally informative for model estimation may not be maximally informative for test time prediction.

We propose to solve this issue by optimizing the training set while taking into account the un-labelled test set. Since the trained model will be tested using the test set, one should select training points which have the most relevance to the test set. Essentially, there is no real need to learn the labeling function over the whole feature space which may be very complex and require many data points. Traditionally, a pre-processing stage prunes the training set from data points which are irrelevant to the test, but this requires domain knowledge regarding the similarity between training and test sets.

Criteria such as BALD and MU do not take into account the un-labelled test set and select data points based solely on the training pool. In [27], a criterion denoted as Universal Active Learning (UAL) was derived based on universal source coding and minimax regret minimization. UAL utilizes the unlabeled test set in order to learn data points which are most relevant to the test set. It was shown in [27] that UAL is related to BALD and MU and is basically a generalization of the two. In [33], UAL is also proposed using heuristic arguments and denoted as Expected Predictive Information Gain (EPIG).

In chapter 3, active learning for linear separators is addressed and a low complexity, noise robust algorithm, denoted as Successive Posterior Matching (SPM) is proposed. It is shown via simulations that SPM achieves asymptotically optimal sample complexity for this hypothesis class. Moreover, it is proved that SPM generates an active learning selection policy for which the criterion presented in chapter 2, decays exponentially fast to zero. Using the Capacity - Redundancy theorem, this means that the minimax Redundancy also decays exponentially fast to zero for SPM.

2.1 Stochastic Setting

In this section, the stochastic setting for learning is presented and discussed. This setting was presented in the context of universal prediction in [25] and the adaptation to learning is presented in this section. The main assumption is that the data follows some stochastic parametric hypothesis class. This assumption is very useful for deriving theorems but is not easily verifiable in real world applications. However, it will be shown that empirically, the derived criterion and algorithms provide good performance in real world applications.

Denote Θ as a general index set, this class is a set of conditional probability distributions, or sometimes referred to as the hypothesis class:

$$P_\Theta = \{p\left(y|x,\theta\right)|\theta \in \Theta\} \tag{2.1}$$

We assume a training set consisting of N pairs of examples is provided to the learner:

$$z^N = \{(x_n, y_n)\}_{n=1}^N \tag{2.2}$$

where x_n is the n-th data point and y_n is its corresponding label

In the stochastic setting, the data's distribution follows:

$$p\left(y^N|x^N,\theta_0\right) = \Pi_{n=1}^N p\left(y_n|x_n,\theta_0\right) \tag{2.3}$$

where $p(y|x,\theta_0) \in P_\Theta$ with $\theta_0 \in \Theta$.

The goal of a learner (passive or active) is to predict an unknown test label y given its test data, x, by assigning a probability distribution $q\left(\cdot|x,z^N\right)$ for each training z^N. In all subsequent sections, performance is evaluated using the log-loss function, i.e. $-\log\left(q\left(\cdot|x,z^N\right)\right)$. We do not make a distinction yet between active and passive learning since we assume that the data is provided to the learner and still do not discuss how it was acquired.

Had the learner known θ_0, then the best probability assignment would be:

$$q\left(\cdot|x,z^N\right) = p\left(y|x,\theta_0\right)$$

and there is no need for the training z^n.

However, θ_0 is unknown and needs to be estimated based on z^N. Therefore, we formulate the following learning problem as the expected log-loss difference (regret) between a learner $\mathbf{q}$ and the reference learner (which knows θ_0):

$$R\left(q;x,\theta_0,z^n\right) = \mathbf{E}\left\{\log\left(\frac{p\left(y|x,\theta_0\right)}{q\left(y|x,z^n\right)}\right)\right\} \tag{2.4}$$

where the expectation is over the conditional $p\left(y|x,\theta_0\right)$. The regret measures how good a learner q is compared to the best learner $p\left(y|x,\theta_0\right)$.

Since we have no access to θ_0, it is not feasible to optimize (2.4). Therefore, we propose to modify the learning objective and to minimize the maximal (worst θ) expected log-loss regret of this learner. In other words, if we do not know the regret to the correct θ_0, we can upper bound the regret by the worst case regret. If for a specific training set the worst case regret will be low, then the true regret (for θ_0) will be lower.

The minimax log loss regret, R_ϕ, after learning N examples for a specific feature selection policy ϕ, is:

$$R_\phi = \min_q \max_\theta \mathbf{E}\left\{\log\left(\frac{p\left(y|x,\theta\right)}{q\left(y|x,x^N,y^N\right)}\right)|\theta\right\} \tag{2.5}$$

where the expectation in (2.5) is performed over the joint probability:

$$\begin{aligned} p\left(y,x,x^N,y^N|\theta\right) = p\left(y|\theta,x\right)\Pi_{t=1}^{N} p\left(y_t|x_t,\theta\right) \\ \phi\left(x_t|x^{t-1},y^{t-1}\right)p(x) \end{aligned} \tag{2.6}$$

and $\phi(x_t|x^{t-1},y^{t-1})$ is the sequential selection policy which gives a probability distribution for each training feature, x_t, based *only* on the past observed training data x^{t-1}, y^{t-1}. Another assumption we make is that $p(x|\theta) = p(x)$ since the feature prior should be independent of the model.

Remark 1. *Note that the selection may be stochastic, which means that after observing the past examples there may be some randomness in choosing the next feature. For example, in passive learning, the distributions* $\left\{\phi(x_t|x^{t-1},y^{t-1})\right\}_{t=1}^N$ *are uniform, since the examples are drawn uniformly from the training pool.*

Remark 2. *We would like emphasize the fact that we are concerned with the prediction problem which requires minimum loss on the predicted test label given test feature and not with estimating* θ_0*. There is a conceptual difference between the model estimation and prediction problems and we argue that in real world applications the most important thing is the prediction error and not if the model is well estimated. Sometimes it is a much harder problem to do model estimation than prediction, especially when we have distribution shift between the training and test sets.*

2.2 Universal Active Learning

Now we can use the regret defined in the section above to present the active learning setup. In active learning, the objective is to sequentially select features and collect N training examples (examples contain features $x^N = \{x\}_{i=1}^N$ and labels $y^N = \{y\}_{i=1}^N$). This training set is used to find a probabilistic learner for a test label y, given a test feature x: $q\left(y|x,x^N,y^N\right)$, such that it will perform as close as possible to the best learner in the hypotheses class: $p(y|x,\theta)$, i.e. the Oracle. Essentially, we would like to find the best learner for a given N, without knowing the best learner. A related analysis for passive learning was provided in [31] but assumes i.i.d training samples.

Following the formulation in the previous section, we want to optimize the selected policy, ϕ. Therefore we would like to minimize (2.5) over $\left\{\phi(x_t|x^{t-1},y^{t-1})\right\}_{t=1}^N$. The final active learning problem formulation can be stated as finding the policy ϕ which minimizes R_ϕ, i.e:

$$R = \min_{\{\phi_t\}_{t=1}^N} \min_q \max_\theta \mathbf{E} \left\{ \log \left(\frac{p\left(y|x,\theta\right)}{q\left(y|x,x^N,y^N\right)} \right) \right\} \tag{2.7}$$

The following theorem is the basis for active learning in the stochastic setting:

Theorem 1 (Redundancy-Capacity). *The minimax active learning problem defined in (2.7) is equivalent to the conditional model capacity,*

$$R = \min_{\left\{\phi\left(x_t|x^{t-1},y^{t-1}\right)\right\}_{t=1}^N} \max_{\pi(\theta)} I\left(Y;\theta|X,Y^N,X^N\right) \tag{2.8}$$

and the optimal learner is:

$$q^*\left(y|x,x^N,y^N\right) = \sum_\theta p\left(\theta|y^N,x^N\right) p\left(y|\theta,x\right) \tag{2.9}$$

where $\pi(\theta)$ is a capacity achieving distribution for the channel $\theta \to Y$ given the test X and training X^N, Y^N and $I\left(X;Y|Z\right)$ is the mutual information between X and Y given Z.

The proof for Theorem 1 is provided in Appendix A.1. Note that for any prior distribution $\pi(\theta)$ with $\theta \in \Theta$, any policy ϕ and a given model class $p(y|x,\theta)$, the mutual information $I(\theta;Y|X,X^N,Y^N)$ is well defined. The active learning designer finds ϕ and π which solve the minimax problem in (2.8). Once $\pi(\theta)$ is known, the optimal learner q^* is given by (2.9) for any realization of x^N, y^N, x.

Theorem 1, is denoted as the *Redundancy-Capacity* theorem since it is very similar to the classical result in universal prediction, with the same name, proposed in [34]. In universal prediction, a stream of samples is given sequentially to a predictor and the objective is to predict the next sample based on the constraint that the samples originate from a source belonging to some predefined set of distributions. The *Redundancy-Capacity* links the minimax prediction problem with channel capacity.

Theorem 1, proposes a new criterion for optimal selection policy in active learning. The objective is to find a selection policy which will *minimize* the conditional capacity between the model parameters and test label given the test feature and training data. This is different than active learning strategies used today which do not take into account the test feature prior, $p(x)$, and instead maximize the mutual information between the training and model, ignoring the test set if available. In practical applications, if the test set is available, then there will be a dedicated pre-processing stage to prune the training set from data points which seem irrelevant to the test scenario. This step is implicitly preformed by the proposed criterion. This has the potential to significantly improve performance in active learning for priors which are multi-modal and help avoid learning sub-spaces of features which are non-informative for the test scenario. There is of course an issue on how to find $p(x)$, and if such a probability even exists. However, since the main bottleneck in machine learning is the labeling process and not the amount of training features, then we can assume we can estimate the feature probability in some way and come up with an approximation of $p(x)$ or the relative occurrences of features in real life.

Another useful formulation for the minimax log-loss regret can be the following:

$$R = \min_{\{\phi_t\}_{t=1}^N} \min_q \max_{\pi(\theta)\in\Pi} E\left\{\log\left(\frac{p\left(y|x,\theta\right)}{q\left(y|x,x^N,y^N\right)}\right)\right\} \tag{2.10}$$

where x^N, y^N are the training examples and Π is a set of distributions on the random variable θ.

This formulation is less restrictive than the previous and will be useful for regularized linear regression and Gaussian Process Classification where the prior on θ is either regularized or explicitly given. Also note that for the alternative formulation proposed in 2.10, the prior $\pi(\theta) \in \Pi$ and the same Theorem holds. The expectation is performed over the joint probability:

$$p\left(y,x,x^N,y^N\right) = p\left(y|\theta,x\right)\Pi_{t=1}^N p\left(y_t|x_t,\theta\right)\phi\left(x_t|x^{t-1},y^{t-1}\right)p(x)\pi(\theta) \tag{2.11}$$

The following theorem states that the optimal $\phi(x_t|x^{t-1},y^{t-1})$ places all the probability mass on a specific feature, and is essentially deterministic given the history.

Theorem 2 (Optimal Selection Policies)**.** *The selection policies which optimize (2.8) are deterministic:*

$$\phi(x_t|x^{t-1},y^{t-1}) = \delta\left(x_t - f\left(x^{t-1},y^{t-1}\right)\right)$$

where $\delta(\cdot)$ is the Dirac or Kronecker delta function for continuous or discrete x_t respectively and $f\left(x^{t-1},y^{t-1}\right)$ is a deterministic function from the history x^{t-1},y^{t-1} sequence to a feature x_t.

The proof in provided in Appendix A.2.

The optimization of (2.8) is unfortunately intractable for many hypotheses classes. The reason is that the number of candidate policies grows exponentially fast and thus infeasible to search for the best possible policy. Moreover, the objective function is not sub-modular or adaptively sub-modular [35] and thus greedy algorithms are not guaranteed to converge in the general case.

The proposed criterion, which is denoted as Universal Active Learning (UAL), can be decomposed in the following manner using the chain rule:

$$\begin{aligned} I\left(\theta;Y|X,Y^N,X^N\right) = I\left(\theta;Y|X\right) + I\left(\theta;Y^N|X^N,Y,X\right) \\ - I\left(\theta;Y^N|X^N\right) \end{aligned} \tag{2.12}$$

$I\left(\theta;Y|X\right)$ does not depend on the selection policy and the optimization is only on the difference between two other mutual information terms. We denote $I\left(\theta;Y^N|X^N,Y,X\right)$ and $I\left(\theta;Y^N|X^N\right)$ as the exploitation and exploration respectively.

Exploitation in our case is the minimization of $I\left(\theta;Y^N|X^N,Y,X\right)$, which means that if the test feature and label, (X,Y), were known in advance, then we would like to select training examples which will be as correlative to the test as possible. We can use the fact that:

$$I\left(\theta; Y^N|X^N, Y, X\right) \leq H\left(Y^N|X^N, Y, X\right)$$

Therefore, finding X^N which minimize $H\left(Y^N|X^N, Y, X\right)$ would mean that $I\left(\theta; Y^N|X^N, Y, X\right)$ will also be small. Finding X^N which result with low conditional entropy, $H\left(Y^N|X^N, Y, X\right)$, means that if we knew X, Y, then the uncertainty on the training data would be small. For example, selecting X^N to be very "similar" to X would result with such low conditional entropy. This criterion also takes into account the prior probability $p(x)$ and tries to find the best examples averaged across this prior. This means that we are exploiting the test data and trying to reduce uncertainty.

Exploration in our case is maximization of $I\left(\theta; Y^N|X^N\right)$ which is identical to BALD [19]. This basically means that one wants to find the most uncertain example in the pool. Therefore, UAL balances between exploration and exploitation and finds the most informative example given the specific test set at hand.

2.2.1 Comparison with other Information Theoretic criteria

In this section the relation between UAL and other criteria such as BALD [19] and Maximum Uncertainty (MU) [18] is analyzed. It will be shown that UAL is a generalization of the two criteria which takes additional information on the test set into consideration. First, a brief review of these criteria is provided. The MU criterion [18] selects the feature based on:

$$x_t^* = \underset{x_t}{\operatorname{argmax}}\, H(Y_t|X_t = x_t, x^{t-1}, y^{t-1}) \tag{2.13}$$

MU selects a feature in the training set whose conditional label entropy based on the current training set is the highest. Since the current model cannot label this feature well, then this example can improve the learning process best. However, this example may be noisy and produce high entropy, thus the learner will now add noise to the training set and this is of course not helpful to the learning task and the labeling budget.

The BALD criterion is defined as:

$$x_t^* = \underset{x_t}{\operatorname{argmax}}\, I(\theta; Y_t|X_t = x_t, x^{t-1}, y^{t-1}) \tag{2.14}$$

According to [19], the objective is to find a feature x_t that maximises the decrease in expected posterior entropy and that will reduce the hypotheses class as fast as possible. It is obvious by the definition of mutual information, MU is an upper bound on BALD.

Both of these criteria make sense in an intuitive manner but lack a clear justification as a solution to some optimization problem. There is no clear concept what is the prior of the model θ and no use of the prior on the test features, $p(x)$. Nevertheless, BALD is used to produce very good results for active learning using deep neural network [36] and [23]. Essentially, these criteria are more focused on model estimation and less on prediction.

In order to relate BALD to UAL, for discrete valued labels Y, (2.12), becomes:

$$I\left(\theta; Y|X, Y^N, X^N\right) \leq I\left(\theta; Y|X\right) + \sum_{i=1}^{N} H\left(Y_i|X_i, Y, X\right) - I\left(\theta; Y^N|X^N\right) \tag{2.15}$$

which can be further simplified using the chain rule:

$$I\left(\theta;Y|X,Y^N,X^N\right)\leq I\left(\theta;Y|X\right)+N\log_2\left(|\mathcal{A}|\right)-\sum_{i=1}^{N}I\left(\theta;Y_i|X_i,Y^{i-1}X^{i-1}\right)\quad(2.16)$$

where $Y\in\mathcal{A}$ and $|\mathcal{A}|$ is the size of the alphabet of the random label Y.

We would like to minimize the upper bound on UAL. The first two terms in (2.16) are constant and thus the minimization of the R.H.S is only performed on the third term, which turns into a maximization of $\sum_{i=1}^{N}I\left(\theta;Y_i|X_i,Y^{i-1}X^{i-1}\right)$ which is identical to optimizing BALD. The approximation we took is to upper bound $\sum_{i=1}^{N}H\left(Y_i|X_i,Y,X\right)$ which provides correlation between the candidate points and the test set.

The same argument can be taken on MU since:

$$I\left(\theta;Y|X,Y^N,X^N\right)\leq I\left(\theta;Y|X\right)+2N\log_2\left(|\mathcal{A}|\right)-\sum_{i=1}^{N}H\left(Y_i|X_i,Y^{i-1}X^{i-1}\right)\quad(2.17)$$

Therefore, UAL is a generalization of BALD and MU which takes into account test data.

2.3 Active Learning for Linear Regression

In this section, UAL is applied to the linear regression hypothesis class. It is shown that UAL aligns with commonly used criteria for this setting. This serves as an example for the fact that UAL is a general framework for active learning and coincides with the best known AL methods per hypothesis class. It will be shown (as expected) that for this hypothesis class there is no need for oracle feedback but an offline subset selection process can provide the optimal sample complexity result.

2.3.1 Linear Regression Model

We consider the problem of estimating a vector of unknown parameters $\underline{\theta}\in\mathbb{R}^p$ from observed measurements or experiments $\{\underline{x}_i,y_i\}_{i=1}^{n}$, assuming a linear relationship between $\underline{x}_i$ and y_i:

$$y_i=\underline{x}_i^T\underline{\theta}+z_i, i=1,2,...,n\quad(2.18)$$

where $\underline{x}_i\in\mathbb{R}^p$ is the i-th input feature vector, $y_i\in\mathbb{R}$ is its corresponding output response and $z_i\sim\mathcal{N}\left(0,\sigma^2\right)$ is zero-mean Gaussian noise.

If we stack the feature vectors $\underline{x}_i$ as rows in matrix $X\in\mathbb{R}^{nxp}$ and stack the output responses y_i as a column vector $\underline{y}\in\mathbb{R}^n$, we can write the system of linear equations in matrix form:

$$\underline{y}=X\underline{\theta}+\underline{z}$$

where $\underline{z}\in\mathbb{R}^n$ is a Gaussian noise vector with zero mean and a covariance matrix equal to a scaled identity matrix ($E\left[\underline{z}\underline{z}^T\right]=\sigma^2I_{nn}$).

Under the Gaussian noise condition, the Maximum Likelihood estimator to a linear regression problem is called the Ordinary Least Squares (OLS) solution:

$$\underline{\hat{\theta}}_{OLS}=\left(X^TX\right)^{-1}X^T\underline{y}\quad(2.19)$$

The error $\underline{\epsilon} = \underline{\theta} - \hat{\underline{\theta}}_{OLS}$ is a random Gaussian vector and we can look at the mean and covariance to characterize the OLS's performance.

$$\begin{aligned} E[\underline{\epsilon}] &= E[\underline{\theta} - \hat{\underline{\theta}}_{OLS}] \\ &= E[\underline{\theta} - \left(X^T X\right)^{-1} X^T \left(X\underline{\theta} + \underline{z}\right)] \\ &= E[\left(X^T X\right)^{-1} X^T \underline{z}] = 0 \end{aligned} \tag{2.20}$$

the fourth equality is due to the fact that the noise is assumed zero mean.

The error covariance is computed as follows:

$$\begin{aligned} &E[\underline{\epsilon\epsilon}^T] = E\left(\left(X^T X\right)^{-1} X^T \underline{z}\right)\left(\left(X^T X\right)^{-1} X^T \underline{z}\right)^T \\ &= E\left(\left(X^T X\right)^{-1} X^T \underline{zz}^T X \left(X^T X\right)^{-1}\right) = \sigma^2 \left(X^T X\right)^{-1} X^T X \left(X^T X\right)^{-1} = \sigma^2 \left(X^T X\right)^{-1} \end{aligned} \tag{2.21}$$

Note that X might not be full rank ($rank(X) \leq p$) and thus $X^T X$ will not be invertible. A common practice is to use diagonal loading to increase the rank of this correlation matrix. This act is very common in signal processing and is equivalent to assuming that the model vector is a Gaussian $\underline{\theta} \sim \mathcal{N}\left(0, \sigma_\theta^2 I\right)$. This prior is commonly incorporated in linear regression models for regularization purposes.

After adding the power constraint on the model vector we get the robust version of OLS. This can also be interpreted as the Maximum A Posteriori (MAP) solution or equivalently in the Gaussian zero mean case, the MMSE solution:

$$\hat{\underline{\theta}}_{MMSE} = E\left(\underline{\theta}\underline{y}^T\right) E\left(\underline{yy}^T\right) \underline{y} \tag{2.22}$$

which is equivalent in the linear case to:

$$\hat{\underline{\theta}}_{MMSE} = \sigma_\theta^2 X^T \left(\sigma_\theta^2 X X^T + \sigma^2 I\right)^{-1} \underline{y} = X^T \left(X X^T + \frac{\sigma^2}{\sigma_\theta^2} I\right)^{-1} \underline{y} \tag{2.23}$$

Using SVD, we can write the MMSE estimator as:

$$\hat{\underline{\theta}}_{MMSE} = \left(X^T X + \frac{\sigma^2}{\sigma_\theta^2} I\right)^{-1} X^T \underline{y} \tag{2.24}$$

The mean of the new estimator is of course biased per specific θ but is unbiased when considering the prior on $\underline{\theta}$:

$$\begin{aligned} &E[\underline{\epsilon}] = E[\underline{\theta} - \hat{\underline{\theta}}_{MMSE}] = E[\underline{\theta} - \left(X^T X + \sigma_\theta^2 I\right)^{-1} X^T \left(X\underline{\theta} + \underline{z}\right)] \\ &= E[\left(I - \left(X^T X + \sigma_\theta^2 I\right)^{-1} X^T X\right)\underline{\theta} + \left(X^T X + \sigma_\theta^2 I\right)^{-1} X^T \underline{z}] = E[\left(X^T X + \sigma_\theta^2 I\right)^{-1} X^T \underline{z}] = 0 \end{aligned} \tag{2.25}$$

where the fourth equality is due to the fact that the noise is assumed zero mean.

The error covariance is computed using the standard MMSE identities:

$$\begin{aligned} E[\underline{\epsilon\epsilon}^T] &= E\left(\underline{\theta\theta}^T\right) - E\left(\underline{\theta y}^T\right) E\left(\underline{yy}^T\right)^{-1} E\left(\underline{y\theta}^T\right) \\ &= \sigma_\theta^2 I - \sigma_\theta^2 X^T \left(\sigma_\theta^2 X X^T + \sigma^2 I\right)^{-1} X \sigma_\theta^2 \\ &= \sigma_\theta^2 \left(I - \left(X^T X + \frac{\sigma^2}{\sigma_\theta^2} I \right)^{-1} X^T X \right) \end{aligned} \tag{2.26}$$

We can see that for $\sigma^2 = 0$, then the error covariance will be zero as expected.

The above analysis was done for model estimation, meaning that we wish to estimate θ as best as possible. However, as indicated earlier, in real world machine learning applications, we are more interested in the prediction error on the test matrix X_{test} and its corresponding responses $\underline{y}_{test}$.

The prediction error for OLS is:

$$\begin{aligned} E[\|\underline{y}_{test} - \underline{\hat{y}}_{OLS}\|^2] &= E[\|X_{test}\underline{\theta} - X_{test}\underline{\hat{\theta}}_{OLS}\|^2] \\ = E[\|X_{test}\left(\underline{\theta} - \underline{\hat{\theta}}_{OLS}\right)\|^2] &= \sigma^2 Tr\{X_{test}\left(X^T X\right)^{-1} X_{test}^T\} \end{aligned} \tag{2.27}$$

Respectively, the prediction error for the MMSE estimator (Regularized):

$$\begin{aligned} E[\|\underline{y}_{test} - \underline{\hat{y}}_{MMSE}\|^2] &= E[\|X_{test}\underline{\theta} - X_{test}\underline{\hat{\theta}}_{MMSE}\|^2] \\ = E[\|X_{test}\left(\underline{\theta} - \underline{\hat{\theta}}_{MMSE}\right)\|^2] &= Tr\left\{ X_{test}\sigma_\theta^2 \left(I - X^T \left(X X^T + \frac{\sigma^2}{\sigma_\theta^2} I \right)^{-1} X \right) X_{test}^T \right\} \end{aligned} \tag{2.28}$$

Note that:

$$\sigma_\theta^2 I - \sigma_\theta^2 X^T \left(\sigma_\theta^2 X X^T + \sigma^2 I\right)^{-1} X \sigma_\theta^2 = \left(\frac{1}{\sigma_\theta^2} I + \sigma^{-2} X^T X \right)^{-1} \tag{2.29}$$

$$E[\|\underline{y}_{test} - \underline{\hat{y}}_{MMSE}\|^2] = Tr\left\{ X_{test} \left(\frac{1}{\sigma_\theta^2} I + \sigma^{-2} X^T X \right)^{-1} X_{test}^T \right\} \tag{2.30}$$

Therefore, the two error covariance matrices are very similar (up to regularization factor) and we will use them in order to optimize the design matrix.

2.3.2 Optimal Design of Experiments

Optimal design of experiments [37] is a branch of mathematical statistics that involves finding the best way to design an experiment in order to achieve certain objectives, such as minimizing the variance of the estimated parameters or maximizing the precision of the estimates. In this field, the design of an experiment is considered to be optimal if it provides the most information with the fewest observations or measurements.

The goal of optimal design is to find a design matrix that will provide the most information about the parameters of interest. This is typically done by minimizing some measure of the variance of the estimated parameters, such as the determinant of the covariance matrix or the trace of the inverse of the covariance matrix. One of the fundamental concepts in the optimal design of experiments is the concept of a design matrix. A design matrix is a matrix of values that specifies the values of the independent variables or factors that are to be varied in an experiment. The rows of the design matrix correspond to the experimental units or observations, while the columns correspond to the independent variables or factors.

The classical experimental design is defined as selecting a small subset $S \subset \{1, ..., n\}$ of r rows: X_S, from X so that estimating $\underline{\theta}$ is optimized on the selected design X_S. Using the selected training set, one can derive the OLS solution for the parameter vector $\underline{\theta}$. Since we are looking for S such that X_S is most statistically efficient, the optimal design problem reduces to minimizing the inverse covariance matrix (as we saw earlier for the error covariance):

$$\Sigma^{-1} = \left(X_S^T X_S\right)^{-1}$$

Optimal design can be done using a variety of techniques, including algebraic optimization, numerical optimization, and Bayesian methods. These techniques involve finding a design matrix that maximizes some criterion, such as the determinant of the information matrix, which is a measure of the amount of information that can be obtained from the experiment. In [37] several optimality criteria have been described for measuring how well Σ^{-1} is minimized on a selected design X_S. Some common criteria include A-optimality, V-optimality, and D-optimality. In [38], performance guarantees for the greedy solution of experimental design problems are provided. In particular, it focuses on A optimal designs, for which typical guarantees do not apply since the mean-square error of the estimation error covariance matrix is not sub-modular. In the next sections we present several optimal design criteria.

2.3.2.1 A Optimal Design

A-optimal design is a criterion for the optimal design of experiments that seeks to minimize the trace of the inverse of the covariance matrix of the estimated parameters. The covariance matrix measures the variability of the estimated parameters, and minimizing its trace ensures that the estimates are as precise as possible.

More specifically, let Y be a vector of observations or measurements that depend on a vector of unknown parameters θ, and let X be a design matrix that specifies the values of the independent variables or factors in the experiment. The goal of A-optimal design is to find a design matrix X that minimizes the trace of the inverse of the covariance matrix of the estimated parameters, denoted by $(X^T X)^{-1}$.

Therefore, the A-optimal design criterion can be formulated as the following optimization problem:

$$\hat{X} = \arg\min_X Tr\left((X^T X)^{-1}\right)$$

subject to some constraints on X, such as the number of observations or the range of the independent variables.

The trace of the inverse of the covariance matrix can then be written as:

$$Tr((X^T X)^{-1}) = \Sigma_{i=1}^{p} \frac{1}{\lambda_i}$$

where λ_i are the eigenvalues of $(X^T X)^{-1}$.

This problem can be solved using various optimization techniques, such as linear programming, quadratic programming, or convex optimization. The resulting design matrix X provides the optimal allocation of resources to the different parts of the experiment, in order to obtain the most precise estimates of the parameters of interest.

2.3.2.2 V Optimal Design

V-optimal design is a criterion for the optimal design of experiments that seeks to minimize the average prediction variance, i.e. the variance of the prediction variable Y. Therefore, the V-optimal design criterion can be formulated as the following optimization problem:

$$\hat{X} = \arg\min_X Tr\left(X_{test}(X^T X)^{-1} X_{test}^T\right)$$

where X_{test} is a matrix whose rows are different features from a test set. The idea is to minimize the average prediction error on the test set. This problem can be solved using various optimization techniques, such as linear programming, quadratic programming, or convex optimization.

2.3.2.3 D Optimal Design

D-optimal design is a criterion for the optimal design of experiments that seeks to minimize the determinant of Σ^{-1}, which is proportional to the volume of the confidence ellipsoid (for a fixed confidence level). Thus, D-optimality shrinks the ellipsoid in all directions in order to minimize total volume:

$$\hat{X} = \arg\min_X \det\left(X^T X\right)^{-1}$$

subject to some constraints on X, such as the number of observations or the range of the independent variables. Equivalently, one can maximize the determinant of the information matrix $X^T X$.

D-optimal design is known for its efficiency and effectiveness in providing high-quality data for complex models. However, finding D-optimal designs can be computationally intensive, as it involves solving a non-linear optimization problem. Despite this, the benefits of achieving high precision in parameter estimation make D-optimal design a popular choice in many scientific and engineering applications.

2.3.3 Universal Active Learning for Linear Regression

The goal of active learning in this setting is to pick a small number of feature vectors, $\underline{x}^N$, from the space of possible features so that the underlying model, which relates input variables to output responses, is estimated accurately. As we saw in the last section, the linear regression model has the property that the error covariance matrix depends on neither the true parameter vector $\underline{\theta}$ nor the observed response y. This suggests that we can "optimize" the covariance of the estimator a-priori, even before taking any measurements, transforming the problem from interactive querying an oracle to subset selection of feature vectors.

In the following theorem, we find an expression for UAL using the linear regression hypothesis class. We define the linear regression hypothesis class with a power regularization factor.

Theorem 3. *Consider the hypothesis class:*

$$P_\Theta = \{p\left(y|x,\theta\right)|\theta \in \mathbb{R}^d, \mathbb{E}\left(\theta\right) = 0, Tr\left(\mathbb{E}\left(\theta\theta^T\right)\right) \le \sigma_\theta^2\}$$

$$p\left(y|x,\theta\right) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}\left(y - x^T\theta\right)^2\right)$$

where $x \in \mathbb{R}^d$ and σ^2 is known a-priori.

Then UAL is equivalent to the following subset selection problem:

$$C_n = \min_{\underline{x}^n} Tr\left(\mathbb{E}\left(\underline{x}_{test}\underline{x}_{test}^T\right)\left(X_n^T X_n + \frac{\sigma^2}{\sigma_\theta^2} I_d\right)^{-1}\right)$$

where X and X_n are the concatenation of the test vectors and the concatenation of the training vectors respectively.

The full proof is in appendix A.3.

Remark 3. *An interesting result of Theorem 3 is that for linear regression, there is a closed form solution for the conditional capacity minimization (UAL). More importantly, the capacity achieving distribution, $\pi(\theta)$ is a Gaussian with zero mean and covariance $\sigma_\theta^2 I$. When we will consider other hypothesis classes, this will not be the case and the capacity achieving distribution will be very difficult to derive.*

One can notice that UAL in this case is equivalent to (2.30) and V-optimal design (up to a regularization factor). This is interesting since UAL was derived as a general active learning criterion, not specific to any hypothesis class. The equivalence to V-optimal design shows that once an hypothesis class is selected, UAL will converge to a well known criterion which was designed specifically to that family. The criterion is also equivalent to the transductive experimental design proposed heuristically in [39] and UAL has provided the mathematical reasoning for this criterion.

Note that this criterion is a function of the training features $\underline{x}^n$ only and has no dependence on their respective labels y^n. Therefore, there is no real need for online feedback in the active linear regression problem and the training set problem can be cast as a subset selection problem performed offline. This problem is NP hard and thus approximate solutions are needed.

Remark 4. *If we take BALD and MU for linear regression, we observe that BALD will sequentially maximize the conditional mutual information* $I\left(\theta; y^n|\underline{x}^n\right)$. *It is not clear which prior* $\pi(\underline{\theta})$ *should be used for BALD since this was not addressed in [19]. Thus, we will use the same prior used in UAL and the same entropy calculation to get the respective BALD criterion:*

$$\min_{\underline{x}^n} I\left(\theta; y^n|\underline{x}^n\right) = \min_{\underline{x}^n} \log\det\left(Q\left(\underline{x}^n\right)\right) \tag{2.31}$$

BALD converges to D-optimal design [37]. Note that D-optimal design is a submodular objective and thus greedy optimisation, as BALD suggests, will provide a close to optimal solution [38].

In conclusion, UAL and BALD converge to two different experimental design criteria which are suited for different applications as described in [37]. Note that MU in this case will be identical to BALD since $h(y_t|\underline{x}_t, \underline{\theta}, \underline{x}^{t-1}, y^{t-1}) = h(z)$, *and is not a function of the trianing set.*

2.4 Gaussian Process Classification

In this section, UAL is compared to BALD, MU and passive learning in an empirical test using Gaussian Process Classification (GPC) over a synthetic data set. Note that GP's are essentially an assumption on the prior of the model parameters and therefore the alternative formulation of the minimax regret is taken with a specific given prior. GP's are a powerful and popular non-parametric tool for regression and classification and a detailed introduction to them can be found in [40].

2.4.1 Gaussian Process

A Gaussian process is a statistical method which allows for supervised regression of data. Classification of data is also possible, but needs extra computations through approximations which are not Gaussian. The Gaussian process is named after the Gaussian probability distribution, from which it is a generalization [40]. A Gaussian process describes a distribution over functions. Every point of a continuous input space is associated with a normally distributed random variable. The joint distribution of all these random variables is the distribution of the Gaussian process.

The theoretical foundation for Gaussian processes was already formulated in the 1940's with Wiener-Kolgomorov predictions and time series analysis. In 1978 Gaussian processes were introduced to one-dimensional curve fitting by O'Hagan [41, 42]. Gaussian processes were popularized in machine learning in the 1990's by the upcoming of backpropagation in neural networks and the introduction of a Bayesian framework [43, 44]. Gaussian processes were developed independently in geo-spatial sciences in the 1950's with the name Kriging [45]. Gaussian processes as statistical methods have a direct relationship to Machine Learning algorithms. Neal showed that large

Neural Networks converge eventually to Gaussian processes [44]. GP advocates sometimes claim that the equivalent model for a neural network as a Gaussian process would be easier to handle and interpret than the neural network [46], since the models are analytical tractable [47]. Furthermore, GPs perform well on binary classification problems and score results comparable to neural networks. A major disadvantage of a Gaussian process is the computational effort. The implementation features inversions of matrices of the size *nxn* where *n* is the number of points where the Gaussian process is evaluated. Using standard linear algebra techniques, the inversion of this matrix has complexity $O(n^3)$.

The Gaussian process is a generalization of the Gaussian distribution. The mean vector is replaced with a mean function and the covariance matrix is replaced with a covariance function. It is a stochastic process specified by its mean function μ and covariance function k:

$$\mu\,(x) = \mathbb{E}\{f\,(x)\}$$

$$k\,(x, \tilde{x}) = \mathbb{E}\{f\,(x - \mu(x))\,f\,(\tilde{x} - \mu(\tilde{x}))\}$$

where x is a input.

The notation for a Gaussian process

$$f \sim GP(\mu(\cdot), k(\cdot, \cdot))$$

where f is a function that is distributed as a Gaussian process.

The Gaussian process is defined by the mean function μ and the covariance function k. The mean function $\mu(x)$ describes the mean value for every dimension. During preprocessing, the input data is usually centred around the origin of the coordinate system. The covariance function describes the similarity of two data points as a scalar. The notion of similarity is essential in machine learning problems, since similar inputs usually yield similar outputs and the applied covariance function should reflect this. According to Rasmussen and Williams the covariance function "is the crucial ingredient in a Gaussian process predictor, as it encodes our assumptions about the function we wish to learn." [46]. Mathematically, the covariance function is defined on a pair of inputs and can be any positive semi-definite function $k\,(x, \tilde{x})$. The output is scalar and symmetric: $k\,(x, \tilde{x}) = k\,(\tilde{x}, x)$. In the literature, alternative names for the covariance function are kernel or kernel function.

The squared exponential kernel:

$$k\,(x, \tilde{x}) = exp\left(\frac{|x - \mu(x)|^2}{2\alpha^2}\right)$$

with α as length scale is a covariance function that satisfies the previous properties. The squared exponential covariance function is a widely used kernel in the machine learning field and is useful when the unknown function that GP tries to model is smooth. The hyper-parameter α is a length scale which indicates when the two inputs of the covariance function become uncorrelated.

2.4.2 Active Learning with GP

In [19], BALD was analyzed using GPC and compared to other active learning algorithms including MU. In this section, the same mathematical model and approximations as in [19] are used and repeated again for clarity.

The probabilistic model underlying GPC is as follows:

$$\begin{aligned} f &\sim GP(\mu(\cdot), k(\cdot, \cdot)) \\ y|x, f &\sim Bernoulli\left(\Phi\left(f(x)\right)\right) \end{aligned} \tag{2.32}$$

where the parameter f, is a function of a feature point $\underline{x}$ and is assigned a Gaussian process prior with mean $\mu(\cdot)$ and covariance function $k(\cdot, \cdot)$. The label y is Bernoulli distributed with probability $\Phi(f(x))$, where Φ is the Gaussian CDF.

Inference in GPC is intractable, since given a training set, the posterior over f (per feature x) becomes non-Gaussian and complicated. In the following test, Expectation Propagation (EP) [48] was used for approximating this posterior.

UAL requires the computation of $I(f; y|x, x^n, y^n)$, which can be written using the mutual information chain rule as:

$$I(f; y|x, x^n, y^n) = I(f; y|x) + I(f; y^n|x, y, x^n) - I(f; y^n|x^n) \tag{2.33}$$

Since $I(f; y|x)$ is a constant value and assuming $z^{n-1} = \{x^{n-1}, y^{n-1}\}$ is known:

$$\arg\min_{x_n} I(f; y|x, x^n, y^n) = \arg\min_{x_n} \{I(f; y_n|x, y, x^n, y^{n-1}) - I(f; y_n|x^n, y^{n-1})\} \tag{2.34}$$

We will first approximate $I(f; y_n|x^n, y^{n-1})$ using the difference between conditional entropies. Following the definition of entropy:

$$\begin{aligned} H(y_n|x_n, z^{n-1}) &= H\left(\int p\left(y_n|x_n, f_{x_n}\right) p\left(f_{x_n}|z^{n-1}\right) df_{x_n}\right) \\ &= H\left(\int \Phi\left(f_{x_n}\right) \mathcal{N}\left(f_{x_n}|\mu_{x_n, z^{n-1}}, \sigma^2_{x_n, z^{n-1}}\right) df_{x_n}\right) \end{aligned} \tag{2.35}$$

where $\mu_{x_n, z^{n-1}}$ and $\sigma^2_{x_n, z^{n-1}}$ are the mean and variance of the Gaussian approximation (using EP for example) for the posterior $p(f_{x_n}|z^{n-1})$.

Using standard results for integrals with Gaussian functions:

$$H(y_n|x_n, z^{n-1}) = H\left(\Phi\left(\frac{\mu_{x_n, z^{n-1}}}{\sqrt{\sigma^2_{x_n, z^{n-1}} + 1}}\right)\right) \tag{2.36}$$

Next, we compute:

$$\begin{aligned} H(y_n|x_n, f_{x_n}, z^{n-1}) &= \int H\left(p\left(y_n|x_n, f_{x_n}\right)\right) p\left(f_{x_n}|z^{n-1}\right) df_{x_n} \\ &= \int H\left(\Phi\left(f_{x_n}\right)\right) p\left(f_{x_n}|z^{n-1}\right) df_{x_n} \end{aligned} \tag{2.37}$$

We will use the Taylor approximation described in [19], $H\left(\Phi\left(f_x\right)\right) \approx exp\left(-\frac{f_x^2}{\pi \ln 2}\right)$ and using Gaussian integrals identities:

$$\begin{aligned} H(y_n|x_n, f_{x_n}, z^{n-1}) &\approx \int exp\left(-\frac{f_{x_n}^2}{\pi \ln 2}\right) p\left(f_{x_n}|z^{n-1}\right) df_{x_n} \\ &= \frac{C}{\sqrt{\sigma^2_{x_n,z^{n-1}} + C^2}} e^{\left(\frac{-\mu^2_{x_n,z^{n-1}}}{2\left(\sigma^2_{x_n,z^{n-1}}+C^2\right)}\right)} \end{aligned} \tag{2.38}$$

where $C = \sqrt{\frac{\pi \ln 2}{2}}$.

Therefore, we can write the mutual information as a difference between two entropies:

$$I\left(f; y_n|x_n, z^{n-1}\right) \approx H\left(\Phi\left(\frac{\mu_{x_n,z^{n-1}}}{\sqrt{\sigma^2_{x_n,z^{n-1}} + 1}}\right)\right) - \frac{C}{\sqrt{\sigma^2_{x_n,z^{n-1}} + C^2}} e^{\left(\frac{-\mu^2_{x_n,z^{n-1}}}{2\left(\sigma^2_{x_n,z^{n-1}}+C^2\right)}\right)} \tag{2.39}$$

UAL minimizes the difference between two mutual information measures as described in (2.34). The first was approximated above and we can use the same approximations for the second:

$$I\left(f; y_n|x_n, x, y, z^{n-1}\right) = \int \left(\sum_{y=-1}^{1} I\left(f; y_n|x_n, x, y, z^{n-1}\right) p(y|x, z^{n-1})\right) p(x) dx \tag{2.40}$$

where:

$$p(y|x, z^{n-1}) = \int \Phi\left(f_x\right) \mathcal{N}\left(f_x; \mu_{x,z^{n-1}}, \sigma^2_{x,z^{n-1}}\right) df_x = \Phi\left(\frac{\mu_{x,z^{n-1}}}{\sqrt{\sigma^2_{x,z^{n-1}} + 1}}\right)$$

The final UAL approximation for GPC:

$$\hat{x}_n = \arg\min_{x_n}\{\Gamma\left(x_n, z^{n-1}\right) - \Delta\left(x_n, z^{n-1}\right)\} \tag{2.41}$$

where:

$$\Delta\left(x_n, z^{n-1}\right) = H\left(\Phi\left(\frac{\mu_{x_n,z^{n-1}}}{\sqrt{\sigma^2_{x_n,z^{n-1}} + 1}}\right)\right) - \frac{C}{\sqrt{\sigma^2_{x_n,z^{n-1}} + C^2}} e^{\left(\frac{-\mu^2_{x_n,z^{n-1}}}{2\left(\sigma^2_{x_n,z^{n-1}}+C^2\right)}\right)} \tag{2.42}$$

and

$$\Gamma\left(x_n, z^{n-1}\right) \approx \sum_{x,y} \frac{1}{N}\Phi\left(\frac{\mu_{x,z^{n-1}}}{\sqrt{\sigma^2_{x,z^{n-1}} + 1}}\right)\left(H\left(\Phi\left(\frac{\mu_{x_n,D}}{\sqrt{\sigma^2_{x_n,D} + 1}}\right)\right) - \frac{C}{\sqrt{\sigma^2_{x_n,D} + C^2}} e^{\left(\frac{-\mu^2_{x_n,D}}{2\left(\sigma^2_{x_n,D}+C^2\right)}\right)}\right) \tag{2.43}$$

where we approximated the expectation over x as a sum of N random samples and $D = \{x, y, z^{n-1}\}$. This means that for each samples test point x we need to train the GPC with all possible labels and the available data.

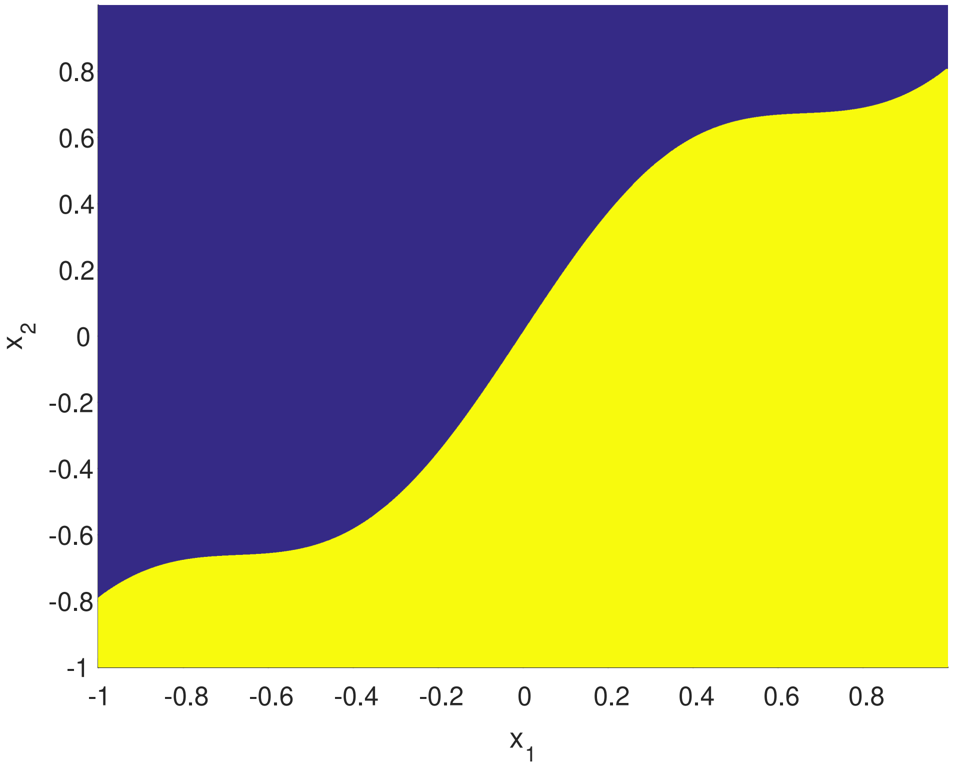


Figure 2.1: Training Set, different colors indicate the label of each feature

2.4.3 Simulation Results

In this section, we will analyze the performance of UAL for GPC under a synthetic example. The synthetic data set consists of two dimensional feature vectors with binary labels as shown in Figure 2.1, where the yellow color indicates '-1' label and blue is '+1'. In Figure 2.2, the test set is shown and is basically a smaller sub-set of the feature space. This simulates a scenario where the test is concerned with a particular region of the feature space and there is no real need to learn the whole labeling function which may be very complex and require many data points.

In practice, there may be a pre-processing stage which prunes the training set from data point which are irrelevant to the test, but this requires domain knowledge regarding the similarity between data points. The main strength of UAL, is that it implicitly takes into account the unlabelled test data to improve the resulting classifier.

The unlabelled test set is given to the learner along with an initial labelled training set and the active selection of training data starts. The labelled training data consisting of 50 random initial training data points. The active learning process is performed by adding a new data point each iteration based on the different criteria.

In Figure 2.3, for each iteration, the error probability on the test set is plotted. It can be observed that passive learning has the worst performance in terms of sample complexity given error probability. BALD and MU have comparable performance since they do not utilize the test set features and simply sample the boundary curve at multiple locations.

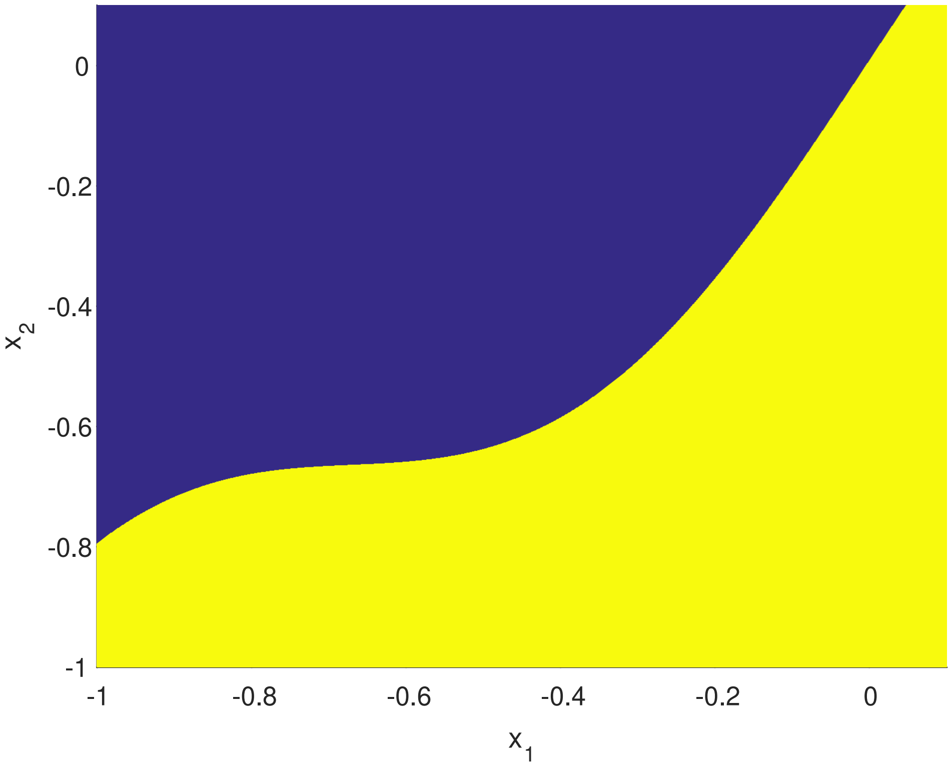


Figure 2.2: Test Set, different colors indicate the label of each feature

In Figure 2.4 one can see a large concentration of training point in the test set region since UAL takes into account the test set distribution. Also in the same figure, the contours of the predictive probability for each test point is plotted. We can see good fit to the test set as depicted in Figure 2.2.

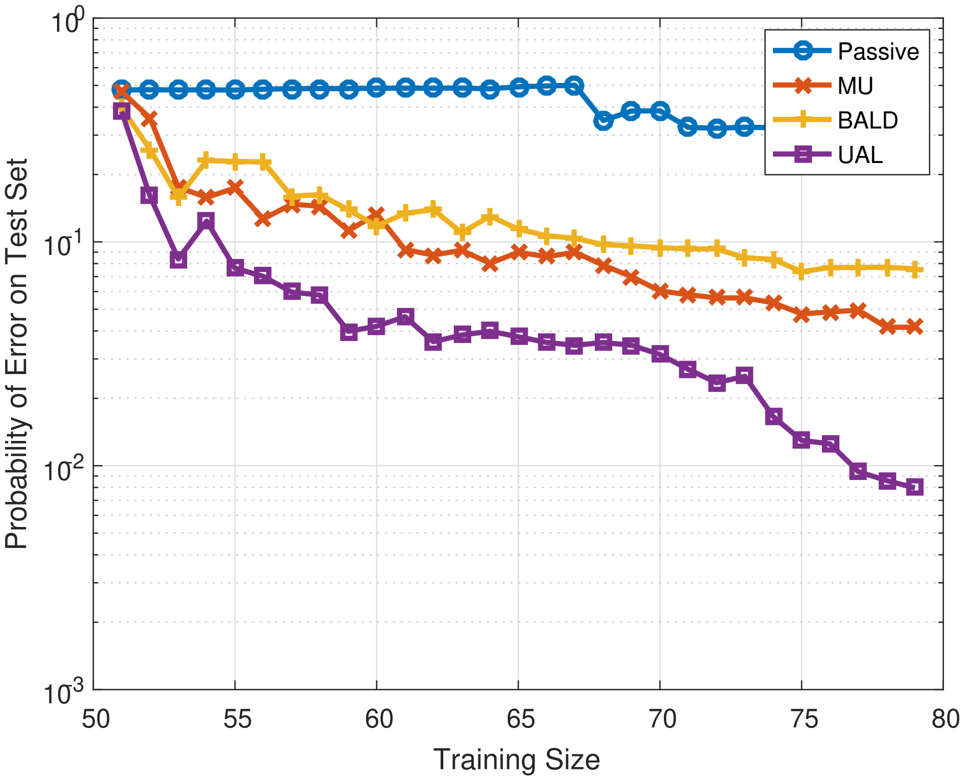


Figure 2.3: Error Probability as computed on the test set

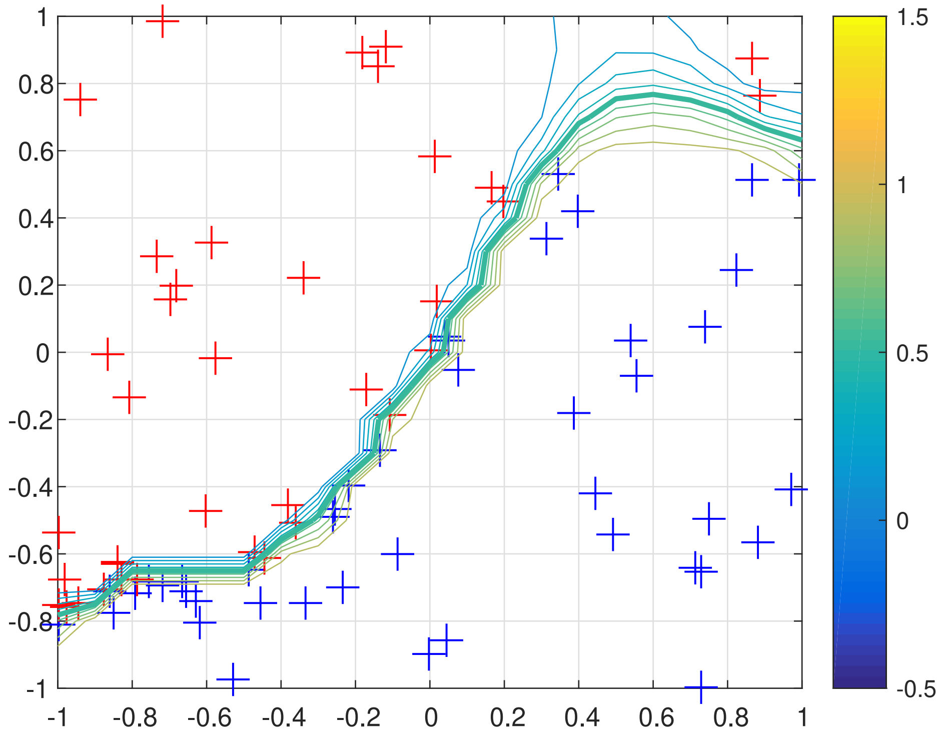


Figure 2.4: GPC predictive probability contour lines with data points acquired using UAL

Chapter 3

Linear Separators with Label Noise

In this chapter, learning half-spaces in $\mathbb{R}^n$ is considered. This learning problem is probably the most well studied for active learning with well established bounds and algorithms for different label noise models and feature priors. In [15], the authors present an algorithm which achieves near optimal sample complexity for a noiseless Oracle. The algorithm uses a margin based active learning criterion. This algorithm performs well under low noise conditions and log-concave feature distributions. In [49], an efficient Perceptron-based algorithm for active learning homogeneous half-spaces under the uniform distribution over the unit sphere was proposed. This algorithm performs well also under the bounded noise condition [50], where each label is flipped with probability at most $\eta \leq 0.5$. In [51], a margin based algorithm is presented which handles bounded noise using a polynomial regression approach for shrinking the disagreement region.

However, all these algorithms achieve good sample complexity only under log concave feature priors and symmetric binary noise models, i.e:

$$P(y = 1|x \geq \theta) = P(y = 0|x \leq \theta)$$

In this section, we would like to address a general case of noisy Oracles for learning hyper-planes. We will present a low complexity algorithm for learning half planes in $\mathbb{R}^n$ and show that the minimax regret decays exponentialy fast to zero.

The model for the noisy Oracle is based on an hypotheses class composed of a one dimensional linear separator with threshold θ_0, followed by a BAC (Binary Asymmetric Channel) with parameters (p, q), as described in Fig. (3.1). The higher dimensional linear separator is generalized accordingly. The parameters p, q are assumed in this work to be known a-priori and it could be for future work to address the joint estimation of θ, q and p using active learning.

The algorithm which will be developed in this section can handle any Discrete Memory-less Channel (DMC) noise which can be asymmetric as shown in Fig. (3.1). Also, we will not use the assumption of log-concave feature priors since our algorithm will not randomly draw features from the pool and thus eliminate the need for log-concavity of this prior. The algorithm proposed will have a polynomial computational complexity making it usable for real-world usage. Finally, the achievable performance for UAL in the linear separator hypothesis class is examined with the proposed algorithm.

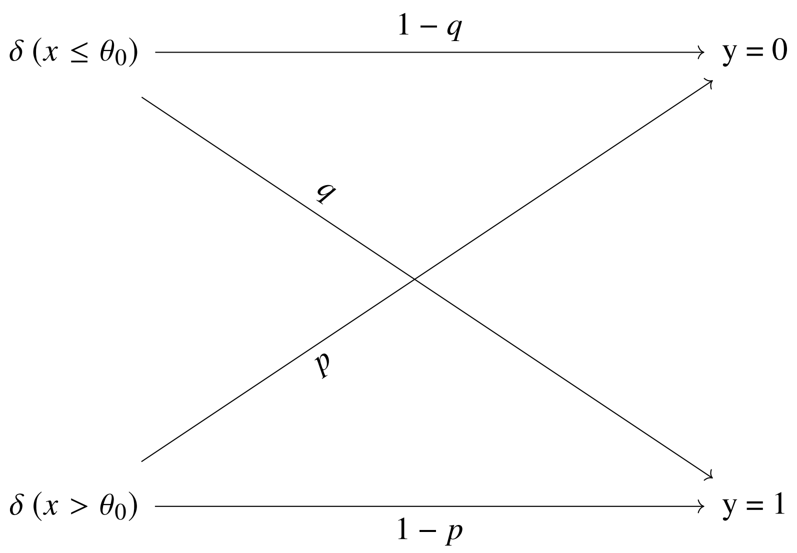


Figure 3.1: Noisy Asymmetric One Dimensional Linear Separator

3.1 Communication with Noiseless Feedback and the Posterior Matching Scheme

The problem of active learning a classifier with a noisy oracle is closely related to communication over a noisy binary channel with noiseless feedback. In this section, we will discuss this relationship and provide a short overview of Posterior Matching [52], which is a capacity achieving transmission scheme which utilizes noiseless feedback.

In the world of communications theory the problem of achieving capacity in noisy binary channels is well studied, where the underlying objective is to develop coding and decoding schemes which approach zero error probability as the block length grows. In order to achieve transmission at capacity approaching rates, one needs to develop complex channel codes and employ computationally intensive decoding algorithms. Feedback cannot increase the capacity of memoryless channels as proved by *Shannon*, but utilization of noiseless feedback can boost reliability, allow rate adaptation to cope with unknown channels and significantly simplify transmission schemes. In Figure 3.2, a general setup for communication over noisy channels via noiseless feedback is described. We note that the transmitter is described by a function of the message θ_0 and the previously received channel outputs y^{t-1}.

In [53], Horstein presented a simple feedback utilising scheme for the Binary Symmetric Channel (BSC). In that work, information is represented by a uniformly distributed message point, θ_0 over the unit interval, its binary expansion representing an infinite random binary sequence. The message point is conveyed to the receiver in an increasing resolution by always indicating whether it lies to the left or to the right of its posterior distribution's median, which is also available to the transmitter via feedback. This, in analogy to active learning, is to transmit the point which answers the most informative binary question that can be posed by the receiver based on its received information. Bits from the binary representation of the message point are decoded

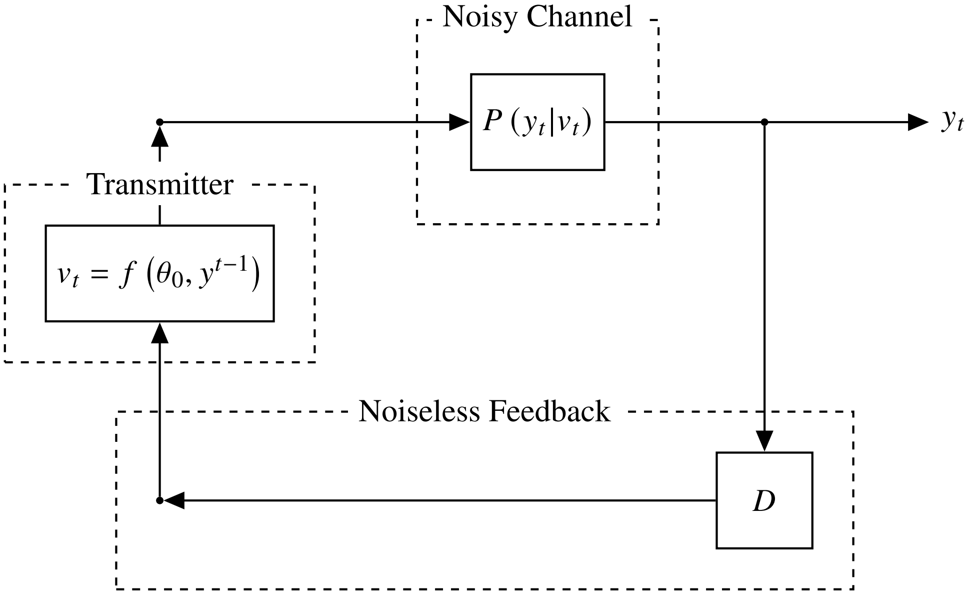


Figure 3.2: Communication over Noisy Channel with Noiseless Feedback Block Diagram

by the receiver whenever their respective intervals accumulate a sufficient posterior probability mass.

In [52], Shayevitz and Feder showed that Horstein's method is a specific instance of a more general approach which they called Posterior Matching (PM). This scheme utilizes the noiseless feedback to achieve capacity for any Discrete Memory-less Channel (DMC). The flow of PM is as follows: At each time instance, the transmitter computes the posterior distribution of the message point given the receiver's observations. According to the posterior, it "shapes" the message point into a random variable that is independent of the receiver's observations and has the desired input distribution, and transmits it over the channel. Intuitively, this random variable captures the information still missing at the receiver, described in a way that best matches the channel input. In the special cases of a BSC with uniform input distribution, PM is reduced to to Horstein's scheme. The PM scheme is defined for a channel input and output X and Y respectively with known prior and channel transition probability law: $P(x)$ and $P(Y|X)$ respectively. As with active learning, the channel output Y_t is passed to the transmitter via noiseless feedback and helps the PM scheme to generate a new channel input X_t. The receiver can then use all the received signals Y^t to generate an estimate of the message θ_0. The block diagram for PM is described in Figure 3.3 where the next channel input is given by:

$$X_{t+1} = F_X^{-1}\left(F_{\theta_0|Y^t}\left(\theta_0|Y^t\right)\right) \tag{3.1}$$

where F_X, $F_{\theta_0|Y^t}$ and θ_0 are c.d.f's and the message respectively.

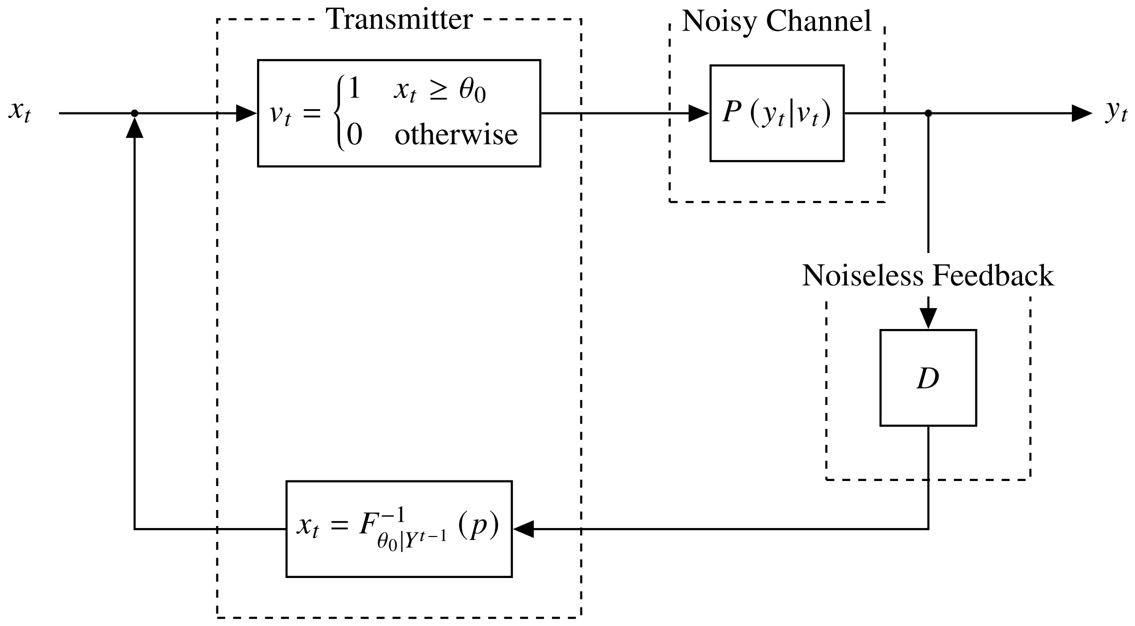

Transmitter
Noisy Channel
x_t
$v_t = \begin{cases} 1 & x_t \geq \theta_0 \\ 0 & \text{otherwise} \end{cases}$
$P(y_t|v_t)$
y_t
Noiseless Feedback
D
$x_t = F^{-1}_{\theta_0|Y^{t-1}}(p)$

Figure 3.3: Communication via Posterior Matching

3.2 One Dimensional Noisy Linear Separator

Returning to the learning problem, in Fig. 3.4 the basic flow diagram of the learning problem is shown and the dashed boxes represent the different components (Learner and Oracle). The feature x_t is selected by a selection policy ϕ based on the past training. This feature is passed through a one dimensional linear separator, generating a single bit, representing the true label associated with this feature. This label is passed through a noisy channel and this is basically the mechanism generating the training features and labels.

The flow in Fig. 3.4 can also be viewed as communicating the threshold (message) θ_0 over a noisy channel, $p(y_t|v_t)$, with noiseless feedback as shown in Fig. 3.2 [2]. Therefore, the noisy oracle can be viewed as part of the transmitter and noisy channel as denoted by the densely dotted boxes. The transmitter's output is the "clean" label bit generated by some feedback driven coding scheme. In order to have as few oracle labeling operations as possible, the objective will be that the Oracle "transmit" as much information over as few channel uses over the noisy channel as possible and correctly decode and recover θ_0. The input to the noisy Oracle can be viewed as a coding function on a message θ_0 and then transmission through the noisy channel. This is exactly the same as designing a transmission scheme which achieves capacity over this channel. Therefore, if we use the same scheme as in Figure 3.3, we will transmit θ_0 at capacity which effectively means as much information per channel use/ oracle label.

In the next theorem, it is shown that active learning based on PM (with appropriate input channel distribution) produces a selection policy such that the active learning criterion for the one dimensional threshold decays exponentially fast to zero. Moreover, this result provides an exponent for the decay of (5), which is equivalent to the *Shannon* capacity of the noisy channel (W) - C_W. The main benefit of using PM is that the noisy channel is any arbitrary DMC.

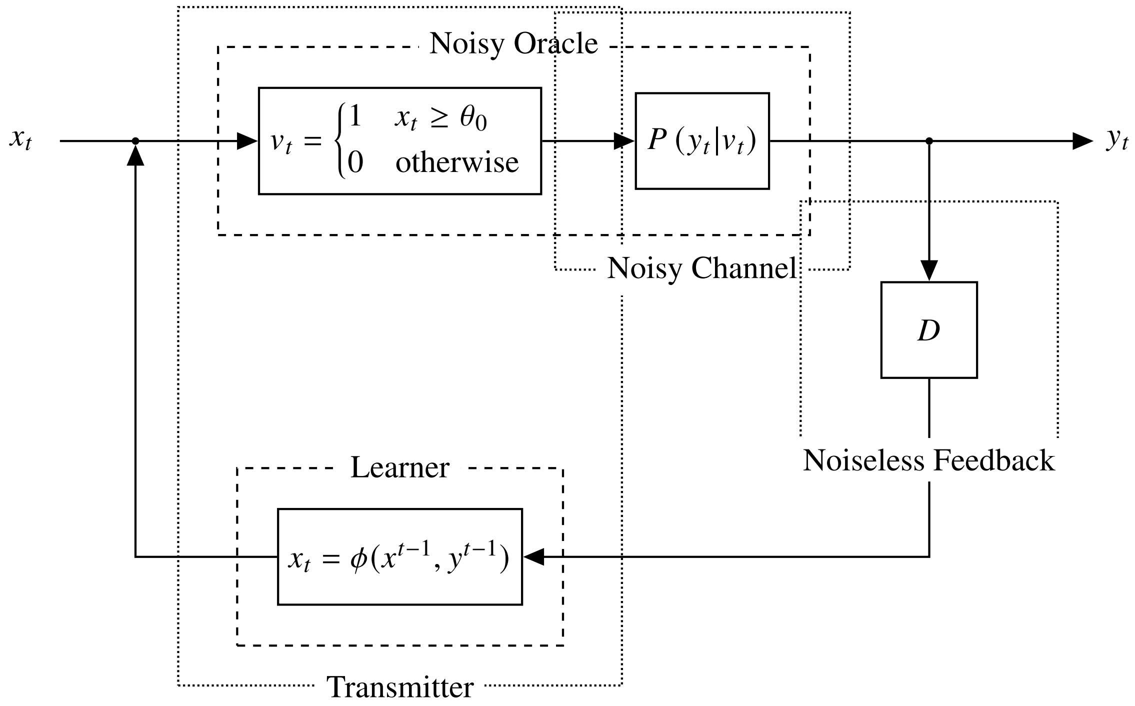

Noisy Oracle
x_t
$v_t = \begin{cases} 1 & x_t \geq \theta_0 \\ 0 & \text{otherwise} \end{cases}$
$P(y_t|v_t)$
y_t
Noisy Channel
D
Noiseless Feedback
Learner
$x_t = \phi(x^{t-1}, y^{t-1})$
Transmitter

Figure 3.4: 1-Dimensional Noisy Linear Separator Block Diagram

Theorem 4. *The 1-dimensional barrier hypotheses class is defined as:*

$$p(v|x,\theta) = \begin{cases} 1 & \text{if } x > \theta \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

where the input is x, output is v and the threshold is θ. The output, v, is the input to a Binary Asymmetric Channel (BAC) with output, y, as defined in Figure 3.5 and $\forall x \in X$ *,* $p(x) \leq \alpha$*.*

PM induced active learning produces a selection policy such that:

$$\lim_{n\to\infty} I(\theta; Y|X, x^n, y^n) = O\left(2^{-nC_W}\right)$$

where C_W *is the Shannon capacity of the BAC channel W and* $\pi(\theta)$ *is a uniform distribution on the appropriate interval.*

The proof is detailed in Appendix A.4.

Remark 5. *What happens if p and q are unknown? For the Binary Symmetric Channel (BSC), if there exists an upper bound on p, then one can transmit, in principle, at any rate below the capacity derived from this upper bound. More generally, this is proved in Theorem 8 in [54] under the discussion on Mismatch Achievability. In that theorem, Shayevitz and Feder prove that when the true channel is* $p(Y^*|X^*)$ *and induces some stationary input distribution* $p(X^*)$*. Then a scheme designed for a pair of an input distribution* $p(X)$ *and a noisy channel* $p(Y|X)$ *will have a penalty in the rate (relative to* $I(X^*;Y^*)$*) given by:* $D\left(p(Y^*|X^*)||p(Y|X)|p(X^*)\right) - D\left(p(Y^*)||p(Y)\right)$*, where D is the Kullback-Leibler divergence. Therefore, one can use PM with a mismatched prior and channel model and achieve a rate which is lower than the actual capacity of the channel.*

In Theorem 4, the prior $\pi(\theta)$ is chosen to be uniform since the convergence of PM to the correct message θ is guaranteed for a uniform prior on the messages. However, the mutual information maximizing prior $\pi^*(\theta)$ of (5) may not be uniform. In the next theorem, it is proven that when using the capacity achieving prior and the training set selected by PM (using uniform prior), UAL still decays to zero at the same exponential rate.

Theorem 5. *Given a training set* (x^n, y^n) *selected by PM using a uniform prior* $\pi_u(\theta)$ *and*

$$\pi^*(\theta) = \underset{\pi(\theta)}{\operatorname{argmax}}\, I\left(Y; \theta | X, Y^n, X^n\right)$$

Then,

$$\lim_{n\to\infty} I(\theta; Y|X, x^n, y^n) = O\left(2^{-nC_W}\right)$$

where the conditional mutual information above is computed using the prior $\pi^*(\theta)$

This theorem basically means that the uniform prior is as good as the capacity achieving prior. Theorem 4 confirms that UAL behaves similarly to other criteria in the one dimensional linear separator hypothesis class. Moreover, the decay factor for this convergence is provided, which is the *Shannon* capacity of the noisy channel. In the next section, higher dimensional linear separators will be addressed and the exponential decay of UAL will be demonstrated using a novel active learning algorithm.

3.3 Active Learning Hyper-planes via Successive Posterior Matching

In this section, a label efficient, low complexity algorithm for active learning high dimension linear separators with noisy labels under bounded prior distributions is proposed. The basic idea is to successively localize the spherical coordinates of the normal vector $\underline{w}$, representing the linear separator, using PM. This algorithm, which is denoted as Successive Posterior Matching (SPM) achieves an exponential improvement over passive learning in label complexity with the label noise channel capacity divided by the dimension as the exponent's decay coefficient.

In this setup, the features $\underline{x} \in \mathbb{R}^d$ are assumed to have a bounded feature distribution, $p(\underline{x}) \leq \alpha$, for all $\underline{x}$. The hypotheses class contains all possible homogeneous hyperplanes with normal vector $\underline{w}$. The relation between feature $\underline{x}$ and label v is defined as follows,

$$p(v|\underline{x}, \underline{w}) = \begin{cases} 1 & \text{if } \underline{w}^T\underline{x} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

However, labeling may be a noisy process and the oracle may make errors. The noisy label y, outputted by the oracle is modeled as the output of a binary asymmetric channel detailed graphically in Fig. 3.5. It is important to note here that the proposed algorithm SPM can also work for a noisy channel with $K \geq 2$ possible output labels and the binary channel is used here for simplicity purposes.

It is assumed that the parameters of the noisy channel p, q are known a-priori and can be different. SPM is detailed in Algorithm 1, where the estimations of the

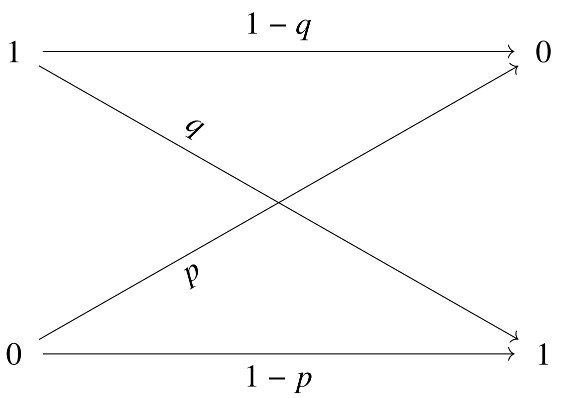


Figure 3.5: Binary Asymmetric Channel

spherical coordinates of $\underline{w}$ are denoted by $\underline{\hat{\theta}}$. In Figure 3.6, we can see an example of a two dimensional plane and its normal vector which needs to be estimated. In the initialization stage, each entry in $\underline{\hat{\theta}}$, is set to $\frac{\pi}{2}$ and its respective posterior is uniform. In Figure 3.7, we can see the uniform spherical distribution around the azimuth coordinate. In iteration i, SPM localizes the boundary between two hyper planes by querying points $\underline{x}$ with spherical coordinates fixed to $\underline{\hat{\theta}}$ and sweeping over θ_i as shown in Figure 3.8. After acquiring n training points using PM, the median of $p(\theta_i|\underline{x}^n, y^n)$ is computed. In order to generate $\hat{\theta}_i$, $\frac{\pi}{2}$ is added to the computed median to account for the fact that the normal vector needs to be estimated. This process repeats for the next angle θ_{i-1} and the uniform distribution for the elevation is shown in Figure 3.9 and the convergence to a localized boundary is shown in Figure 3.10. Note that the number of labeling operations is dn where $d+1$ and n are the dimension of the vector space and the number of labeling operations for each iteration, respectively.

In order to analyze the performance of SPM for UAL, the capacity achieving prior $\pi(\underline{\theta})$ needs to be computed. This is quite difficult and a clear analytical solution is hard to find. Therefore, a uniform prior is used and achieves close to optimal performance based on the reasoning from Theorem 5. The convergence of SPM is detailed in the following theorem:

Theorem 6. *Suppose $\underline{x} \in \mathbb{R}^{d+1}$ with a bounded p.d.f on the test feature $\forall \underline{x}$, $p(\underline{x}) \leq \alpha$. Also, assume the Oracle is some member of a d dimensional homogeneous hyper-plane hypotheses class followed by a BAC.*

Then, SPM algorithm produces a selection policy which satisfies:

$$\lim_{n\to\infty} R = \lim_{n\to\infty} I(\theta; Y|X, \underline{x}^n, y^n) = \mathcal{O}\left(2^{-\frac{n}{d}C_W}\right)$$

where n is the total number of Oracle queries and C_W is the Shannon capacity of the BAC with transition probability W.

The proof is provided in Appendix A.6.

Note that the update function in step 9 refers to a Bayesian computation of the posterior of the threshold point, based on all the observed training examples. The posterior $p(\theta_i|\underline{x}^i_{1:n}, y^i_{1:n})$ is updated at each iteration and the threshold point needs to be localized with very high accuracy. The Naïve approach would be to quantize the interval

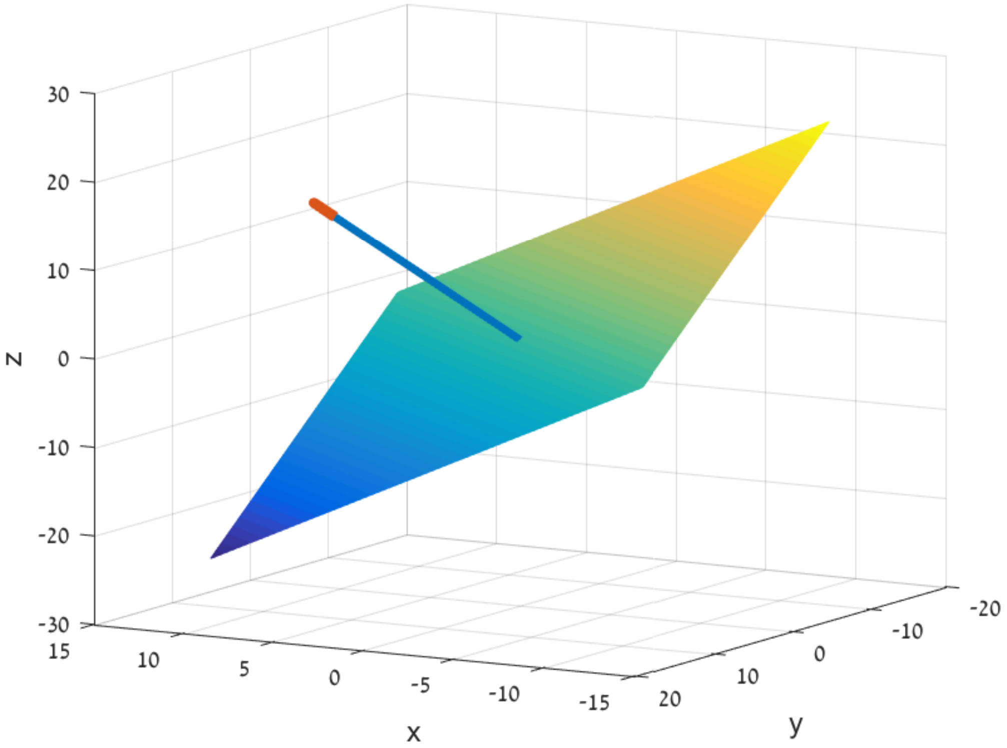


Figure 3.6: Classifier Hyperplane with Normal Vector

Algorithm 1 Active Learning via Successive Posterior Matching

```
Init: $\underline{\hat{\theta}} = [\frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2}, ..., \frac{\pi}{2}]$,
Init: $\forall i \in [1 : d-1], p(\theta_i) = Unif[0, \pi]$
for $i \leftarrow d-1$ to 1 do
    for $k \leftarrow 1$ to $n$ do
        $\hat{\theta}_i = F^{-1}_{\theta_i|\underline{x}^i_{1:k-1}, y^i_{1:k-1}}\left(\frac{p-0.5}{p+q-1}\right)$
        $\underline{x}^i_k = [\Pi_{l=1}^{d-1}\sin(\hat{\theta}_l), \cos(\hat{\theta}_{d-1})\Pi_{l=1}^{d-2}\sin(\hat{\theta}_l)$
                    $, ..., \cos(\hat{\theta}_i)\Pi_{l=1}^{i-1}\sin(\hat{\theta}_l), ..., \cos(\hat{\theta}_1)]$
        $y^i_k = Label(\underline{x}^i_k)$
        Update $p(\theta_i|\underline{x}^i_{1:k}, y^i_{1:k})$
    end for
    $\hat{\theta}_i = \hat{\theta}_i + \frac{\pi}{2}$
end for
```

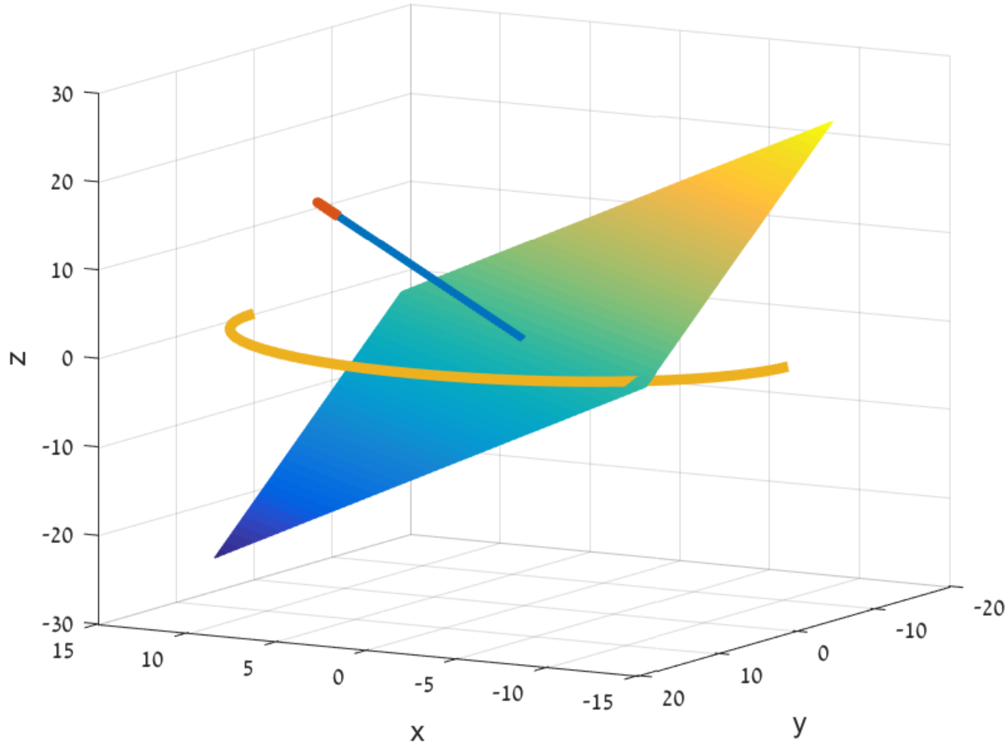


Figure 3.7: First Iteration of SPM

$[0, \pi]$ and compute the posterior for each quantiztion level. However, this approach is computationally expensive and also limited in accuracy. Since the hypothesis class is a linear separator followed by a noisy binary channel, then the posterior of the intersection angle is a multiplication of different step functions. This enables SPM to only maintain a list of the step points and update the value of the posterior between these points. Since the number of points is exactly the number of training examples, then the complexity of the calculation is proportional to n, and so the whole computational complexity of the algorithm is linear with n with no approximations taken.

3.3.1 Simulation Results

In this section, SPM is compared to a widely used passive learning algorithm for learning hyper planes - Support Vector Machine (SVM) which is known to perform very well even in noisy conditions. The comparison will be for feature spaces with $d = 200$ and $d = 500$ and using a BAC with $q = 10^{-3}$ and $p = 10^{-4}$. A Monte Carlo simulation was implemented to estimate the error probability for an active learner based on SPM and a passive learner based on SVM. In Figure 3.12, the error probabilities as a function of the total number of labels performed are presented for different space dimensions. Each test for $d = 200$ or $d = 500$ has the SVM and SPM error probabilities and also the trend line as predicted by Theorem 6 for SPM. It can be seen that the error probability decay is exponential with the decay factor related to the channel capacity divided by the degrees of freedom, which is in agreement with the theory.

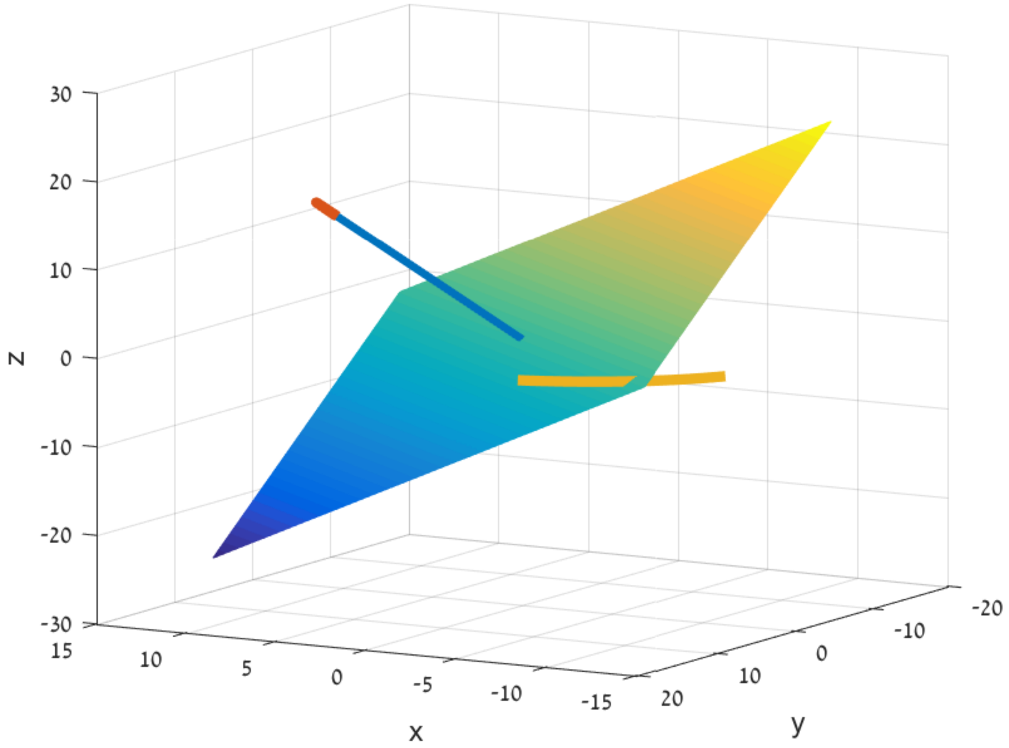


Figure 3.8: After First Iteration of SPM

In Figure 3.13, the error probabilities for $d = 200$ with different noise levels: $p = 10^{-2}$ and $p = 10^{-3}$ are plotted and it can be seen that the theory holds in these cases too.

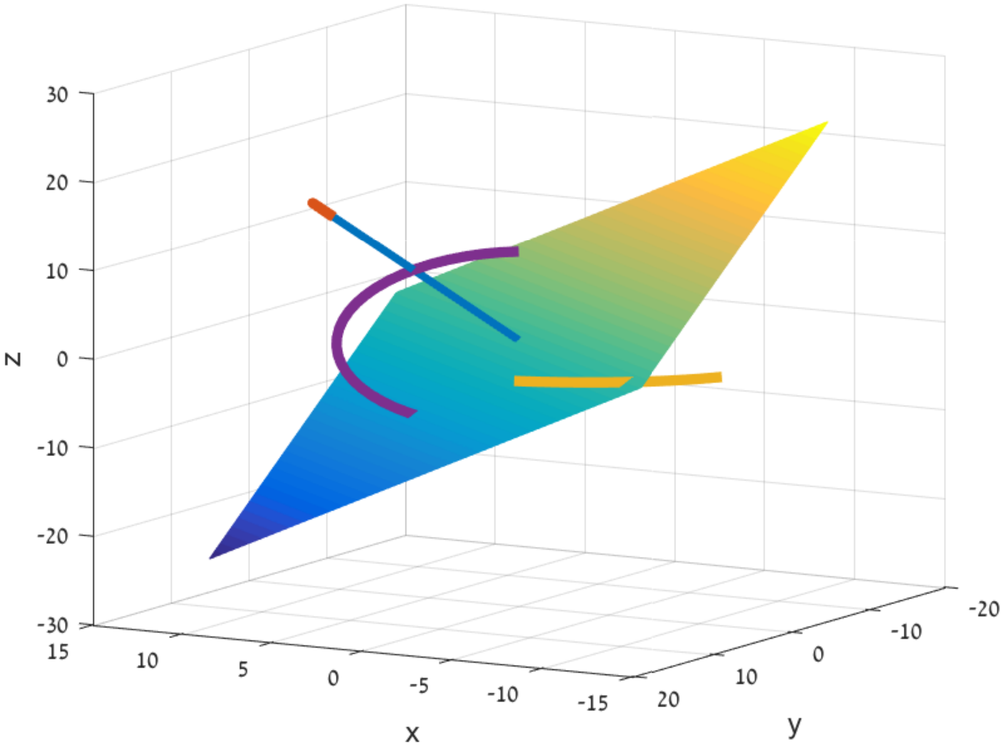


Figure 3.9: Second Iteration of SPM

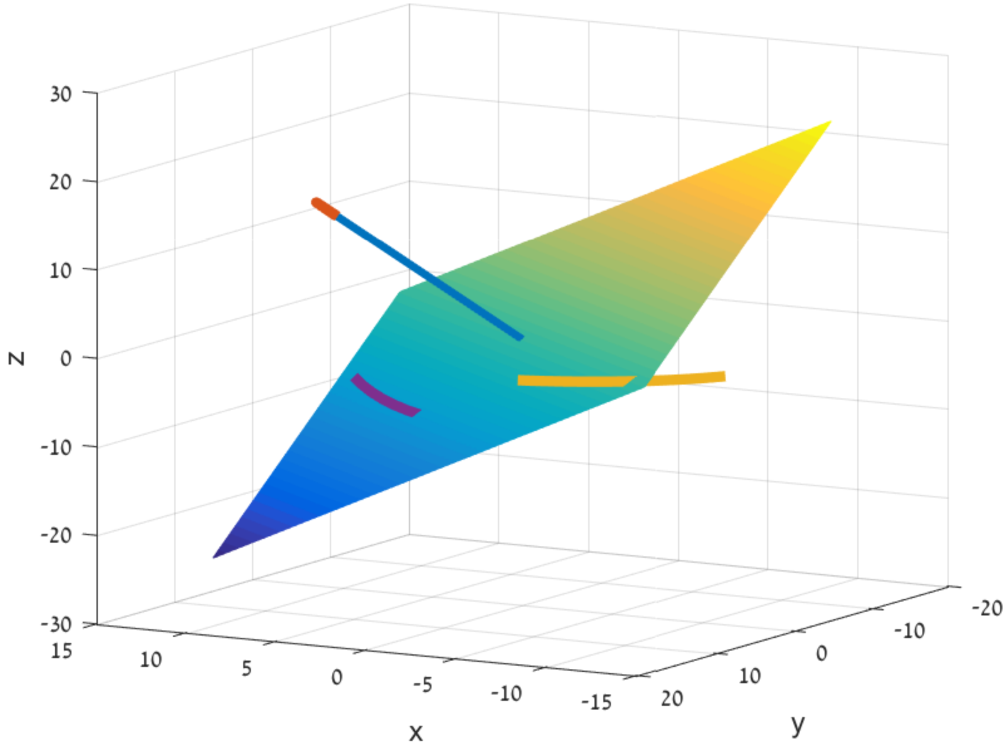


Figure 3.10: After Second Iteration of SPM

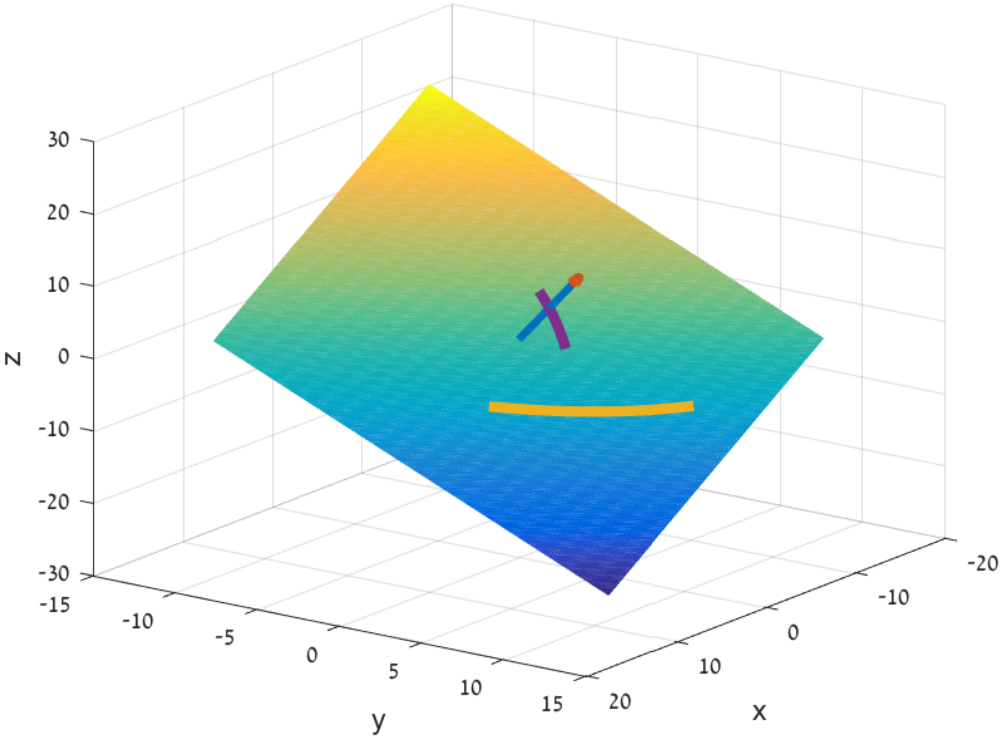


Figure 3.11: Convergence of Angular Coordinates

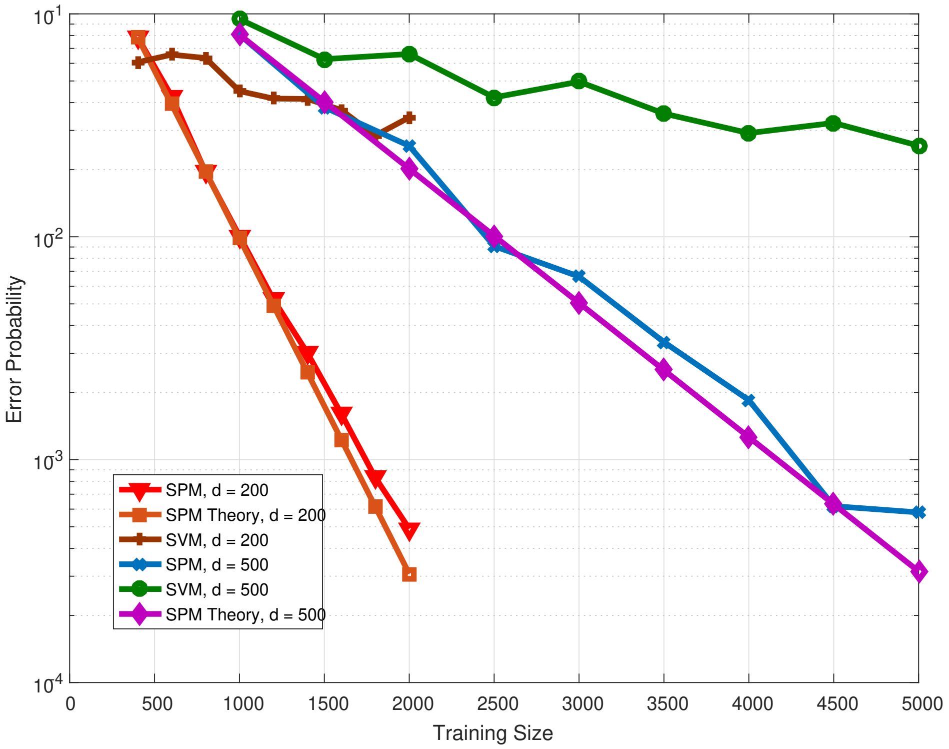


Figure 3.12: Error probability for linear separator in $\mathbb{R}^{200}$ and $\mathbb{R}^{500}$ under BAC label noise

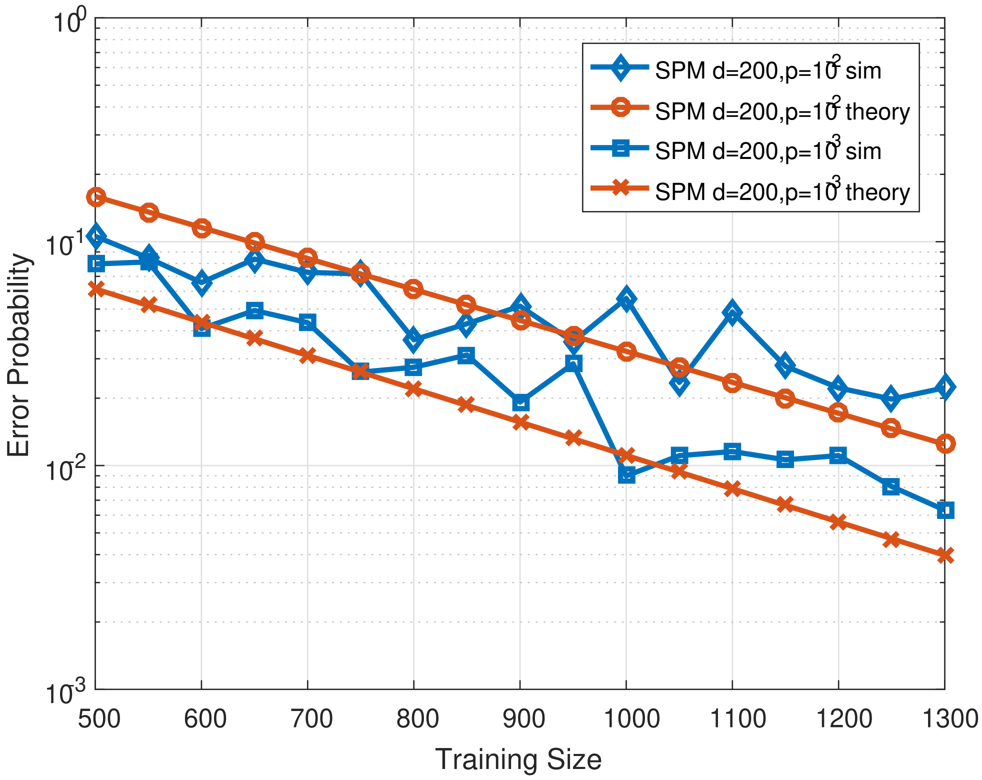


Figure 3.13: Error probability for linear separator in $\mathbb{R}^{200}$ with different noise levels

Part II

Active Learning in the Individual Setting

Chapter 4

Individual Active Learning via Predictive Maximum Likelihood Minimization

In Chapters 2 and 3, the stochastic setting was considered and an active learning criterion was proposed and analyzed. This setting has a fundamental disadvantage: it assumes that the data is generated according to a distribution which belongs to a given hypothesis class. This assumption cannot be verified on real world data thus limiting the application of UAL. In Chapter 4, active learning in the individual setting will be presented and is based on the work in [28, 30]. In this setting the data is not generated by a distribution from some parametric hypothesis class or even any distribution but an individual realization.

As an alternative to making distributional assumptions, we build upon the *individual setting* [25]. This setting does not assume any probabilistic connection between the training and test data. Moreover, the relationship between labels and data can even be determined by an adversary. The generalization error in this setting is known as the *regret* [31], which is defined as the log-loss difference between a learner and a *genie*: a learner that knows the specific test label but is constrained to use an explanation from a set of hypotheses.

The predictive Normalized Maximum Likelihood (pNML) learner [31] was proposed as the min-max solution of the regret, where the minimum is over the learner choice and the maximum is for any possible test label value. The pNML was previously developed for linear regression [55] and was evaluated empirically for DNN [56]. In section 4.1, the individual learning setting is introduced and the Predictive Normalized Maximum Likelihood (pNML) is reviewed. In section 4.2, Individual Active Learning (IAL) is proposed which is motivated by the minimax regret problem discussed in the previous section. In section 4.3, the binary classification case is analyzed. It is shown that for linearly separable data, IAL coincides with binary search. In section 4.4, IAL is analyzed for the linear regression hypothesis class and the relation to optimal design of experiments is described. Finally, in section 4.5, it is shown via simulations that IAL for Gaussian Process Classification (GPC) achieves superior performance in terms of error probability compared to passive learning, BALD, MU and UAL.

Throughout this chapter, the following notation for a sequence of samples will be used $x^n = (x_1, x_2, ..., x_n)$. The variables $x \in \mathbb{X}$ and $y \in \mathbb{Y}$ will represent the features and labels respectively with $\mathbb{X}$ and $\mathbb{Y}$ being the sets containing the features and label's alphabet respectively.

4.1 Individual Data Setting

In supervised learning, a training set consisting of n pairs of examples is provided to the learner

$$z^n = \{(x_i, y_i)\}_{i=1}^n \tag{4.1}$$

where x_i is the i-th data point and y_i is its corresponding label. The goal of a learner is to predict an unknown test label y given its test data, x, by assigning a probability distribution $q\left(\cdot|x, z^n\right)$ for each training set z^n.

In the commonly used stochastic setting, the data follows a distribution assumed to be part of some parametric family of hypotheses. A more general framework named the *individual setting* [25], does not assume that there exist some probabilistic relation between a feature x and a label y, and so the sequence $z^n = \{x^n, y^n\}$ is an individual sequence where the relation can even be set by an adversary. Since there is no distribution over the data, finding the optimal learner, $q\left(\cdot|x, z^n\right)$, is an ill-posed problem. In order to mitigate this problem, an alternative objective is proposed: find a learner $q\left(\cdot|x, z^n\right)$ which performs as well as a reference learner on the test set.

Denote Θ as a general index set. Let P_Θ be a set of conditional probability distributions

$$P_\Theta = \{p\left(y|x, \theta\right)|\theta \in \Theta\} \tag{4.2}$$

It is assumed that the reference learner knows the test label value y but is restricted to use a model from the given hypothesis set P_Θ. This reference learner then chooses a model, $\hat{\theta}\left(x, y, z^n\right)$, that attains the minimum loss over the training set and the test sample:

$$\hat{\theta} = \arg\max_{\theta\in\Theta} \left[p\left(y|x, \theta\right) w\left(\theta\right) \Pi_{i=1}^n p\left(y_i|x_i, \theta\right)\right] \tag{4.3}$$

where performance is evaluated using the log-loss function, i.e. $-\log\left(q\left(\cdot|x, z^n\right)\right)$. Note that in this work we extended the individual setting of [57] and allowed the usage of some prior $w(\theta)$ over the parameter space which may be useful for regularization purposes.

The learning problem is defined as the log-loss difference between a learner q and the reference learner (genie)

$$R_n\left(q, y; x\right) = \log \frac{p\left(y|x, \hat{\theta}\right)}{q\left(y|x, z^n\right)}. \tag{4.4}$$

Note that the reference learner has access to the true test label, y in analogy to the stochastic setting where the best learner knows the true parameter, θ in the hypothesis class. These assumptions allow the reference learner to be the best possible learner and therefore we wish to minimize the regret to it.

An important result for this setting is provided in [31] and provides a closed form expression for the minimax regret along with the optimal learner, q_{pNML}:

Theorem 7 (Fogel and Feder (2018))**.** *The universal learner, denoted as the pNML, minimizes the worst case regret:*

$$R_n(x) = \min_q \max_{y\in\mathbb{Y}} \log\left(\frac{p\left(y|x,\hat{\theta}\right)}{q\left(y|x,z^n\right)}\right)$$

The pNML probability assignment and regret are:

$$q_{pNML}(y|x,z^n) = \frac{p\left(y|x,\hat{\theta}\right)}{\sum_y p\left(y|x,\hat{\theta}\right)}$$
$$R_n(x) = \log \sum_{y\in\mathbb{Y}} p\left(y|x,\hat{\theta}\right)$$

Since the main contribution of this chapter relies on this theorem, we provide a short proof here:

Proof. We note that the regret, $R_n(x)$, is equal for all choices of y. Now, if we consider a different probability assignment, then it would assign a smaller probability for at least one of the possible outcomes. In this case, choosing one of those outcomes will lead to a higher regret and then the maximal regret will be higher, leading to a contradiction. □

The pNML regret is associated with the *stochastic complexity* of an hypothesis class as discussed by [58] and [56]. It is clear that for pNML, a model that fits almost every data pattern, would be much more complex than a model that provides a relatively good fit to a small set of data. Thus, high pNML regret indicates that the model class may be too expressive and overfit. The pNML learner is the min-max solution for supervised batch learning in the individual setting [31]. For sequential prediction it is termed the conditional normalized maximum likelihood [59, 60]. Also, note that any estimation algorithm can be used to estimate θ and the same Theorem will hold for the respective $\hat{\theta}$.

Several methods deal with obtaining the pNML learner for different hypothesis sets. [55] and [61] showed the pNML solution for linear regression. [62] proposed an NML based decision strategy for supervised classification problems and showed it attains heuristic PAC learning. [63] used the pNML for model optimization based on learning a density function by discretizing the space and fitting a distinct model for each value.

4.2 Active learning for individual data

In active learning, the learner sequentially selects data instances x_i based on some criterion and produces n training examples z^n. The objective is to select a subset of the unlabelled pool and derive a probabilistic learner $q\left(y|x,z^n\right)$ that attains the minimal prediction error (on the test set) among all training sets of the same size. Most selection criteria are based on uncertainty quantification of data instances to quantify their informativeness. However, in the individual setting, there is no natural uncertainty measure since there is no distribution governing the data.

We propose to use the min-max regret R_n as defined in Theorem 7 as an active learning criterion which essentially quantifies the prediction performance of the training set z^n for a given unlabeled test feature x. A "good" z^n minimizes the min-max regret for any test feature and thus provides good test set performance. Since R_n is a point wise quantity, we suggest to look at the average over all test data.

We propose the following criterion:

$$C_n = \min_{x^n} \max_{y^n} \sum_x \left(\sum_y p\left(y|x, \hat{\theta}\right) \right) \tag{4.5}$$

where $\hat{\theta} = \hat{\theta}\left(x, y, z^n\right)$. The idea is to find a set of training points, x^n that minimizes the averaged log normalization factor (across unlabeled test points), for the worst possible labels y^n. This criterion looks for the worst case scenario since there is no assumption on the data distribution and we assume individual sequences.

Since (4.5) is difficult to solve for a general hypothesis class, we define a greedy form which we denote as Individual Active Learning (IAL):

$$C_{n|n-1} = \min_{x_n} \max_{y_n} \sum_x \left(\sum_y p\left(y|x, \hat{\theta}\right) \right) \tag{4.6}$$

Note that when computing (4.6), the previously labeled training set, z^{n-1}, is assumed available for the learner and $\hat{\theta} = \hat{\theta}\left(x, y, x_n, y_n, z^{n-1}\right)$. The objective in (4.6) is to find a single point x_n from the unlabelled pool as opposed to the objective in (4.5) that tries to find an optimal batch x^n.

Note that we could have also defined IAL as the average over the regret values directly and not the normalization factors:

$$R_{n|n-1} = \min_{x_n} \max_{y_n} \sum_x \log \left(\sum_y p\left(y|x, \hat{\theta}\right) \right) \tag{4.7}$$

Due to Jensen, $R_{n|n-1} \leq C_{n|n-1}$ and so the optimal point x_n based on average normalizing factors will also provide a low average regret. For Chapter 4, IAL as defined in (4.6) will be used and in Chapter 5, IAL in (4.7) will be used.

In the next sections, we will analyze the performance of IAL on linear regression and binary classification. First, we will prove that IAL coincides with commonly used criteria for linear binary separators and linear experimental design. These results suggest IAL can be viewed as a unified active learning approach for general hypothesis classes. Finally, we will derive IAL for Gaussian Process Classification (GPC) and analyze the performance on real data.

4.3 Binary Classification with Separable Data

This section presents an analysis of the performance of IAL applied to one dimensional linear binary classifiers. When dealing with a 1-dimensional barrier, the margin based active learning, also known as binary search, achieves optimal sample complexity [7, 20, 64, 65]. We verify the optimality of our criterion by showing that, for a 1-dimensional barrier, it indeed produces the same result as binary search. To support this claim, we present a theorem proving the equivalence between IAL and binary search in the simple case of a 1-dimensional barrier. This result confirms that IAL is a viable active learning criterion.

The 1-dimensional barrier hypotheses class is defined as:

$$p(y=1|x,\theta)=\begin{cases}\alpha & \text{if } x>\theta\\ 1-\alpha & \text{otherwise}\end{cases} \tag{4.8}$$

where $\alpha\in\{0,1\}$, the input is $x\in[0,1]$, output is $y\in\{0,1\}$ and the unknown threshold is $\theta\in[0,1]$.

Theorem 8. *For 1 dimensional linearly separable data, IAL induces a selection policy which coincides with binary search.*

The full proof is in appendix B.1 but we provide a sketch here:

Proof sketch. The greedy criterion defined in (4.6) can be written as

$$C_{n|n-1}=\min_{x_n\in\mathbb{X}}\max_{y_n\in\mathbb{Y}}\int_{x\in\mathbb{X}}\log\sum_{y\in\mathbb{Y}}p\left(y|x,\hat{\theta}^n\right)dx \tag{4.9}$$

where y, x and $\hat{\theta}^n$ are the test label, feature and maximum likelihood estimation based on training and test data respectively

$$\hat{\theta}^n=\arg\max_{\theta\in\Theta}p\left(y^n,y|x^n,x,\theta\right). \tag{4.10}$$

We can write the likelihood for z^{n-1} as

$$p\left(y^{n-1}|x^{n-1},\theta\right)\sim\mathbb{1}\left(\theta\geq\theta_{min}^{n-1}\right)\mathbb{1}\left(\theta<\theta_{max}^{n-1}\right) \tag{4.11}$$

where θ_{min}^{n-1} and θ_{max}^{n-1} represent the support of the posterior on θ given x^{n-1},y^{n-1}.

For each unlabelled pool point x_n, the updated likelihood window function gets split based on y_n.

For $y_n=1-\alpha$:

$$\int_0^1\log\sum_{y=0}^{1}p\left(y|x,\hat{\theta}^n\right)dx=|x_n-\theta_{max}^{n-1}| \tag{4.12}$$

and for $y_n=\alpha$:

$$\int_0^1\log\sum_{y=0}^{1}p\left(y|x,\hat{\theta}^n\right)dx=|\theta_{min}^{n-1}-x_n|. \tag{4.13}$$

Therefore,

$$C_{n|n-1}=\min_{x_n\in\mathbb{X}}\max\{|x_n-\theta_{max}^{n-1}|,|\theta_{min}^{n-1}-x_n|\}. \tag{4.14}$$

The point x_n which minimizes the maximal length is the mid point of the interval $\left[\theta_{min}^{n-1},\theta_{max}^{n-1}\right]$. □

Notice that the fact that IAL and binary search are identical for this hypothesis class does not mean they are identical for general hypothesis classes. The main advantage of IAL is that it adapts to different hypothesis classes and is a general active learning criterion.

4.4 Linear Regression

In this section, active learning for linear regression in the individual setting is considered. The goal is to pick a small number of vectors from the space of possible features so that the resulting learner, $q(y|x, z^n)$ performs well in some sense. It will be shown that IAL for this case coincides with a commonly used criterion, thus further demonstrating IAL is a unified framework for active learning.

We remind the linear regression model:

$$\underline{y} = X\underline{\theta} + \underline{z} \tag{4.15}$$

where $X \in \mathbb{R}^{nxp}$, $\underline{\theta} \in \mathbb{R}^p$, $\underline{y} \in \mathbb{R}^n$ and $\underline{z} \in \mathbb{R}^n$ are the design matrix, model vector, vector of observable responses and i.i.d Gaussian noise vector with zero mean and finite variance $\sigma_Z^2 I$.

The Maximum Likelihood estimator $\hat{\underline{\theta}} = \left(X^T X\right)^{-1} X^T \underline{y}$, known as Ordinary Least Squares (OLS), has the property that the error covariance matrix depends on neither the true parameter vector $\underline{\theta}$ nor the observed response $\underline{y}$. This suggests that we can "optimize" the covariance of the estimator a-priori, even before taking any measurements, transforming the problem from interactive querying an oracle to subset selection of feature vectors. In experimental design, a small subset $S \subset \{1, ..., n\}$ of column vectors are selected from X thus generating a smaller design matrix X_S. Using X_S, one can derive the OLS solution for the parameter vector $\underline{\theta}$. As described before, the design problem reduces to minimizing the covariance matrix $\Sigma^{-1} = \left(X_S^T X_S\right)^{-1}$.

In this section, IAL in batch form (4.5) is applied to the linear regression problem. The following theorem states that IAL coincides with V optimal design, which minimizes average prediction variance. Other designs are described in [37]. Note that the values of the pNML normalization factor used for scoring IAL, may be too conservative when the model class is very expressive. Therefore, using large model classes can result in over-fitting the query point arbitrarily well to any label. Therefore, we propose to control the expressivity of the model class by regularization in the form of a prior on the model parameter, $p\left(\theta\right)$. The selected θ will maximize both data likelihood and a regularization term, or prior, over parameters. In our case we will opt for a Gaussian prior with scale factor $\lambda > 0$.

Theorem 9. *Consider the hypothesis class:*

$$P_\Theta = \{p\left(y|x, \theta\right) | \theta \in \mathbb{R}^d\}$$

$$p\left(y|x, \theta\right) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}\left(y - x^T\theta\right)^2\right)$$

where $x \in \mathbb{R}^d$ and σ^2 is known a-priori.

Then IAL is equivalent to the following subset selection problem:

$$C_n = \min_{X_n} \mathrm{Tr}\left(X^T X \left(X_n^T X_n + \frac{\sigma^2}{\lambda} I\right)^{-1}\right)$$

where X and X_n are the concatenation of the test vectors and the concatenation of the training vectors respectively. Also, the estimation $\hat{\theta}$ is computed using L2 regularization with a factor λ. This factor can also be viewed as a Gaussian prior on the model vector θ.

Note that IAL is a function of the training features x^n only and have no dependence on their respective labels y^n. Therefore, there is no real need for online feedback in the active linear regression problem and the training set problem can be cast as a subset selection problem performed offline. This problem is NP hard and thus approximate solutions are needed.

This result further demonstrates that IAL can be viewed as a unified framework for active learning in a variety of hypothesis classes. Since V design is not sub-modular, then this theorem acts a counter example to IAL being a sub-modular criterion in general. Note that also UAL in [27], coincided with V-optimal but and the Gaussian prior was derived as a maximizing prior for the constraint on the family of possible priors. In the individual case, it is part of the MAP estimation of the optimal model $\hat{\theta}$.

4.5 Gaussian Process Classification

In this section, we will analyze IAL for Gaussian Process Classification (GPC). GPC is a powerful, non-parametric kernel-based model that poses a challenging problem for information-theoretic active learning since the parameter space is infinite dimensional and the posterior distribution is analytically intractable. A detailed introduction to GPC can be found in [40].

In [19], BALD was analyzed for GPC and compared to other active learning algorithms including MU. In [27], UAL was analyzed for GPC and was shown to perform well when given access to the un-labelled test set. In this section, we use the mathematical model of [19] which is repeated here for clarity.

The probabilistic model underlying GPC is as follows:

$$\begin{aligned} f &\sim GP(\mu(\cdot), k(\cdot,\cdot)) \\ y|x, f &\sim Bernoulli\left(\Phi\left(f_x\right)\right) \end{aligned} \tag{4.16}$$

where the parameter f is a function of a feature point x and is assigned a Gaussian process prior with mean $\mu(\cdot)$ and covariance function $k(\cdot,\cdot)$. The label y is Bernoulli distributed with probability $\Phi(f_x)$, where Φ is the Gaussian CDF and f_x is a function of f and x, for example an inner product.

The conditional distribution for the labels is similar to logistic regression due to the fact that a function of the weights and data point is passed through an activation function. However, the main advantage of GPC is in the statistical structure of the latent vector $\underline{f}$, which is induced by the Gaussian prior. Unlike logistic regression where the weights may change per data point, GPC introduces a correlation between the weights based on the correlation between the data points.

Without any prior, pNML will give over confident scores for models with very high degrees of freedom. In [56], logistic regression was investigated and a regularization prior was introduced. In GPC, a regularization prior is part of the model as the Gaussian process on f! Therefore, this prior will limit the possible solutions of the hypothesis class and avoid over-fitting.

For GPC, (4.6) can be written as:

$$C_{n|n-1} = \min_{x_n \in \mathbb{X}} \max_{y_n \in \mathbb{Y}} \sum_{v \in \mathbb{V}} \int_{u \in \mathbb{U}} p\left(v|\hat{f}_u\right) du \tag{4.17}$$

The maximum likelihood estimate, $\hat{f}_u$ (for test point u) is based on training and test data. Again, note that the conditional probability of v depends on the random variable f_u which is a function of f and u.

The MAP estimation for the model parameter vector, $\underline{f}$ (for all possible feature points):

$$\hat{\underline{f}} = \arg\max_{\underline{f}} p\left(y^n, v|x^n, u, \underline{f}\right) p(\underline{f}) \tag{4.18}$$

where $p(\underline{f})$ is the Gaussian process introduced in (4.16), which acts as a regularization prior over the latent vector $\underline{f}$.

Exact inference in GPC is intractable, since given a training set, the likelihood $p\left(y^n, v|x^n, u, \underline{f}\right)$, becomes non-Gaussian and complicated. In order to compute IAL in this case, we approximate the posterior as a Gaussian on the latent model $\underline{f}$ as described in chapter 3 in [46]. The basic idea in this approximation is to model the posterior:

$$p(\underline{f}|x^{n-1}, y^{n-1}) \approx \mathcal{N}\left(\underline{f}; m, K\right) \tag{4.19}$$

where m and K are the mean and covariance given $\left(x^{n-1}, y^{n-1}\right)$.

Once a new data point is added with its corresponding label (x_n, y_n), use Bayesian updating to incorporate the new data point and apply a variational approximation to model the posterior as a Gaussian distribution:

$$p\left(\underline{f}|x^n, y^n\right) \approx \mathcal{N}\left(\underline{f}; \mu, V\right) \tag{4.20}$$

where μ and V are the mean and covariance of the variational approximation of the posterior given (x^n, y^n).

For the binary case we can write:

$$C_{n|n-1} = \min_{x_n \in \mathbb{X}} \max_{y_n \in \mathbb{Y}} \int_{u \in \mathbb{U}} \left(\Phi\left(\hat{f}_u^{v=1}\right) + \left(\Phi\left(-\hat{f}_u^{v=-1}\right)\right)\right) du \tag{4.21}$$

where $\hat{f}_u^{v=1}$ and $\hat{f}_u^{v=-1}$ are the maximum likelihood estimates of the latent parameter f_u for test point u with corresponding label v.

We observe that in order to find the next data point x_n, we do not have to approximate the Gaussian posterior over all $\underline{f}$ but only on points related to the features. Ideally, we would use all the data $[u, v, x^n, \overline{y}^n]$ to approximate a Gaussian posterior and select the maximum point. However, since we are looking at all possible labels for all possible training and test points, we need to use another approximation. We propose to use (4.19) as a prior for the EP approximation and update the Gaussian approximation using the test and candidate training points only and not z^{n-1}. The mean of the resulting Gaussian will be used for the IAL computation.

$$p\left(f_{x_n}, f_u | D^{n-1}, x_n, v, u_k, l\right) \approx \mathcal{N}\left(f_u, f_{x_n} | \hat{\mu}, \hat{V}\right)$$

where $\hat{\mu}$ and $\hat{V}$ are the mean and covariance of the variational approximation of the posterior given (x_n, u, y_n, v) and (4.19) as a prior.

The resulting IAL is summarized in Algorithm 3. First, the algorithm uses an approximate inference method to compute a Gaussian approximation for the posterior using the available training set. Next, for each training point, all possible labels are examined along with a sweep on the test set with all possible labels. We run MAP estimation for all the different configurations of training and test and recover the MAP estimate for the test points. We accumulate the probability of the test label given these estimations (pNML regrets). Finally, we find the training point, for which the worst case regret is minimal over the sum of the test points. For all subsequent tests, Expectation Propagation (EP) [48] was used for approximating this posterior using the GPML toolbox [66].

Algorithm 2 Individual Active Learning

```
Input: Training Data $\{x^{n-1}, y^{n-1}\}$
Training and Test samples $\{x_i\}_{i=1}^{N}$ and $\{u_i\}_{i=1}^{K}$.
Output: Next data point for labelling - $x_n$ IAL - GPC
Set $D = [x^{n-1}, y^{n-1}]$
Set EP prior $q_{prior}^{EP} = \mathcal{N}\left(\underline{f} | 0, \log \lambda I\right)$
Run EP: $q^{n-1}\left(\underline{f}\right) = EP(D, q_{prior}^{EP})$
$\mathbf{S} = zeros(N, |\mathbb{Y}|)$
for $i \leftarrow 1$ to $N$ do
    for $j \in \mathbb{Y}$ do
        for $k \leftarrow 1$ to $K$ do
            for $l \in \mathbb{Y}$ do
                Set $D = [x_i, j, u_k, l]$
                Set EP prior $q_{prior}^{EP} = q^{n-1}\left(\underline{f}\right)$
                $\mathcal{N}\left(f_{u_k}, f_{x_i} | \hat{\mu}, \hat{V}\right) = EP(D, q_{prior}^{EP})$
                $\hat{f}_{u_k}^{l}, \hat{f}_{x_i}^{j} = \hat{\mu}$
                $\mathbf{S}(i, j) = \mathbf{S}(i, j) + \Phi\left(l \cdot \hat{f}_{u_k}^{l}\right)$
            end for
        end for
    end for
end for
$\hat{i} = \operatorname{argmin}_i \max_j \mathbf{S}$
$x_n = x_{\hat{i}}$
```

Table 4.1: Algorithms Parameters

Parameter	**Value**
Passive Regularization λ	5
MU Regularization λ	5
BALD Regularization λ	5
UAL Regularization λ	5
IAL Regularization λ	5
Initial training set	2 examples (1 for each class)
Unlabelled test set	5 random test features

4.5.1 Simulation Results

In this section, we conduct an empirical comparison between Individual Active Learning (IAL) and alternative active learning methodologies, employing Gaussian Process Classification (GPC) on both synthetic and real datasets. Throughout this analysis, we aim to identify the specific conditions under which IAL demonstrates superior performance compared to all other active learning schemes, including Universal Active Learning (UAL), which also considers test distribution, as well as instances where IAL's performance aligns with other active learning schemes. The findings reveal that IAL excels, particularly in scenarios involving Out-of-Distribution cases and when confronted with extremely limited initial training datasets.

In Table 4.1, the parameters for the different algorithms in the subsequent experiments are detailed. The regularization factor λ of the Gaussian kernel as defined in Algorithm 3, is detailed for each selection scheme. Also, the initial training set size and the number of unlabelled test features used for IAL and UAL are detailed.

The synthetic data set consists of a two dimensional feature space with binary labels. The training pool is a square in the two dimensional plane with corners at four points (-1, -1), (1, -1), (-1, 1), and (1, 1) and divides them to two non overlapping regions, as shown in Figure 4.1, where the two labels are encoded to two colors. The test set is a smaller sub-set with corners at four points (-1, -0.5), (1, -0.5), (-1, -0.25), and (1, -0.25). This simulates a scenario where the test set is concentrated in a particular region of the feature space and there is no real need to learn the whole labeling function which may be very complex and require many data points for learning. In practice, there may be a pre-processing stage which prunes the training set from data points which are irrelevant to the test, but this usually requires domain knowledge which may be unavailable in real world applications. Also, as mentioned before, this pruning stage will increase the sample complexity since it requires labelling.

In Figure 4.2, we have conducted a comparison of Passive, MU, BALD, UAL, and IAL in terms of error rates per training set size. The results indicate that UAL and IAL exhibit similar and significantly better performance compared to BALD, MU, and Passive selections. The effectiveness of UAL and IAL stems from their ability to learn the labeling function within the relevant regions of the feature space, optimizing the use of the labeling budget by focusing on areas pertinent to the test set.

Furthermore, the similar performance of UAL and IAL in this context can be attributed to the hypothesis class (GPC) effectively modeling the data distribution.

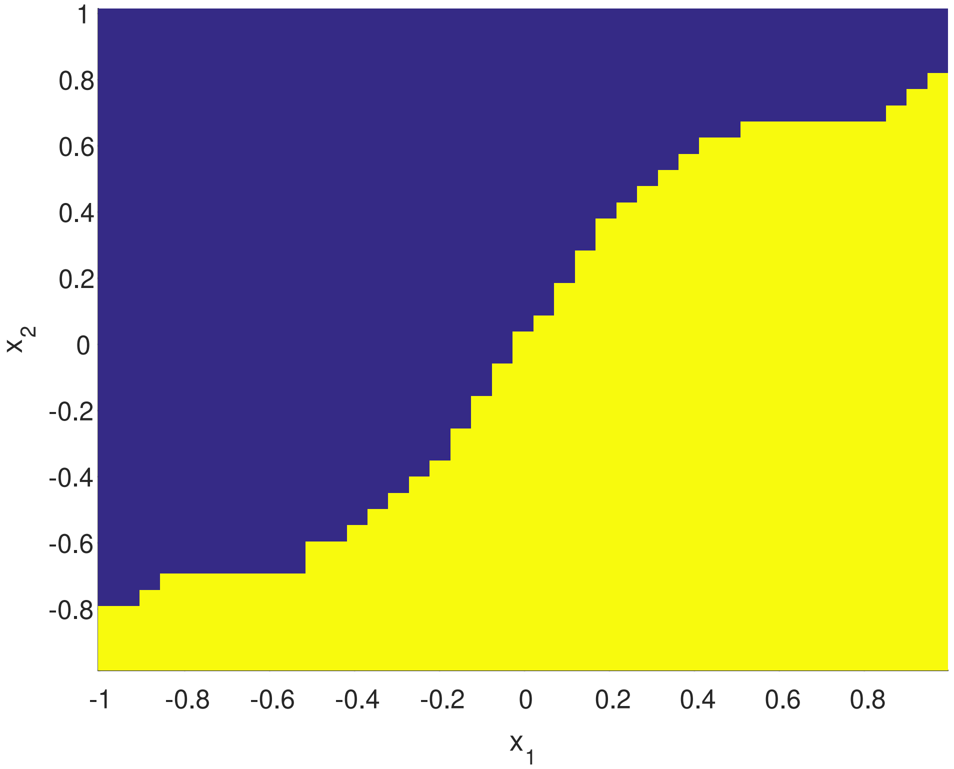


Figure 4.1: Synthetic Data Set

The Gaussian kernel approximates the separating curve well, aligning with the true distribution within the hypothesis class. Therefore, in this valid stochastic setting, there is no substantial difference between IAL and UAL. Notably, UAL gains an advantage with larger training sets, as IAL considers the worst-case scenario while UAL leverages its posterior, becoming more accurate.

It is important to highlight that when the test set matches the training set, both UAL and IAL do not exhibit a clear advantage over MU and BALD. This scenario, known as In-Distribution (IND), is characterized by a distinct boundary between classes, and the data aligns with a specific hypothesis class.

In summary, this synthetic example underscores that when the stochastic setting is valid, UAL and IAL hold a distinct advantage over active learning schemes that overlook the (unlabeled) test set.

In the next scenario, IAL is compared to UAL, BALD, MU and Passive learning in an empirical analysis using Gaussian Process Classification (GPC) over a real data set. The difference from the synthetic case is that now the test and train distributions do not necessarily belong to the GPC hypothesis class and the stochastic setting can no longer be verified, thus we expect to see an advantage for IAL over UAL in this case.

The dataset is the USPS hand-written digits data set [67]. There are a total of 9298 handwritten single digits between 0 and 9, each of which consists of 16×16 pixel image. Half of 9298 digits are designated as the training set and the other half are the test test (labels are not known by the learner). Pixel values are normalized to be in the range of [-1, 1]. Each feature has dimension 256 which requires significant computational

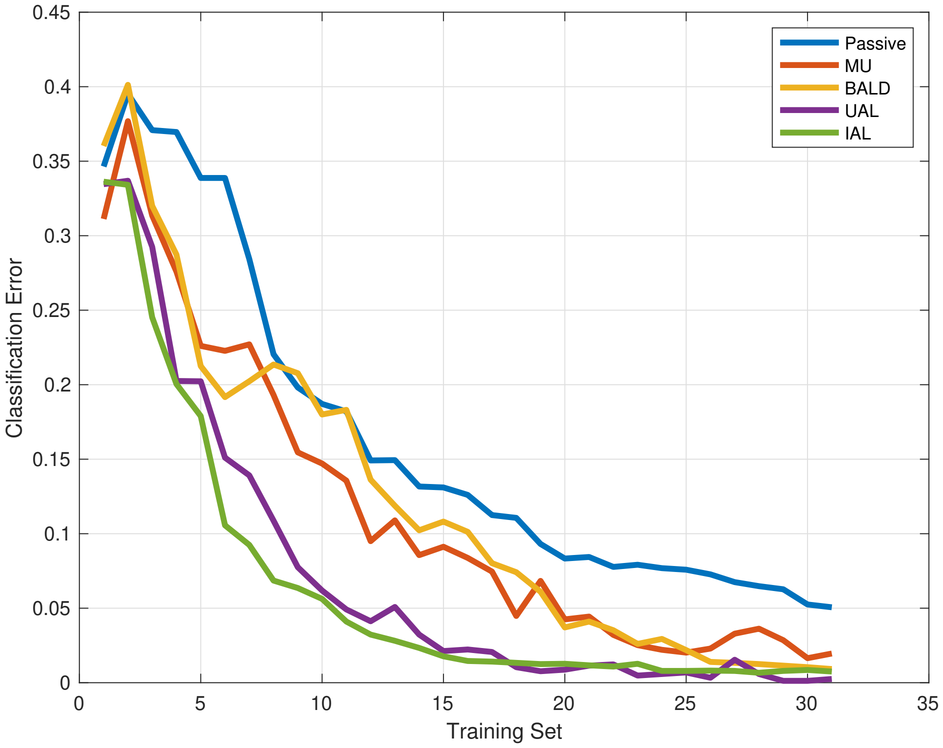


Figure 4.2: Error Probability: Synthetic Data Set

resources for the EP approximation. In order to reduce the dimension of the data space, PCA (Principal Component Analysis) is applied using the entire un-labelled training data. After centering and PCA, the eigen-vectors corresponding to the 65% largest eigen-values of the PCA are used as the feature space for classification. A small random subset of the unlabeled test set is given to the learner and a small random initial example per class (bootstrap the GPC learner). Active learning is performed by adding a new data point each iteration based on the different criteria.

The objective is to classify the digit 7 versus 9. We chose these two digits since they are graphically similar and thus we expect it will be hard to separate between them, so the boundary will not be very clear. For example, distinguishing between 0 and 1 proves to be a very easy learning task and with very few examples, the GPC learns the optimal separator, so there is no benefit for using active learning for this task.

The initial scenario under scrutiny involves both the test and training sets comprising solely images of the digits 7 and 9, denoted as In-Distribution (IND). From a practical standpoint, this scenario is deemed less compelling since active learning typically grapples with datasets containing Out-Of-Distribution (OOD) samples, with the primary goal being to minimize human intervention. Nonetheless, from a theoretical perspective, it is intriguing to observe that IAL holds its ground against other methods in this context.

In Figure 4.3, the classification error probability is graphed, revealing a comparable performance among UAL, MU, and BALD, while IAL exhibits a slight superiority in the small training set regime. This can be attributed to the invalidation of the stochastic assumption, where UAL, BALD and MU lack a guarantee of optimal performance and

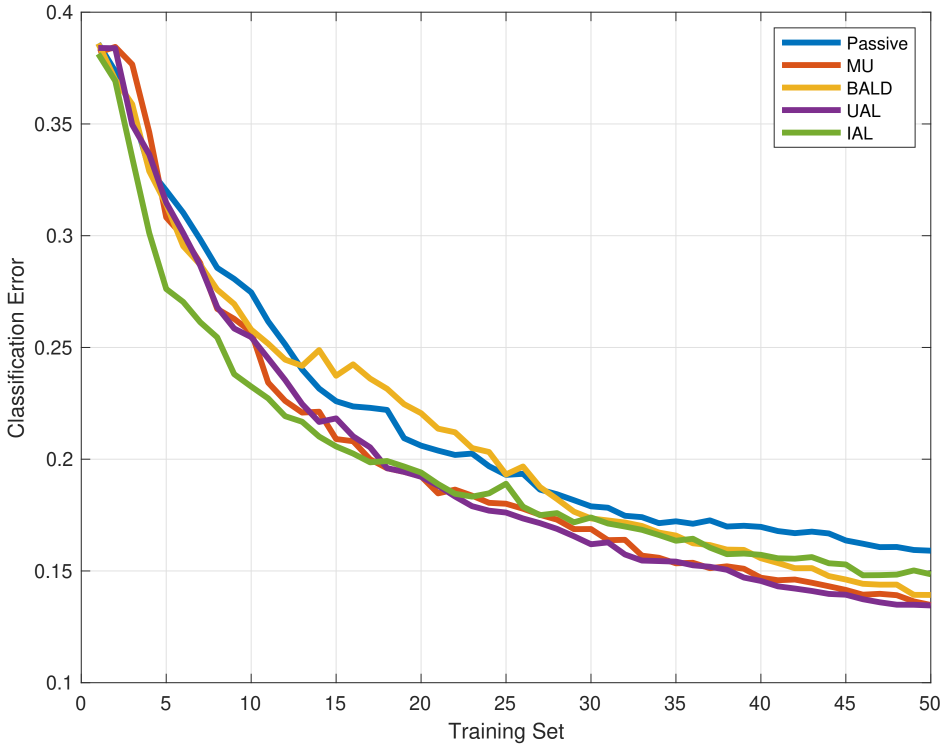


Figure 4.3: Error Probability: Hand-written digits data set, IND

the fact that the test and training distributions are the same. The resemblance of this scenario to the In-Distribution (IND) case in the synthetic example underscores that all active learning schemes surpass passive learning. However, it's crucial to note that the margin between the passive scheme and other schemes is relatively small, rendering this scenario less practically significant. The computational complexities associated with active learning do not manifest a substantial improvement over random sampling. Despite its limited practical interest, we present and analyze this IND scenario to underscore that, for training and test sets within the same distribution, IAL and UAL exhibit performance akin to MU and BALD, albeit without introducing notable practical value.

Subsequently, we investigate the Out-Of-Distribution (OOD) scenario where the training set encompasses images ranging from 0 to 9, while the test set exclusively consists of images depicting the digits 7 and 9. This scenario mirrors real-world conditions where training data invariably includes OOD images, necessitating the active learning scheme's ability to effectively handle such instances.

Unlike the two previous examples, in the OOD case, the selection algorithm may select an OOD example. Once an OOD feature is passed to an Oracle, it will label it as OOD and this feature will not be included in the training set. However, a labelling operation has occurred and the objective of active learning is to minimize any such labelling. Therefore, the x-axis in the following error figures will be denoted as "Oracle Calls" meaning that we count the number of labelling operations done and not the training set size. In Figure 4.4, we plot the error rates for all the schemes in the OOD

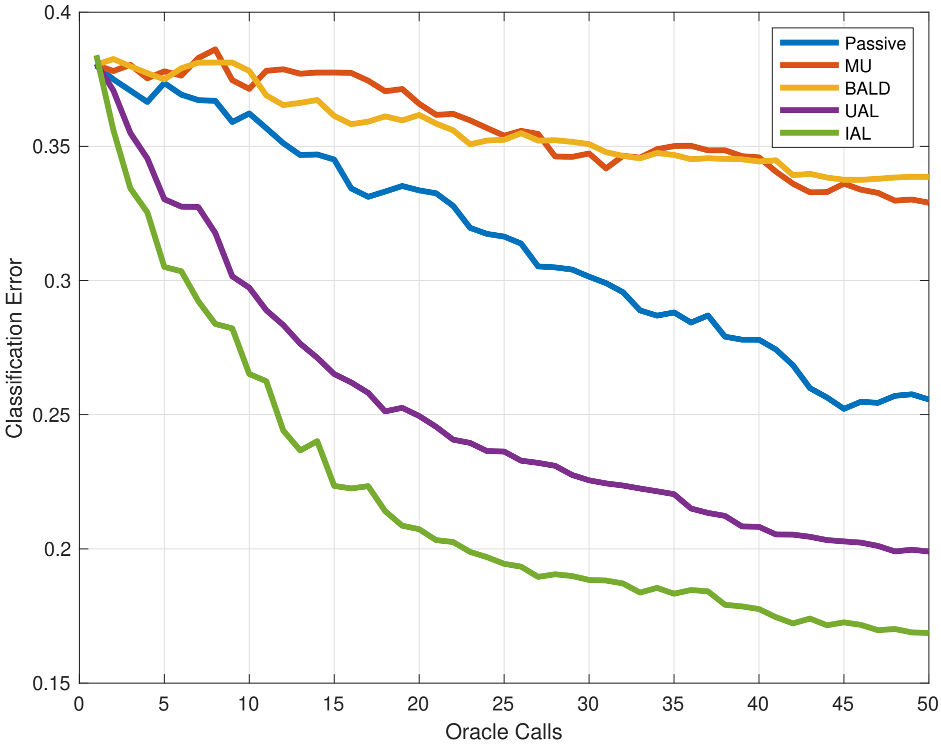


Figure 4.4: Error Probability: Hand-written digits data set, OOD

case. We can see that IAL is superior to the other schemes and UAL is second best.

In Figure 4.5, the probability for selecting an IND point is presented for each AL scheme. We can see that the passive scheme has a fixed 0.2 probability as expected since 7 and 9 represent 20% of the dataset. IAL has the highest selection probability which accounts why it performs so well. This advantage over UAL is most probably due to the fact that the stochastic setting is not valid and the "true" data distribution is not part of GPC.

Moreover, in Figure 4.6, the training set size is presented as a function of the Oracle calls. The idea is to show the efficiency of IAL's selection policy. Again, this shows that IAL is better at finding informative IND features than the other schemes. First order analysis can model the training set size as accumulating Bernoulli random variables with the selection probability as described in Figure 4.5. Therefore, the slope of the mean training set size will be np (n is the selection iteration and p is the IND selection probability). For IAL $p \approx 0.7$ and for passive $p = 0.2$. Clearly, IAL has a larger linear slope for the mean training set size.

In Figure 4.7, we compare the error rates for IAL in both the In-Distribution (IND) and Out-Of-Distribution (OOD) scenarios, with normalization based on the effective training set in the OOD case. The aim is to see if IAL's performance in the OOD case is comparable to its performance in the IND case. This involves comparing the classification error when only IND samples are available versus when IAL selects IND samples from a pool that includes OOD samples. Notably, the two error rates closely resemble each other, suggesting that IAL adeptly avoids OOD samples while

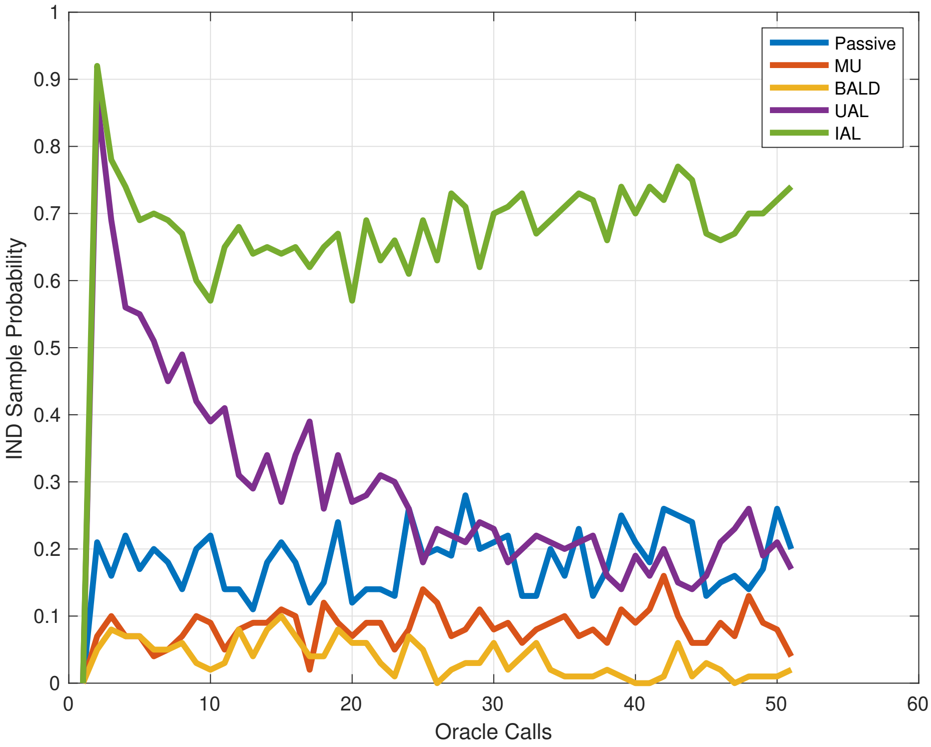


Figure 4.5: IND selection probability: Hand-written digits data set

strategically selecting the most informative samples from the IND distribution, in contrast to a random selection from the IND samples.

A potential counterargument might suggest coupling BALD or MU with an OOD detector to achieve similar results to IAL. However, this approach entails training the OOD detector with numerous IND samples, fine-tuning its threshold and parameters, and crucially, necessitates an expert to define the similarity metric (as different OOD detectors may yield varying results). IAL holds the advantage of implicitly detecting OOD samples while concurrently selecting informative IND samples. We defer to future research the exploration of coupling BALD and MU with distinct OOD detectors for comparison with IAL.

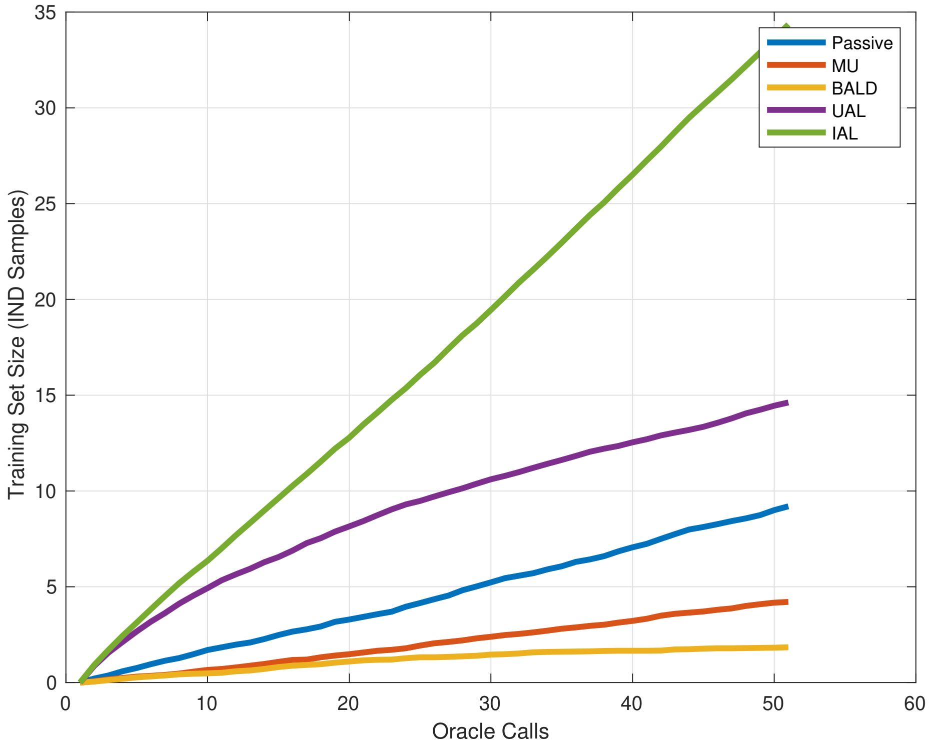


Figure 4.6: Training set size vs Oracle calls: Hand-written digits data set

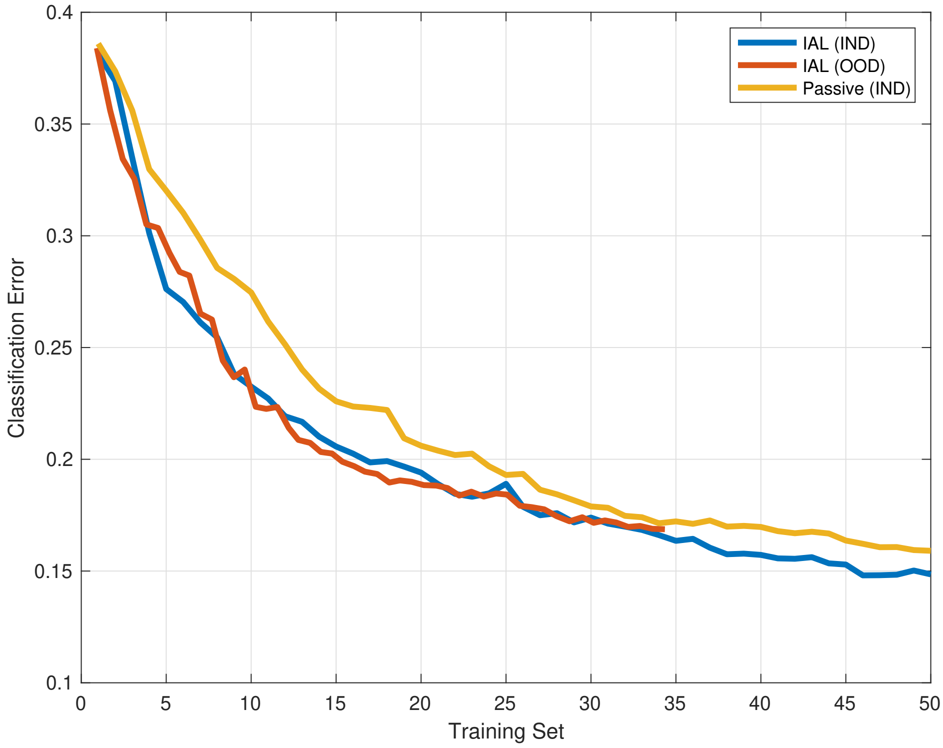


Figure 4.7: Normalized OOD and IND Performance analysis: Hand-written digits data set

Chapter 5

Deep Individual Active Learning

In this chapter we deal with the individual setting and concentrate on the Deep Neural Network (DNN) hypothesis class and is based on the work in [29]. Recent leading strategies are based on the assumption that the training pool has the same distribution as the test set, which may not be the case in privacy-sensitive applications where user data cannot be annotated. In this work, we rely on the individual setting, which does not assume a probabilistic relationship between the training and test data. We propose a criterion that chooses to label data points that minimize the min-max regret on the test set. By applying an approximate version of this criterion to neural networks, we show that in the presence of out-of-distribution data, the proposed criterion reduces the required training set size by up to 15.4%, 11%, and 35.1% for CIFAR10, EMNIST, and MNIST datasets respectively.

5.1 Related Work

Recent research has focused on obtaining a diverse set of samples for training deep learning models with reduced sampling bias. The strategies [11, 19, 23, 27] aim to quantify the uncertainties of samples from the unlabeled pool and utilize them to select a sample for annotation. Their underlying assumptions are that the distribution of the unlabeled pool and the test set are similar and that the data follows some parametric distribution. However, this may not always be true, particularly in privacy-sensitive applications where real user data cannot be annotated [24] and the unlabeled pool may contain irrelevant information. In such cases, choosing samples from the unlabeled pool may not necessarily improve model performance on the test set.

A widely used criterion for active learning is Bayesian Active Learning by Disagreement (BALD) which was originally proposed by [19]. This method finds the unlabeled sample $\hat{x}_i$ that maximizes the mutual information between the model parameters θ and the candidate label random variable Y_i given the candidate x_i and training set $z^{n-1} = \{(x_i, y_i)\}_{i=1}^{n-1}$:

$$\hat{x}_i = \underset{x_i}{\operatorname{argmax}}\, I(\theta; Y_i | x_i, z^{n-1})$$

where $I(X; Y|z)$ denotes the mutual information between the random variables X and Y conditioned on a realization z. The idea in BALD's core is to minimize the uncertainty about model parameters using Shannon's entropy. This criterion also appears as an upper bound on information based complexity of stochastic optimization [20] and also for experimental design [21, 22]. There is an issue of postulating a reasonable prior for this Bayesian approach. Empirically, this approach was investigated by [23], where a heuristic Bayesian method for deep learning was proposed leading to several heuristic active learning acquisition functions that were explored within this framework.

However, BALD has a fundamental disadvantage if the test distribution differs from the training set distribution since what is maximally informative for model estimation may not be maximally informative for test time prediction. In a previous work, [27], we derived a criterion named Universal Active Learning (UAL) that takes into account the unlabeled test set when optimizing the training set:

$$\hat{x}_i = \underset{x_i}{\operatorname{argmin}}\, I(\theta; Y|X, x_i, Y_i, z^{n-1})$$

where X and Y are the test feature and label random variables. UAL is derived from a Capacity-Redundancy theorem [26] and implicitly optimizes an exploration-exploitation trade-off in feature selection. In addition, in the derivation of [26], the prior on θ is expressed as the Capacity maximizing distribution for $I(\theta; Y|X, x_i, Y_i, z^{n-1})$.

It should be noted that recently [68], have proposed a criterion denoted Expected Predictive Information Gain (EPIG) which also takes into account the unlabelled test set and focuses on prediction and not model estimation:

$$\hat{x}_i = \underset{x_i}{\operatorname{argmax}}\, I(Y; Y_i|X, x_i, z^{n-1})$$

We show in Appendix B.3 that EPIG is equivalent to UAL, but unlike EPIG which does not optimize the model prior, UAL provides an expression for the optimal model prior.

UAL and BALD assume that both training and test data follow a conditional distribution which belongs to a given parametric hypothesis class, $\{p\,(y|x, \theta)\}$. This assumption cannot be verified on real world data thus limiting its application. As an alternative to making distributional assumptions, we build upon the *individual setting* [25]. This setting does not assume any probabilistic connection between the training and test data. Moreover, the relationship between labels and data can even be determined by an adversary. The generalization error in this setting is known as the *regret* [31], which is defined as the log-loss difference between a learner and a *genie*: a learner that knows the specific test label but is constrained to use an explanation from a set of hypotheses.

The predictive Normalized Maximum Likelihood (pNML) learner [31] was proposed as the min-max solution of the regret, where the minimum is over the learner choice and the maximum is for any possible test label value. The pNML was previously developed for linear regression [55] and was evaluated empirically for DNN [56].

The setting considered in this chapter, i.e. active learning with no distributional assumption, is related to the active online learning literature [69, 70] which deals primarily with task-agnostic learning which does not assume a connection between the training and test tasks. [69] proposed an active learning that works efficiently with the deep networks. A small parametric module, named “loss prediction module”, is attached to a target network, and learns it to predict target losses of unlabeled inputs. Then, this module can suggest data that the target model is likely to produce a wrong prediction. This method is task-agnostic as networks are learned from a single loss regardless of target tasks. [70] suggested a pool-based semi-supervised active learning algorithm that implicitly learns a sampling mechanism in an adversarial manner. Unlike conventional active learning algorithms, this approach is task agnostic, i.e., it does not depend on the performance of the task for which we are trying to acquire labeled data. This method learns a latent space using a variational autoencoder (VAE) and an adversarial network trained to discriminate between unlabeled and labeled data. The minimax game between the VAE and the adversarial network is played such that while the VAE tries to trick the adversarial network into predicting that all data points are from the labeled pool, the adversarial network learns how to discriminate between dissimilarities in the latent space.

In this chapter, we derive an active learning criterion that takes into account a trained model, the unlabeled pool, and the unlabeled test features. The criterion is designed to select a sample to be labeled in such a way that, when added to the training set with its worst-case label, it attains the minimal pNML regret for the test set. Additionally, we provide an approximate version of this criterion that enables faster and practical application for deep neural networks (DNNs).

Throughout this chapter, a sequence of samples will be denoted $x^n = (x_1, x_2, ..., x_n)$. The variables $x \in \mathbb{X}$ and $y \in \mathbb{Y}$ will represent the features and labels respectively with $\mathbb{X}$ and $\mathbb{Y}$ being the sets containing the features and label's alphabet respectively.

5.2 Deep individual active learning

The DNN hypothesis class poses a challenging problem for information-theoretic active learning since its parameter space is of very high dimension and the weights posterior distribution is analytically intractable. For the DNN hypothesis set, [71] estimated the pNML distribution by fine-tuning the last layers of the network for every test input and label combination. This approach is computationally expensive since training is needed for every test input. [56] suggested a way to accelerate the pNML computation in DNN by using approximate Bayesian inference techniques to produce a tractable approximation to the pNML. Moreover, direct application of deep active learning schemes is unfeasible for real world large scale data since it requires training the entire model for each possible training point. To make matters worse, for IAL, the network also needs to be trained for every test point and every possible corresponding label.

In this section, we derive an approximation of IAL for DNNs which is based on variational inference algorithms [23, 36, 64]. We define the hypothesis class in this case as follows:

$$p\left(y|x,\theta\right) = softmax\left(f_\theta\left(x\right)\right) \tag{5.1}$$

where θ are all the weights and biases of the network and $f_\theta\left(x\right)$ is the model output before the last softmax layer. Note that x, y and $p(\theta)$ are test feature, test label and prior on the weights respectively.

The MAP estimation for θ is

$$\hat{\theta} = \arg\max_\theta p\left(y^n, y|x^n, x, \theta\right) p(\theta), \tag{5.2}$$

where the prior $p(\theta)$ acts as a regularizer over the latent vector θ. It is common practice to use some regularization mechanism to control the training error for DNN's. In order to embed the regularization mechanism into the MAP we introduced this prior $p(\theta)$.

Given a training set, the maximization of the likelihood function $p\left(y^n, y|x^n, x, \theta\right) p(\theta)$ is performed by training the DNN with all the data and converging to a steady state maxima. Note that x^{n-1}, y^{n-1} are assumed known while x_n, y_n, x and y are not known and all the different possibilities need to be considered resulting with multiple $\hat{\theta}$'s.

In order to avoid re-training the entire network for all possible values of x, y, x_n and y_n, we utilize the independence between soft-max scores in the MAP estimation. Using Bayes, we observe that (5.2) can be re-written as:

$$\hat{\theta} = \arg\max_{\theta} p\left(y|x, \theta\right) p\left(y_n|x_n, \theta\right) p\left(\theta|y^{n-1}, x^{n-1}\right) \tag{5.3}$$

where $p\left(\theta|y^{n-1}, x^{n-1}\right)$ is the posterior of θ given the available data $z^{n-1} = (x^{n-1}, y^{n-1})$.

The posterior $p\left(\theta|y^{n-1}, x^{n-1}\right)$ is not dependent on the test data (x, y) and the evaluated labeling candidate (x_n, y_n), thus can be computed once per selection iteration and then used in the IAL selection process. This is a very important point which needs to be highlighted: **There is no need to re-train the network for every (x, y) and (x_n, y_n). We only need to train the network using x^{n-1}, y^{n-1} and then during the IAL selection process run forward passes on different θ to compute $p\left(y|x, \theta\right)$ and $p\left(y_n|x_n, \theta\right)$.** This fact is a significant computational complexity reduction since the number of possible points x_n can be huge. This trick is what makes this algorithm practical.

However, we cannot be satisfied with just a single DNN training pass since we want to acquire a distribution over the weights θ. This requires some advanced techniques [32, 72, 73] which involve multiple training passes over the network but significantly less than the feature space. To make matters worse, for a DNN, the posterior, $p\left(\theta|y^{n-1}, x^{n-1}\right)$ is multi modal and intractable to compute directly. Therefore, we propose to approximate it by some simpler distribution which will allow easier computation of the maximum likelihood $\hat{\theta}$. We recall that Algorithm 3 also estimates this posterior using EP and we tried to use this method to approximate the posterior. This approach didn't produce good results and we hypothesize that since EP is based on a single mode Gaussian approximation and the posterior is multi-modal, the approximation is not good enough. Also, computing EP with every training and test points on a DNN is computationally prohibitive and thus we conclude that a different approach for approximating the posterior is needed.

5.2.1 Variational Inference

Variational inference is a technique used in probabilistic modeling to approximate complex probability distributions that are difficult or impossible to calculate exactly [73, 74, 75]. Variational inference has been used in a wide range of applications, including in Bayesian neural networks, latent Dirichlet allocation, and Gaussian processes. The goal of variational inference is to find an approximation, $q^*(\theta)$ from a parametric family $\mathbb{Q}$, to the true distribution, $p(\theta|z^{n-1})$, that is as close as possible to the true distribution, but is also computationally tractable. This goal is formulated as minimizing the Kullback-leibler (KL) divergence between the two distributions (also called Information projection):

$$q^*(\theta) = \operatorname*{argmin}_{q \in \mathbb{Q}} D_{KL}\left(q(\theta)||p(\theta|z^{n-1})\right)$$

There are different algorithms for implementing variational inference, most involve optimizing a lower bound on the log-likelihood of the data under the true distribution (called evidence). The lower bound is defined as the difference between the true distribution's data log-likelihood and the Kullback-Leibler (KL) divergence between the true distribution and the approximation. The KL divergence measures the distance between the two distributions, and so, optimizing the lower bound is equivalent to minimizing the distance between the true distribution and the approximation.

One common algorithm for implementing variational inference is called mean field variational inference [76]. In this approach, the approximation to the true distribution is factorized into simpler distributions that are easier to work with, such as Gaussians or Bernoullis. The parameters of these simpler distributions are then optimized to minimize the KL divergence between the true distribution and the approximation.

Another algorithm for variational inference is called stochastic variational inference [77]. In this approach, the optimization is performed using stochastic gradient descent, with a random subset of the data used in each iteration. This allows the algorithm to scale to large datasets and complex models.

In this work, we opted to use the method in [32], denoted as MC-Dropout (Monte Carlo Dropout), due to its computational simplicity and favorable performance. MC dropout is a technique used in deep learning to estimate the uncertainty of a neural network's predictions. It involves randomly dropping out (setting to zero) some of the neurons in a neural network during training, and then running multiple forward passes on the same input with different dropout masks, which generates different outputs. At inference time, MC dropout is used to obtain a probabilistic estimate of the network's prediction by performing multiple forward passes with different dropout masks, and taking the average or majority vote of the outputs. The variance of the outputs across the different passes gives an estimate of the uncertainty of the prediction. This can be particularly useful in applications such as medical diagnosis or self-driving cars, where knowing the uncertainty associated with a prediction can be important for making informed decisions.

In [32], the authors argued that performing MC-Dropout on DNNs with dropout applied before every weight layer is mathematically equivalent to minimizing the KL divergence between the weight posterior of the full network and a parametric distribution which is controlled by a set of Bernoulli random variables defined by the dropout probability. Therefore, $p\left(\theta|y^{n-1}, x^{n-1}\right)$ can be approximated in KL-sense by a distribution which is controlled by a dropout parameter. We can use this idea in order to approximate (5.3) and find an approximated weight distribution, $q\left(\theta\right)$. Therefore, we can re-write (5.3) as:

$$\hat{\theta} = \arg\max_{\theta} p\left(y|x,\theta\right) p\left(y_n|x_n,\theta\right) q\left(\theta\right) \tag{5.4}$$

However, $q\left(\theta\right)$ as described in [32] is still complex to analytically compute. In fact in [32], the authors do not explicitly compute this distribution but sample it and compute integral quantities on this distribution (such as expectation and variance) using sampling and averaging of multiple independent realizations and the Law of Large Numbers (LLN). Since we focus on point-wise samples from $q\left(\theta\right)$, we cannot use the same approach as in [32].

In this work, we propose to sample M weights, θ_m, from $q\left(\theta\right)$ and find $\hat{\theta}$ among all the different samples. Since the weights are embedded in a high dimensional space, then the probability of the sampled weights can be assumed to be relatively uniform. Therefore we propose to approximate (5.4) as:

$$\hat{\theta} = \arg \max_{\{\theta_m\}_{m=1}^{M}} p\left(y|x, \theta_m\right) p\left(y_n|x_n, \theta_m\right) \tag{5.5}$$

As observed by [32], (5.5) can be computed by running multiple dropout inference passes on the network trained using dropout with z^{n-1}. The dropout inference passes will be the same for both x and x_n. The resulting algorithm denoted Deep Individual Active Learning (DIAL) is shown in Algorithm 3 and follows these steps:

1. Train a model on the labeled training set z^{n-1} with dropout.

2. Run MC-Dropout inference for M iterations on all the unlabeled pool and test set.

3. Find the weight that maximizes DNN prediction of the test input and the unlabeled candidate input as in (5.3).

4. Accumulate the pNML regret of the test point given these estimations.

5. Find the unlabeled candidate for which the worst case averaged regret of the test set is minimal as in (4.6).

For step 2, since the variational posterior associated with MC-Dropout is difficult to evaluate, we assume that it is uniform for all the sampled weights.

We emphasize the significant complexity reduction provided by our approximation: a naive implementation of pNML computation would require training the network over all possible training points x_n and test points x with all possibilities of their respective labels y_n, y. This would render our criterion unfeasible for real-world applications. Our proposed approach, DIAL, only requires performing training with dropout on z^{n-1} and then performing only inference passes (considerably faster than training passes) to get multiple samples of the weights.

5.3 Experiments

In this section, we analyze the performance of DIAL and compare its performance to state-of-the-art active learning criteria. We tested the proposed DIAL strategy in two scenarios:

- The initial training, unlabeled pool, and test data come from the same distribution (IND scenario).

- There are OOD samples present in the unlabeled pool (OOD scenario).

Algorithm 3 DIAL: Deep Individual Active Learning

```
**Input** Training set $z^{n-1}$, unlabeled pool and test samples $\{x_i\}_{i=1}^{N}$ and $\{x_k\}_{k=1}^{K}$.
**Output** Next data point for labeling $\hat{x}_i$
Run MC-Dropout using $z^{n-1}$ to get $\{\theta_m\}_{m=1}^{M}$
$\mathbf{S} = zeros(N, |\mathbb{Y}|)$
for $i \leftarrow 1$ to $N$ do
  for $y_i \in \mathbb{Y}$ do
    for $k \leftarrow 1$ to $K$ do
      $\Gamma = 0$
      for $y_k \in \mathbb{Y}$ do
        $\hat{\theta} = \text{argmax}_{\theta_m} p(y_k|x_k, \theta_m) p(y_i|x_i, \theta_m)$
        $\Gamma = \Gamma + p\left(y_k|x_k, \hat{\theta}\right)$
      end for
      $\mathbf{S}(i, y_i) = \mathbf{S}(i, y_i) + \log \Gamma$
    end for
  end for
end for
$\hat{x}_i = \text{argmin}_{x_i} \max_{y_i} \mathbf{S}$
```

The reason for using the individual setting and DIAL as its associated strategy in the presence of OOD samples is that it does not make any assumptions about the data generation process, making the results applicable to a wide range of scenarios, including PAC [78], stochastic [25], adversarial settings, as well as samples from unknown distributions.

We considered the following datasets for training and evaluation of the different active learning methods:

- **The MNIST dataset** [79] consists of 28 × 28 grayscale images of handwritten digits, with 60K images for training and 10K images for testing.

- **The EMNIST dataset** [80] is a variant of the MNIST dataset that includes a larger variety of images (upper and lower case letters, digits, and symbols). It consists of 240K images with 47 different labels.

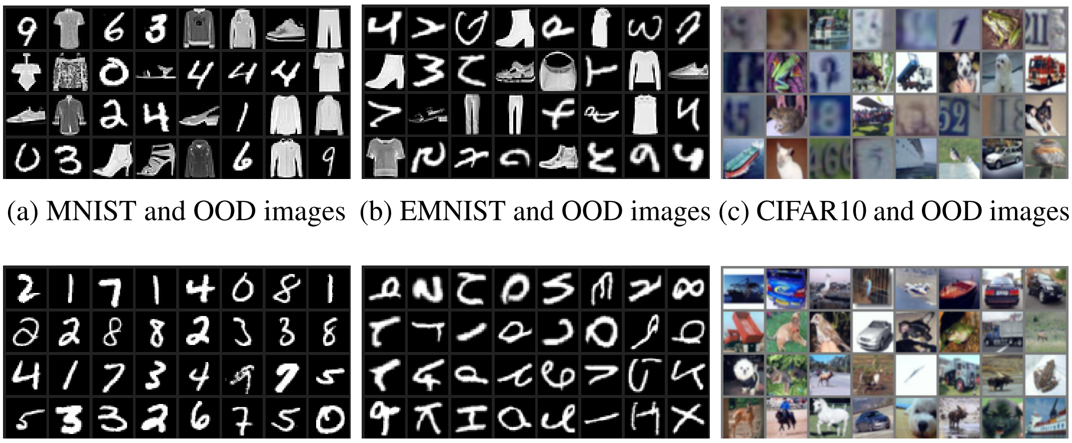
- **The CIFAR10 dataset** [81] consists of 60K 32 × 32 color images in 10 classes. The classes include objects such as airplanes, cars, birds, and ships.

- **Fashion MNIST** [82] is a dataset of images of clothing and accessories, consisting of 70K images. Each image is 28 × 28 grayscale pixels.

- **The SVHN dataset** [83] contains 600K real-world images with digits and numbers in natural scene images collected from Google Street View.

We built upon [84] and [68] open-source implementations of the following methods:

The Random sampling algorithm is the most basic approach in learning. It selects samples to label randomly, without considering any other criteria. This method can be



(a) MNIST and OOD images (b) EMNIST and OOD images (c) CIFAR10 and OOD images

(d) MNIST test images (e) EMNIST test images (f) CIFAR10 test images

Figure 5.1: Datasets that contain a mix of images with OOD samples. (Top) Unlabeled pool contains OOD samples (Bottom) Test set includes only valid data.

useful when the data are relatively homogeneous and easy to classify, but it can be less efficient when the data are more complex or when there is a high degree of uncertainty.

The Bayesian Active Learning by Disagreement (BALD) method [23] utilizes an acquisition function that calculates the mutual information between the model's predictions and the model's parameters. This function measures how closely the predictions for a specific data point are linked to the model's parameters, indicating that determining the true label of samples with high mutual information would also provide insight into the true model parameters.

The Core-set algorithm [11] aims to find a small subset from a large labeled dataset such that a model learned from this subset will perform well on the entire dataset. The associated active learning algorithm chooses a subset that minimizes this bound, which is equivalent to the k-center problem.

The Expected Predictive Information Gain (EPIG) method [68] was motivated by BALD's weakness in prediction-oriented settings. This acquisition function directly targets a reduction in predictive uncertainty on inputs of interest by utilizing the unlabelled test set. It is shown in Appendix B.3 that this approach is similar to UAL [27], where the main difference is that UAL assumes the stochastic setting, where the data follow some parametric distribution.

5.3.1 Experimental Setup

The first setting we consider consists of an initial training set, an unlabeled pool (from which the training examples are selected), and an unlabeled test set, all drawn from the **same distribution**. The experiment includes the following four steps:

1. A model is trained on the small labeled dataset (initial training set).

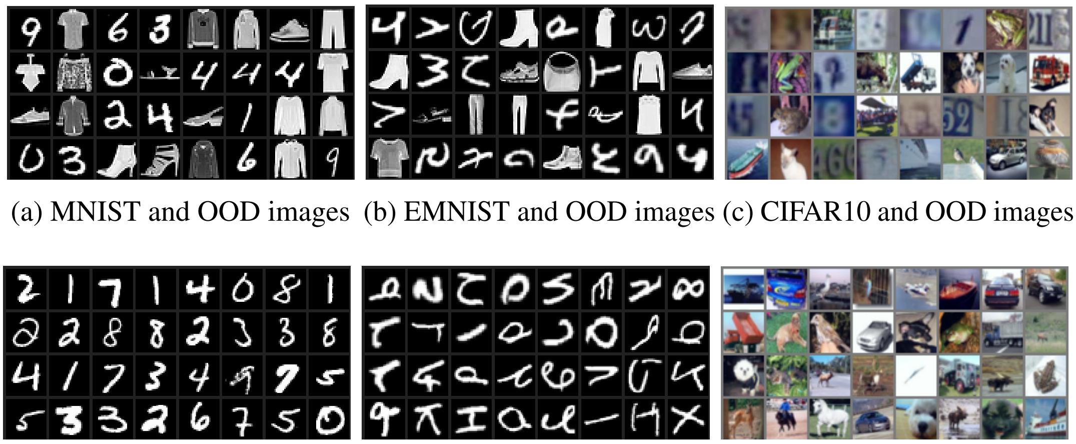
2. One of the active learning strategies is utilized to select a small number of the most informative examples from the unlabeled pool. Since it is computationally expensive to select one sample at a time, the 256 samples with the highest score are taken per AL scheme.

3. The labels of the selected samples are queried and added to the labeled dataset.
4. The model is retrained using the new training set.

Steps 2–4 are repeated multiple times, with the model becoming more accurate with each iteration, as it is trained on a larger labeled dataset.

In addition to the standard setting, we evaluate the performance in **the presence of OOD samples**. In this scenario, the initial training and test sets are drawn from the same distribution, but the unlabeled pool contains a mix of OOD samples. When an OOD unlabeled sample is selected for annotation, it is not used in training of the next iteration of the model. Across all x-axis values in the subsequent test accuracy figures, the presented metric is the count of Oracle calls, reflecting the instances when a selection strategy chose a sample, whether it be IND or OOD. It is crucial to differentiate this metric from the training set size, as the selection of an OOD sample leads to an increase in the number of Oracle calls, while the training set size remains unaffected. An effective strategy would recognize that OOD samples do not improve performance on the test set and avoid selecting them.

A visual representation of the scenario with OOD samples is illustrated in Figure 5.2a–c. These figures show the unlabeled pool, which contains a mixture of both IND and OOD samples. Figure 5.2d–f show the test set, which contains only IND samples. We argue that this is a representative setting for active learning in real life. In the real world, unlabelled pools are collected from many data sources and will most certainly contain OOD data. The process of pruning the unlabelled pool is a costly process and involves human inspection and labeling, which needs to be minimized. This is exactly the goal of active learning and finding a criterion which implicitly filters OOD data is of significant interest.



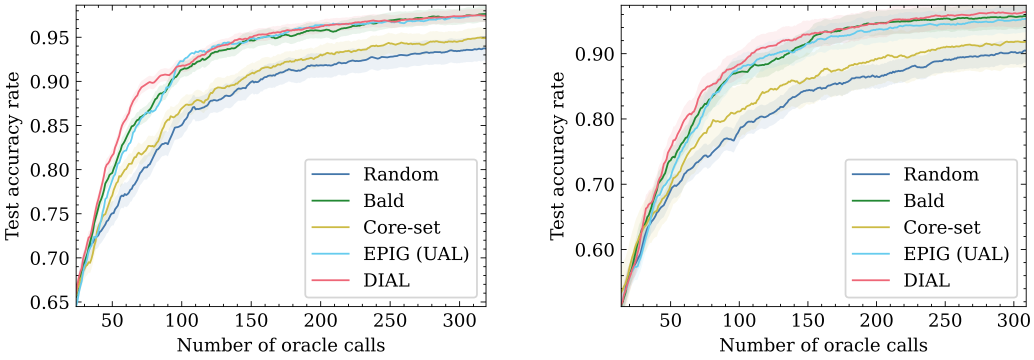
(a) MNIST and OOD images (b) EMNIST and OOD images (c) CIFAR10 and OOD images

(d) MNIST test images (e) EMNIST test images (f) CIFAR10 test images

Figure 5.2: Datasets that contain a mix of images with OOD samples. (Top) Unlabeled pool contains OOD samples (Bottom). Test set includes only valid data.

5.3.2 MNIST Experimental Results

Following [23], we considered a model consisting of two blocks of convolution, dropout, max-pooling, and ReLu, with 32 and 64 5 × 5 convolution filters. These blocks are followed by two fully connected layers that include dropout between them. The layers have 128 and 10 hidden units, respectively. The dropout probability was set to 0.5 in all three locations. In each active learning round, a single sample was selected. We executed all active learning methods six times with different random seeds. For BALD, EPIG, and DIAL, we used 100 dropout iterations and employed the criterion on 512 random samples from the unlabeled pool. MNIST results are shown in Figure 5.3a. The largest efficiency is at a number of Oracle calls of 71, where DIAL attains an accuracy rate of 0.9, while EPIG and BALD achieve an accuracy rate of 0.86.



(a) MNIST

(b) MNIST with OOD

Figure 5.3: Accuracy as function of number of Oracle calls on MNIST dataset. DIAL outperforms the baselines for the two setups.

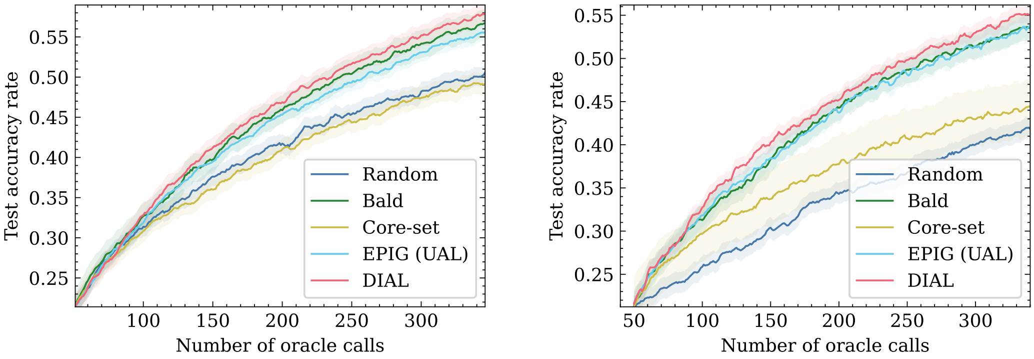
To simulate the presence of OOD samples, we added the Fashion MNIST to the unlabeled pool such that the ratio of Fashion MNIST to MNIST was 1:1. In this setting, DIAL outperforms all other baselines, as shown in Figure 5.3b. DIAL is the top-performing method and has better accuracy than EPIG, BALD, Core-set, and Random. The largest efficiency is an accuracy rate of 0.95, where DIAL uses 240 Oracle calls, while BALD needs 307 (−35.1%). EPIG never reaches this accuracy level. The number of Oracle calls for additional accuracy rates is shown in Table 5.1.

Table 5.1: MNIST with OOD number of Oracle calls at x% accuracy.

Methods	**85% Acc.**	**75% Acc.**	**65% Acc.**
Random	145	73	36
Core-set	117	61	33
BALD	83	51	32
EPIG	84	56	35
DIAL	**73 (−12.1%)**	**48 (−5.9%)**	**30 (−6.2%)**

5.3.3 EMNIST Experimental Results

We followed the same setting as the MNIST experiment with a slightly larger model than MNIST consisting of three blocks of convolution, dropout, max-pooling, and ReLu. The experimental results, shown in Figure 5.4a, indicate that DIAL is the top-performing method. For an accuracy rate of 0.56, it requires 8.3% less Oracle calls when compared to the second best method.



(a) EMNIST

(b) EMNIST with OOD

Figure 5.4: Active learning performance on the EMNIST dataset. DIAL is more efficient than tested baselines in the number of Oracle calls.

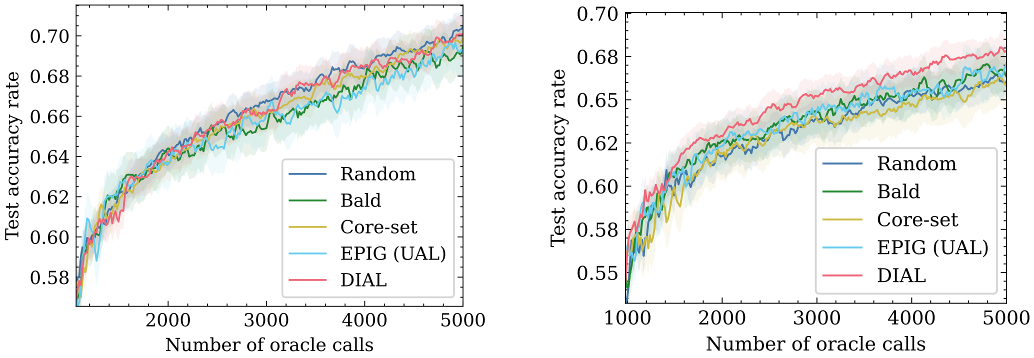
In the presence of OOD samples, the DIAL method outperforms all other baselines, as shown in Figure 5.4b and Table 5.2. For 300 Oracle calls, DIAL achieves a test set accuracy rate of 0.52, while BALD, EPIG, Core-set, and Random attain 0.51, 0.5, 0.42, and 0.40, respectively. For an accuracy rate of 0.53, DIAL needs 308 Oracle calls, while BALD and EPIG require 346 and 342, respectively (−11%). Moreover, Core-set and Random do not achieve this accuracy.

Table 5.2: EMNIST with OOD number of Oracle calls at x% accuracy.

Methods	**40% Acc.**	**30% Acc.**	**25% Acc.**
Random	281	140	80
Core-set	221	96	62
BALD	154	85	**59**
EPIG	157	**84**	**59**
DIAL	**138 (−10.4%)**	**84 (−1.2%)**	**59 (0%)**

5.3.4 CIFAR10 Experimental Results

For the CIFAR10 dataset, we utilized ResNet-18 [85] with an acquisition size of 16 samples. We used 1K initial training set size and measured the performance of the active learning strategies up to a training set size of 3K. The CIFAR10 results are shown in Figure 5.5a. Overall, DIAL and Random perform the same and have a better test set accuracy than the other baselines for Oracle calls greater than 2100.



(a) CIFAR10

(b) CIFAR10 with OOD

Figure 5.5: The left figure illustrates the performance of CIFAR10 using only IND samples. The DIAL method performs similarly to the Random method. The figure on the right shows the performance of a combination of OOD samples, where DIAL outperforms all other methods.

When the presence of OOD samples in the unlabeled pool is considered, as shown in Figure 5.5b, DIAL outperforms the other methods. Table 5.3 shows the number of Oracle calls required for different accuracy levels. For the same accuracy rate of 0.65, DIAL needs up to 15.4% less Oracle calls than the second best method. This can be explained by Figure 5.6, which shows the ratio of OOD samples to the number of Oracle calls. The figure suggests that DIAL outperforms other criteria by selecting fewer OOD samples, contributing to its commendable performance. It is noteworthy that in all OOD scenarios, DIAL demonstrated superior ability to identify in-distribution samples without explicit knowledge of the distribution and solely utilizing unlabeled test features. This underscores the universality of DIAL, showcasing its adaptability to various distribution shifts. Additionally, the second-best performer, EPIG, also considers the unlabeled test set and performs better than other baseline methods but falls short of DIAL. Notably, BALD and Core-set exhibit similar behavior, possibly attributed to their emphasis on model estimation rather than leveraging the test set for predictive focus.

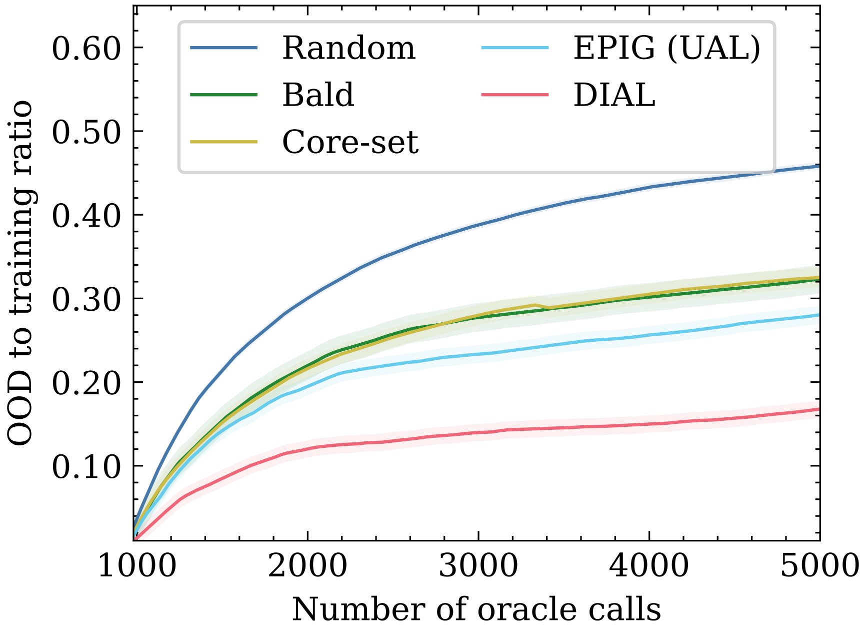


Figure 5.6: The amount of chosen OOD samples for CIFAR10 with the presence of OOD samples.

Table 5.3: CIFAR10: the presence of OOD samples: Number of Oracle calls at specific accuracy rate values.

Methods	**66% Acc.**	**62% Acc.**	**58% Acc.**
Random	3956	1828	1220
Core-set	4468	1844	1412
BALD	4020	1636	1202
EPIG	3636	1700	1108
DIAL	**3076 (−15.4%)**	**1556 (−4.9%)**	**1060 (−4.3%)**

5.4 Limitations

The proposed DIAL algorithm is a min-max strategy for the individual settings. However, DIAL may not be the most beneficial approach in scenarios where the unlabeled pool is very similar to the test set, where different selection strategies may perform similarly and with smaller complexity. This limitation of DIAL is supported by the experimental results of Section 5.3.4, where the DIAL algorithm performed similarly to random selection for the CIFAR10 dataset (but better than all the other baselines).

Another limitation of DIAL is that it has a higher overhead computation compared to other active learning methods such as BALD. This is because DIAL involves computing the regret on the test set, which requires additional computations and could become significant when the unlabeled pool or the test set are very large.

5.5 Conclusions

In this study, we propose a min-max active learning criterion for the individual setting, which does not rely on any distributional assumptions. We have also developed an efficient method for computing this criterion for DNNs. Our experimental results demonstrate that the proposed strategy, referred to as DIAL, is particularly effective in the presence of OOD samples in the unlabeled pool. Specifically, our results show that DIAL requires 12%, 10.4%, and 15.4% fewer Oracle calls than the next best method to achieve a certain level of accuracy on the MNIST, EMNIST, and CIFAR10 datasets, respectively.

Chapter 6

Concluding Remarks and Open Questions

In this thesis, active learning was studied through the lens of information theory and in particular via the concepts of universal prediction [25]. This approach has enabled the discovery of interesting links between machine learning and information theory which are utilized to the derivation of useful active learning criteria. Throughout this work, we have considered the scenario in which the distribution of the test data may shift from the training pool data. We argue that this is the most common scenario in practice since it is very easy to collect a large quantity of unlabeled data and the significant bottleneck is the labelling of these data points. Removing this distribution shift from large scale unlabelled data pools will result with a significant labelling effort which should be avoided. We solve this problem by granting the learner access to a very small unlabelled test set. Since this set is unlabelled then privacy of the test is kept which may be crucial for different applications such as medical data. Our proposed criteria utilizes this test set and implicitly selects informative data points from the large data pool.

This work is be divided to two different data settings: stochastic and individual. The stochastic setting assumes that the data follows some parametric distribution, thus enables elegant mathematical results. It provides fairly low complexity selection algorithms and is more widely used in the active learning literature. However, this assumption is impossible to verify in reality. In contrast, the individual setting does not assume the data follows any distribution and is thus the most general framework. The downside is that the results are worst case and generally more computationally complex.

In the first part of the thesis, the stochastic setting is investigated. First, a Capacity - Redundancy theorem was derived for learning under this setting. This theorem enabled the derivation of a new active learning criterion which was analyzed for different hypothesis classes. The proposed criterion, termed UAL, intrinsically balances an exploration-exploitation trade-off in data selection. It was shown that it is not sufficient to improve the model estimation but there should also be a focus on test prediction. It was further shown, using a simulation that this criterion outperforms commonly used uncertainty maximization criteria which focus on exploration. Later, the linear separator problem with asymmetric noise was considered and a low complexity, noise robust algorithm for active learning has been presented. It was proven that this algorithm achieves exponential decay of redundancy and empirically shown that the error probability decays exponentially with the same rate to achieve optimal sample complexity.

In the second part of the thesis, the individual setting was considered. This setting is the most general approach for machine learning since it has no distributional assumptions and is thus the hardest for the same reason. A min-max active learning criterion, IAL, which does not rely on any distributional assumptions was proposed. It has also been demonstrated that for binary classification, IAL coincides with binary search for separable data and optimal experimental design for linear regression. Therefore the proposed criterion can be viewed as a unified active learning framework not specific to any hypothesis class. An empirical test was conducted comparing IAL with several other active learning criteria and demonstrating that IAL is superior in terms of sample complexity. We have also developed an efficient method for computing this criterion for DNNs. Our experimental results demonstrate that the proposed strategy, referred to as DIAL, is particularly effective in the presence of OOD samples in the unlabeled pool. As future work, we plan to investigate batch acquisition criteria that take into account batch selection. This will allow us to consider the relationship between the selected samples and the overall composition of the batch, which may lead to even further improvements in performance.

Appendix A

Proofs for Part I

A.1 Proof of Theorem 1

Proof. The proof is very similar to the one in [34] but with small technical modifications. First, we induce a probability measure $\pi(\theta)$ over the parameter θ:

$$R = \min_{\{\phi_t\}_{t=1}^N} \min_q \max_{\pi(\theta)} \mathbf{E}\left\{\log\left(\frac{p\left(y|x,\theta\right)}{q\left(y|x,x^N,y^N\right)}\right)\right\} \tag{A.1}$$

where the worst θ is with probability one.

Then, observe that,

$$\begin{aligned}\mathbf{E}\left\{\log\left(\frac{p\left(y|x,\theta\right)}{q\left(y|x,x^N,y^N\right)}\right)\right\} &= \sum_\theta \pi(\theta) \sum_{x^N,y^N,x} \\ p\left(y^N,x^N,x|\theta\right) D_{KL}&\left(p\left(y|x,\theta\right)||q\left(y|x,x^N,y^N\right)\right)\end{aligned} \tag{A.2}$$

Since (A.2) is a non-negative weighted sum of convex functions (for each (x,x^N,y^N), the KL divergence is convex in $q\left(y|x,x^N,y^N\right))$ and concave (linear) in $\pi(\theta)$, and the set of distributions is the probability simplex which is compact and convex, then we can apply the Minimax theorem [86].

Plugging (A.2) in to (A.1) and using the Minimax theorem,

$$R = \min_{\{\phi_t\}_{t=1}^N} \max_{\pi(\theta)} \min_q \mathbf{E}\left\{\log\left(\frac{p\left(y|x,\theta\right)}{q\left(y|x,x^N,y^N\right)}\right)\right\} \tag{A.3}$$

Now we can find the learner q (for each x,x^N,y^N) which optimizes (A.3) for a given $\left\{\phi(x_t|x^{t-1},y^{t-1})\right\}_{t=1}^N$ and $\pi(\theta)$.

Note that:

$$p\left(\theta|y^N,x^N,x\right) = \frac{p\left(y^N,x^N,x,\theta\right)}{p\left(y^N,x^N,x\right)}$$

Then,

$$\begin{aligned}\mathbf{E}\left\{\log\left(\frac{p\left(y|x,\theta\right)}{q\left(y|x,x^N,y^N\right)}\right)\right\} &= \mathbf{E}_{\mathbf{x^N},\mathbf{y^N},\mathbf{x}} \sum_\theta \\ p\left(\theta|y^N,x^N,x\right) D_{KL}&\left(p\left(y|x,\theta\right)||q\left(y|x,x^N,y^N\right)\right)\end{aligned} \tag{A.4}$$

Then, the optimal q which minimizes the KL divergence is:

$$q^*\left(y|x,x^N,y^N\right)=\sum_{\theta}p\left(\theta|y^N,x^N\right)p\left(y|\theta,x\right) \tag{A.5}$$

Note that q is optimal regardless of the selection policy and thus optimal for both passive and active learning. The predictor q is a function of the training set and test feature but also loosely dependent (for large N) on $\pi(\theta)$. The optimal prior $\pi(\theta)$ is different for a given selection policy though.

The expected regret of the optimal predictor given a fixed selection strategy and N examples is the conditional mutual information between the test label and model parameters:

$$\mathbf{E}\left\{\log\left(\frac{p\left(y|x,\theta\right)}{q^*\left(y|x,x^N,y^N\right)}\right)\right\}=I\left(Y;\theta|X,Y^N,X^N\right) \tag{A.6}$$

and $\pi(\theta)$ maximizes the mutual information (capacity achieving distribution) for a given policy. □

A.2 Proof of Theorem 2.

Proof. We wish to analyze the conditional mutual information

$$I\left(\theta;Y|X=x,Y^n=y^n,X^n=x^n\right)$$

First, we analyze the posterior:

$$p\left(y|x,y^n,x^n\right)=\sum_{\theta}p(\theta|x,y^n,x^n)p\left(y|x,y^n,x^n,\theta\right) \tag{A.7}$$

Using the fact that given θ and x, y is independent of x^n, y^n:

$$p\left(y|x,y^n,x^n\right)=\sum_{\theta}p(\theta|y^n,x^n,x)p\left(y|x,\theta\right) \tag{A.8}$$

Using Bayes,

$$p(\theta|y^n,x^n,x)=\frac{p(y^n,x^n,x|\theta)\pi(\theta)}{p(y^n,x^n,x)} \tag{A.9}$$

Therefore,

$$p(\theta|y^n,x^n,x)=\frac{\Pi_{t=1}^n p\left(y_t|x_t,\theta\right)\phi\left(x_t|x^{t-1},y^{t-1}\right)p(x|\theta)\pi(\theta)}{\sum_\theta p(x|\theta)\pi(\theta)\Pi_{t=1}^n p\left(y_t|x_t,\theta\right)\phi\left(x_t|x^{t-1},y^{t-1}\right)} \tag{A.10}$$

Eliminating $\phi\left(x_t|x^{t-1},y^{t-1}\right)$

$$p(\theta|y^n,x^n,x)=\frac{\Pi_{t=1}^n p\left(y_t|x_t,\theta\right)p(x|\theta)\pi(\theta)}{\sum_\theta p(x|\theta)\pi(\theta)\Pi_{t=1}^n p\left(y_t|x_t,\theta\right)} \tag{A.11}$$

Therefore,

$$p\left(y|x, y^n, x^n\right) = \sum_{\theta} p\left(y|x, \theta\right) \frac{\Pi_{t=1}^{n} p\left(y_t|x_t, \theta\right) p(x|\theta)\pi(\theta)}{\sum_{\theta} p(x|\theta)\pi(\theta)\Pi_{t=1}^{n} p\left(y_t|x_t, \theta\right)} \tag{A.12}$$

and thus, for a given $\pi(\theta)$, the value of the posterior $p\left(y|x, y^n, x^n\right)$ does not depend on the value of the selection policy.

We can write the conditional mutual information explicitly,

$$\begin{aligned} I\left(\theta; Y|X, Y^n, X^n\right) = \sum_{x,y^n,x^n} I\left(\theta; Y|X = x, Y^n = y^n, X^n = x^n\right) \cdot \\ p\left(x, y^n, x^n\right) \end{aligned} \tag{A.13}$$

Then,

$$\begin{aligned} I\left(\theta; Y|X, Y^n, X^n\right) = \sum_{x,y^n,x^n} I\left(\theta; Y|X = x, Y^n = y^n, X^n = x^n\right) \cdot \\ \left(\sum_{\theta} p(x|\theta)\pi(\theta)\Pi_{t=1}^{n} p\left(y_t|x_t, \theta\right) \phi\left(x_t|x^{t-1}, y^{t-1}\right)\right) \end{aligned} \tag{A.14}$$

which can be written as,

$$\begin{aligned} I\left(\theta; Y|X, Y^n, X^n\right) = \sum_{y^n,x^n} \phi\left(x_t|x^{t-1}, y^{t-1}\right) \cdot \\ I\left(\theta; Y|X, Y^n = y^n, X^n = x^n\right) \cdot \left(\sum_{\theta} \pi(\theta)\Pi_{t=1}^{n} p\left(y_t|x_t, \theta\right)\right) \end{aligned} \tag{A.15}$$

Since the weighted average of positive values (mutual information is always larger or equal to 0) is always bigger than the minimum of the set, we come to the conclusion that the optimal selection strategy is a delta function for each step which correspond to the trajectory x^n, y^n which minimizes the conditional mutual information $I(\theta; Y|X, X^n = x^n, Y^n = y^n)$. □

A.3 Proof of Theorem 3

Proof. Applying UAL and assuming the noise is Gaussian with the response model (2.18):

$$I(\underline{\theta}; y_{test}|\underline{x}_{test}, \underline{x}^n, y^n) = h(y_{test}|\underline{x}_{test}, \underline{x}^n, y^n) - h(y_{test}|\underline{\theta}, \underline{x}_{test}, \underline{x}^n, y^n) \tag{A.16}$$

Since the noise, z is independent from the label y given the feature vector $\underline{x}$, then

$$I(\underline{\theta}; y_{test}|\underline{x}_{test}, \underline{x}^n, y^n) = h(y_{test}|\underline{x}_{test}, \underline{x}^n, y^n) - h(z) \tag{A.17}$$

Using the expression for Gaussian entropy,

$$I(\underline{\theta}; y_{test}|\underline{x}_{test}, \underline{x}^n, y^n) = h(y_{test}|\underline{x}_{test}, \underline{x}^n, y^n) - \frac{1}{2}\log\left(2\pi e \sigma_Z^2\right) \tag{A.18}$$

UAL first finds the prior $\pi(\underline{\theta})$ which maximizes the mutual information in (A.18). Since there is a power constraint on $\underline{\theta}$ then $\underline{y}$ will also be power limited due to the linear model.

The distribution which will maximize the the differential entropy for $\underline{y}|X, \underline{\theta}$ under the power constraint is an i.i.d Gaussian distribution. This distribution can be achieved if the prior $\underline{\theta} \sim \mathcal{N}(0, \sigma_\theta^2 I_d)$ is used. Therefore, in the case of the linear regression hypothesis class, the capacity achieving prior can be computed analytically.

Using [87],

$$I(\underline{\theta}; y_{test}|\underline{x}_{test}, \underline{x}^n, y^n) = \mathbb{E}_{\underline{x}_{test}}\left(\log\left(1 + \underline{x}_{test}^T Q \underline{x}_{test}\right)\right) \tag{A.19}$$

where $Q = \left(X^T X + \frac{1}{\sigma_\theta^2} I_d\right)^{-1}$ is the inverse covariance matrix of $p\left(\underline{\theta}|\underline{x}^n, y^n\right)$ which is also Gaussian and thus easy to compute using Kalman filtering. The expectation is performed on the distribution of the test features $\underline{x}_{test}$.

Upper bounding (A.19) we get,

$$\mathbb{E}_{\underline{x}_{test}}\left(\log\left(1 + \underline{x}_{test}^T Q \underline{x}_{test}\right)\right) \leq \mathbb{E}_{\underline{x}_{test}}\left(\underline{x}_{test}^T Q \underline{x}_{test}\right) \tag{A.20}$$

where the bound is tight when $\underline{x}_{test}^T Q \underline{x}_{test} << 1$, which corresponds to high SNR scenarios.

Therefore,

$$\begin{aligned} &\min_{\underline{x}^n} I(\underline{\theta}; y|\underline{x}_{test}, \underline{x}^n, y^n) \leq \\ &\quad \min_{\underline{x}^n} Tr\left(\mathbb{E}\left(\underline{x}_{test}\underline{x}_{test}^T\right) Q\left(\underline{x}^n\right)\right) \end{aligned} \tag{A.21}$$

When the training data is full rank then the correlation matrix is the identity matrix and we can neglect it. □

A.4 Proof of Theorem 4

Proof. In [52], it is proved that PM achieves capacity on the BAC. Achieving capacity essentially means that the maximum amount of bits are transmitted and decoded without error with the minimal amount of channel uses. This is analogous to high accuracy on θ_0 (low generalization error) using as few Oracle calls as possible. This is exactly the target of active learning and we will now show that PM on BAC is equivalent to a specific active learning policy for the hypotheses class discussed here.

The proposed selection scheme selects the training feature, x_t, based on previously observed labels y^{t-1} (x^{t-1} are a deterministic function of y^{t-1}):

$$x_t = F^{-1}_{\theta|y^{t-1}}\left(\frac{p - 0.5}{p + q - 1}\right) \tag{A.22}$$

Therefore, the input to the BAC, v_t, is computed according to:

$$v_t = \begin{cases} 0 & x_t \leq \theta_0 \\ 1 & x_t > \theta_0 \end{cases} \tag{A.23}$$

Now, we would like to show that this selection of x_t generates v_t which achieves capacity for the BAC.

Define an auxiliary Bernoulli random variable $Q \sim Ber\left(\frac{p-0.5}{p+q-1}\right)$ and use the fact that a Cumulative Distribution Function (CDF) is always increasing, then v_t can also be described as:

$$v_t = F_Q^{-1}\left(F_{\theta|Y^{t-1}}\left(\theta_0\right)\right) \tag{A.24}$$

which is exactly the PM scheme for a BAC channel with p, q!

Therefore, the error probability on the message θ approaches zero as the number of channel uses, n, goes to infinity:

$$\lim_{n\to\infty} \sup_{\theta_1} \int_{\theta_1}^{\theta_1+2^{-nC_W}} p(\theta|y^n, x^n)d\theta = 1 \tag{A.25}$$

This means that most of the probability mass is centred in an interval of length 2^{-nC_W} where the true barrier, θ_0, resides, where Q is the input distribution to the BAC and W is the BAC transition probability.

Now we can analyze the active learning criterion for the PM selection with training x^n, y^n. We will compute the desired mutual information using the difference of the two conditional entropies:

$$\begin{aligned} &H(Y|X, X^n = x^n, Y^n = y^n) = \\ &\int H_B\left(\int P(Y=1|x,\theta)p(\theta|x^n, y^n)d\theta\right)p(x)dx \end{aligned} \tag{A.26}$$

and the conditional entropy with θ:

$$\begin{aligned} &H(Y|X, \theta, X^n = x^n, Y^n = y^n) = \\ &\int H_B\left(P\left(Y=1|x,\theta\right)\right)p(\theta|x^n, y^n, x)d\theta p(x)dx \end{aligned} \tag{A.27}$$

For BAC, the binary entropy conditioned on a specific x and θ, can be written as,

$$\begin{aligned} &H(Y|x, x^n, y^n) = \\ &H_B\left(\left(q\left(1 - F_{\theta|x^n,y^n}\left(x\right)\right) + (1-p)F_{\theta|x^n,y^n}(x)\right)\right) \end{aligned} \tag{A.28}$$

$$H(Y|x, \theta, x^n, y^n) = H_B\left(q\delta\left(x \le \theta\right) + (1-p)\delta\left(x > \theta\right)\right) \tag{A.29}$$

Therefore,

$$\begin{aligned} \lim_{n\to\infty} H(Y|X, x^n, y^n) = &\int_0^{\theta_1} H_B\left(q\right)p(x)dx \\ + &\int_{\theta_1+2^{-nC_W}}^1 H_B\left(1-p\right)p(x)dx \\ + &\int_{\theta_1}^{\theta_1+2^{-nC_W}} H_B\left(q(1-F_\theta(x)) + (1-p)F_\theta(x)\right)p(x)dx \end{aligned} \tag{A.30}$$

where θ_1 is estimated using (A.25) for a division of the interval $(0, 1)$ to bins on length 2^{-nC_W}.

Similarly,

$$\begin{aligned}\lim_{n\to\infty} H(Y|X,\theta,x^n,y^n) \geq \int_0^{\theta_1} H_B(q)\, p(x)dx + \\ \int_{\theta_1+2^{-nC_W}}^{1} H_B(1-p)\, p(x)dx\end{aligned} \tag{A.31}$$

Therefore the desired mutual information can be upper bounded by,

$$0 \leq \lim_{n\to\infty} I(\theta;Y|X,X^n,Y^n) \leq \alpha 2^{-nC_W} \tag{A.32}$$

This concludes the proof that active learning via PM achieves exponential decay and the important takeaway here is that the decay factor is dependent on the channel and the input distribution which achieved the capacity.

□

A.5 Proof of Theorem 5

Proof. Since (x^n, y^n) were selected using PM, then based on the results from [54], the posterior satisfies:

$$\lim_{n\to\infty} \sup_{\theta_1} \int_{\theta_1}^{\theta_1+2^{-nC_W}} p(\theta|x^n,y^n)d\theta = 1 \tag{A.33}$$

Using Bayes,

$$\lim_{n\to\infty} \frac{1}{Z} \sup_{\theta_1} \int_{\theta_1}^{\theta_1+2^{-nC_W}} p(y^n|x^n,\theta)\pi_u(\theta)d\theta = 1 \tag{A.34}$$

where $Z = \int p(y^n|x^n,\theta)\pi_u(\theta)d\theta$.

We define the interval $A = [\theta_1, \theta_1 + 2^{-nC_W}]$ and thus:

$$\lim_{n\to\infty} \int_{\theta\in A^c} p(y^n|x^n,\theta)\pi_u(\theta)d\theta = 0 \tag{A.35}$$

where the set A^c is the complementary set to A.

For the capacity achieving prior $\pi^*(\theta)$, a given training set size n and using Hölder's inequality:

$$\begin{aligned}0 \leq \int_{\theta\in A^c} p(y^n|x^n,\theta)\frac{\pi_u(\theta)\pi^*(\theta)}{\pi_u(\theta)}d\theta \leq \\ \int_{\theta\in A^c} p(y^n|x^n,\theta)\pi_u(\theta)d\theta \int_{\theta\in A^c} \frac{\pi^*(\theta)}{\pi_u(\theta)}d\theta\end{aligned} \tag{A.36}$$

Based on the multiplication law for limits and since the two limits exist, then:

$$\begin{aligned}0 \leq \lim_{n\to\infty} \int_{\theta\in A^c} p(y^n|x^n,\theta)\pi^*(\theta)d\theta \leq \\ \lim_{n\to\infty} \int_{\theta\in A^c} p(y^n|x^n,\theta)\pi_u(\theta)d\theta \lim_{n\to\infty} \int_{\theta\in A^c} \frac{\pi^*(\theta)}{\pi_u(\theta)}d\theta\end{aligned} \tag{A.37}$$

Using (A.35):

$$\begin{aligned} &0 \leq \lim_{n\to\infty} \int_{\theta\in A^c} p(y^n|x^n,\theta)\pi^*(\theta)d\theta \leq \\ &0 \cdot \lim_{n\to\infty} \int_{\theta\in A^c} \frac{\pi^*(\theta)}{\pi_u(\theta)}d\theta \end{aligned} \tag{A.38}$$

Since $\lim_{n\to\infty} \int_{\theta\in A^c} \frac{\pi^*(\theta)}{\pi_u(\theta)}d\theta$ exists and is finite for any probability distribution $\pi^*(\theta)$, then

$$\lim_{n\to\infty} \int_{\theta\in A^c} p(y^n|x^n,\theta)\pi_u(\theta)d\theta = 0$$

.

□

A.6 Proof of Theorem 6

Proof. Assume there is a homogeneous hyper-plane separating two complementary volumes in $\mathbb{R}^d$. This hyper-plane is defined by a unit length normal vector $\underline{w}$ which can be described by its spherical coordinates $\underline{\theta}$.

The idea of SPM is to successively estimate the spherical coordinates of $\underline{w}$ using PM, one coordinate at a time. In the first iteration, the spherical coordinate, θ_{d-1} is estimated and used for the estimation of the next spherical coordinate, θ_{d-2}. This process repeats until all the coordinates are estimated.

A.6.1 SPM Flow

In the first step of Algorithm 1, which corresponds to θ_{d-1}, SPM searches for the intersection point between the hyper plane defined by $\underline{w}$ and an arc, $\underline{r}(\phi)$ defined by the following description:

$$\underline{r}(\phi) = [sin(\phi), cos(\phi), 0, 0, ..., 0]$$

for $\phi \in (0, \pi)$.

This problem is a 1-dimensional noisy barrier model on the interval $(0, \pi)$, thus PM will query points in this interval and provide an estimate of the intersection point. The estimated intersection point, $\underline{x}_n^{d-1}$, after n training points can be described as:

$$\underline{x}_n^{d-1} = [sin(\phi_n), cos(\phi_n), 0, 0, ..., 0] \tag{A.39}$$

where ϕ_n is final queried point (angle) in the interval $(0, \pi)$.

The relation between ϕ_n and the estimate $\hat{\theta}_{d-1}$ of the spherical coordinate θ_{d-1} (of $\underline{w}$), is given by:

$$\hat{\theta}_{d-1} = \phi_n + \frac{\pi}{2} \tag{A.40}$$

Using (A.25), the following holds:

$$\lim_{n\to\infty} p(\theta_{d-1}|\underline{x}^n, y^n) = 2^{nC_W} \tag{A.41}$$

for any $\theta_{d-1} \in [\hat{\theta}_{d-1} - 2^{-nC_W - 1}, \hat{\theta}_{d-1} + 2^{-nC_W - 1}]$

In the next iteration of step 4 in Algorithm 1, the intersection between the hyperplane and the arc, $\underline{r}(\phi)$:

$$\underline{r}(\phi) = [sin(\hat{\theta}_{d-1})sin(\phi), cos(\hat{\theta}_{d-1})sin(\phi), cos(\phi), 0, 0, ..., 0]$$

for $\phi \in (0, \pi)$

The estimated intersection point after n training points:

$$\begin{aligned}\underline{x}_n^{d-2} = [&sin(\hat{\theta}_{d-1})sin(\phi_n), cos(\hat{\theta}_{d-1})sin(\phi_n),\\ &cos(\phi_n), 0, 0, ..., 0]\end{aligned}$$

Again, the estimated spherical coordinate is:

$$\hat{\theta}_{d-2} = \phi_n + \frac{\pi}{2} \tag{A.42}$$

This process goes on for all the spherical coordinates.

A.6.2 Proof Idea

Now that we have detailed the mechanism generating the estimates for the spherical coordinates, we can show how the active learning criterion decays for this training set selection policy. The main idea is to show that most of the probability mass of the joint posterior for the spherical coordinates reside inside a narrow enough cone in space, such that the active learning criterion decays exponentially fast to zero.

The active learning criterion, which is the conditional mutual information, is a difference of the conditional entropy of the test label Y given the training and test feature X:

$$\begin{aligned}&H(Y|\underline{X}, \underline{X}^{dn} = \underline{x}^{dn}, Y^{dn} = y^{dn}) =\\ &\int H_B\left(\int P(Y=1|\underline{x}, \underline{\theta}) p(\underline{\theta}|\underline{x}^{dn}, y^{dn}) d\underline{\theta}\right) p(\underline{x}) d\underline{x}\end{aligned} \tag{A.43}$$

and the conditional entropy of the test label Y given the training, test feature X and model parameter θ:

$$\begin{aligned}&H(Y|\underline{X}, \underline{\theta}, \underline{X}^{dn} = \underline{x}^{dn}, Y^{dn} = y^{dn}) =\\ &\int H_B\left(P(Y=1|\underline{x}, \underline{\theta})\right) p(\underline{\theta}|\underline{x}^{dn}, y^{dn}) d\underline{\theta} p(\underline{x}) d\underline{x}\end{aligned} \tag{A.44}$$

The spherical coordinates posterior can be decomposed using the chain rule for probabilities,

$$p(\underline{\theta}|\underline{x}^{n_T}, y^{n_T}) = \Pi_{i=d-1}^{1} p(\theta_i|\underline{\theta}_{i+1}^{d-1}, \underline{x}^{n_T}, y^{n_T}) \tag{A.45}$$

where $n_T = dn$.

We will now concentrate on the individual posteriors and show that they concentrate to the correct spherical coordinates fast.

A.6.3 Posterior for θ_{d-2}

For simplicity, we will first compute the posterior $p(\theta_{d-2}|\theta_{d-1}, \underline{x}^{n_T}, y^{n_T})$. After running the PM scheme for θ_{d-2}, all normal vectors, $\underline{w}$, which are possible candidates for the true normal vector, must satisfy the following equality with the estimated threshold point $\underline{x}_n^{d-2}$:

$$\lim_{n\to\infty} \Pr\left(\left|\underline{w}^T \underline{x}_n^{d-2}\right| \leq 2^{-nI}|\theta_{d-1}, \underline{x}^n, y^n\right) = 1$$

This equality basically creates a constraint on the possible values θ_{d-2}, can take and we can explicitly write this as:

$$\begin{aligned}\left|\underline{w}^T \underline{x}_n^{d-2}\right| = |\, &\sin(\hat{\theta}_{d-1}) sin(\phi_n) sin(\theta_{d-1}) sin(\theta_{d-2}) \\ &+ cos(\hat{\theta}_{d-1}) sin(\phi_n) cos(\theta_{d-1}) sin(\theta_{d-2}) \\ &+ cos(\phi_n) cos(\theta_{d-2})| \leq 2^{-nI}\end{aligned}$$

which can be written as:

$$|sin(\phi_n) sin(\theta_{d-2}) \gamma_{d-1} + cos(\phi_n) cos(\theta_{d-2})| \leq 2^{-nI} \tag{A.46}$$

where,

$$\gamma_{d-1} = sin(\theta_{d-1}) sin(\hat{\theta}_{d-1}) + cos(\hat{\theta}_{d-1}) cos(\theta_{d-1}) \tag{A.47}$$

We note that γ_{d-1} is an inner product between two unit length vectors and thus:

$$\gamma_{d-1} = \cos(\theta_{d-1} - \hat{\theta}_{d-1})$$

and according to (A.41), with probability approaching to 1 as n goes to infinity, $\gamma_{d-1} \leq \cos(2^{-nC_W})$. We also note that since 2^{-nC_W} is small then we can approximate γ_{d-1} using its Taylor expansion:

$$\gamma_{d-1} \approx 1 - \frac{2^{-2nC_W}}{2} \tag{A.48}$$

Therefore we can approximate (A.46) as,

$$\left|sin(\phi_n) sin(\theta_{d-2}) \left(1 - \frac{2^{-2nI}}{2}\right) + cos(\phi_n) cos(\theta_{d-2})\right| \leq 2^{-nI} \tag{A.49}$$

This is equivalent to:

$$\left|\cos(\phi_n - \theta_{d-2}) - \frac{2^{-2nI}}{2} \sin(\phi_n) sin(\theta_{d-2})\right| \leq 2^{-nI} \tag{A.50}$$

We will use the reverse triangle inequality and get:

$$\left||\cos(\phi_n - \theta_{d-2})| - \left|\frac{2^{-2nI}}{2} \sin(\phi_n) sin(\theta_{d-2})\right|\right| \leq 2^{-nI} \tag{A.51}$$

Therefore,

$$|\cos(\phi_n - \theta_{d-2})| \leq 2^{-nI} + \frac{2^{-2nI}}{2}$$

For large enough n, we can expand cosine around $\frac{\pi}{2}$ and get that the angles θ_{d-2} satisfy:

$$|\hat{\theta}_{d-2} - \theta_{d-2}| \leq 2^{-nI} + \frac{2^{-2nI}}{2} \tag{A.52}$$

Which basically means that:

$$\lim_{n\to\infty} \Pr\left(|\hat{\theta}_{d-2} - \theta_{d-2}| \leq 2^{-nI} + \frac{2^{-2nI}}{2} |\theta_{d-1}, \underline{x}^n, y^n\right) = 1 \tag{A.53}$$

which basically means for large enough n (d is fixed):

$$\lim_{n\to\infty} \Pr\left(|\hat{\theta}_{d-2} - \theta_{d-2}| \leq 2^{-nI} |\theta_{d-1}, \underline{x}^n, y^n\right) \approx 1 \tag{A.54}$$

Therefore, we approximately get the same condition as in (A.41).

A.6.4 Posterior for θ_i

We can now move to the general case of θ_i. We will show using recursion, that the posterior concentrates to the correct value appropriately. The final threshold point after n labeling operations for the i'th spherical coordinate is defined as:

$$\begin{aligned}\underline{x}_n^i = {} & [\sin(\phi_n)\Pi_{l=1,l\neq i}^{d-1} \sin(\hat{\theta}_l), \cos(\hat{\theta}_{d-1}) \sin(\phi_n)\Pi_{l=1,l\neq i}^{d-2} \sin(\hat{\theta}_l), \\ & ..., \cos(\phi_n)\Pi_{l=1}^{i-1} \sin(\hat{\theta}_l), ..., \cos(\hat{\theta}_1)]\end{aligned}$$

Again, due to PM, the following holds:

$$\lim_{n\to\infty} \Pr\left(\left|\underline{w}^T \underline{x}_n^i\right| \leq 2^{-nI} |\underline{\theta}_{i+1}^{d-1}, \underline{x}^{n_T}, y^{n_T}\right) = 1$$

We define the following recursion rule:

$$\gamma_i = sin(\theta_i) sin(\hat{\theta}_i)\gamma_{i+1} + cos(\hat{\theta}_i) cos(\theta_i) \tag{A.55}$$

with (A.47) as the initial condition.
The inner product ,$\underline{w}^T \underline{x}_n^i$, can be written as:

$$\underline{w}^T \underline{x}_n^i = \sin(\phi_n) sin(\theta_i)\gamma_{i+1} + cos(\phi_n) cos(\theta_i) \tag{A.56}$$

If we knew that $\gamma_{i+1} \approx 1 - \frac{2^{-2nC_W}}{2}$ we could use the same arguments from the previous section to bound the posterior. Using (A.54), $\gamma_{d-2} \approx 1 - \frac{2^{-2nC_W}}{2}$ and applying (A.55) in recursion, we get:

$$\lim_{n\to\infty} \Pr\left(|\hat{\theta}_i - \theta_i| \leq 2^{-nI} |\theta_{i+1}^{d-1}, \underline{x}^{dn}, y^{dn}\right) \approx 1 \tag{A.57}$$

A.6.5 Asymptotic Decay of Mutual Information

Finally, we will use the posteriors computed in the previous sections to give an upper bound on the conditional mutual information. The multiplication of the posteriors, $p(\theta_i|\underline{\theta}_{i+1}^{d-1}, \underline{x}_i^{n_T}, y_i^{n_T})$, form a cone with probability approaching 1 in $\underline{x} \in \mathbb{R}^d$. The unit vector $\underline{\hat{w}}$ is a vector in the center of this cone. Using the results on $p(\theta_i|\underline{\theta}_{i+1}^{d-1}, \underline{x}_i^{n_T}, y_i^{n_T})$, (A.28) and (A.29), we can compute upper bounds on the conditional mutual information.

For the BAC,

$$\begin{aligned} &P(Y=1|\underline{x}, \underline{x}^{n_T}, y^{n_T}) = \\ &= q \int 1\left(\underline{x}^T \underline{w} \le 0\right) \Pi_{i=1}^d p(\theta_i|\underline{\theta}^{i-1}, \underline{x}_i^{n_T}, y_i^{n_T}) d\underline{\theta} + \\ &+ (1-p) \int 1\left(\underline{x}^T \underline{w} \ge 0\right) \Pi_{i=1}^d p(\theta_i|\underline{\theta}^{i-1}, \underline{x}_i^{n_T}, y_i^{n_T}) d\underline{\theta} \end{aligned} \tag{A.58}$$

Therefore,

$$\begin{aligned} &\lim_{n\to\infty} H(Y|\underline{X}, \underline{x}^n, y^n) = \int_{\frac{\langle \underline{x}, \underline{\hat{w}}\rangle}{|\underline{x}|} \le -2^{-nC_W}} H_B\left(q\right) p(\underline{x}) d\underline{x} + \\ &\int_{\frac{\langle \underline{x}, \underline{\hat{w}}\rangle}{|\underline{x}|} > 2^{-nC_W}} H_B\left(1-p\right) p(\underline{x}) d\underline{x} + \int_{\frac{|\langle \underline{x}, \underline{\hat{w}}\rangle|}{|\underline{x}|} \le 2^{-nC_W}} \\ &H_B\left(q(1 - F_{\underline{\theta}|\underline{x}^n, y^n}(\underline{x})) + (1-p) F_{\underline{\theta}|\underline{x}^n, y^n}(\underline{x})\right) p(\underline{x}) d\underline{x} \end{aligned} \tag{A.59}$$

Similarly,

$$\begin{aligned} &\lim_{n\to\infty} H(Y|\underline{X}, \theta, \underline{x}^n, y^n) \ge \int_{\frac{\langle \underline{x}, \underline{\hat{w}}\rangle}{|\underline{x}|} \le -2^{-nC_W}} H_B\left(q\right) p(\underline{x}) d\underline{x} + \\ &\int_{\frac{\langle \underline{x}, \underline{\hat{w}}\rangle}{|\underline{x}|} > 2^{-nC_W}} H_B\left(1-p\right) p(\underline{x}) d\underline{x} \end{aligned} \tag{A.60}$$

Therefore the desired mutual information can be upper bounded by,

$$0 \le \lim_{n\to\infty} I(\theta; Y|\underline{X}, \underline{X}^n, Y^n) \le \alpha 2^{-\frac{n_T}{d} C_W} \tag{A.61}$$

□

Appendix B

Proofs for Part II

B.1 Proof of Theorem 8

Proof. In this proof we will address the two forms of IAL criterion: (4.6) and (4.7). First we look at the greedy criterion defined in Eq. (4.6) which can be written as:

$$C_{n|n-1} = \min_{x_n \in \mathbb{X}} \max_{y_n \in \mathbb{Y}} \sum_{v \in \mathbb{V}} \int_{u \in \mathbb{U}} p\left(v|u, \hat{\theta}^n\right) du \tag{B.1}$$

where $\hat{\theta}^n$ is the maximum likelihood estimation based on training and test data:

$$\hat{\theta}^n = \arg\max_{\theta \in \Theta} p\left(y^n, v|x^n, u, \theta\right) \tag{B.2}$$

In a greedy scheme, z^{n-1} is assumed known and the likelihood function, $p\left(y^n, v|x^n, u, \theta\right)$, is basically a multiplication of indicator functions resulting in a rectangle window function around the correct barrier. There are multiple solutions for the maximum likelihood estimator and we select the mid point of the posterior's support, which is uniformly distributed.

We can write the likelihood for z^{n-1} as:

$$p\left(y^{n-1}|x^{n-1}, \theta\right) \sim \mathbb{1}\left(\theta \geq \theta_{min}^{n-1}\right) \mathbb{1}\left(\theta < \theta_{max}^{n-1}\right) \tag{B.3}$$

where θ_{min}^{n-1} and θ_{max}^{n-1} represent the support of the posterior on θ given x^{n-1}, y^{n-1}.

Once a new feature point x_n is selected (any point in the support of $p\left(y^{n-1}|x^{n-1}, \theta\right)$), then based on its label y_n which can be arbitrary (we train for the worst), the likelihood window function gets split again.

For $y_n = 1 - \alpha$:

$$p\left(y^n|x^n, \theta\right) \sim \mathbb{1}\left(\theta \geq x_n\right) \mathbb{1}\left(\theta < \theta_{max}^{n-1}\right) \tag{B.4}$$

For $y_n = \alpha$:

$$p\left(y^n|x^n, \theta\right) \sim \mathbb{1}\left(\theta \geq \theta_{min}^{n-1}\right) \mathbb{1}\left(\theta < x_n\right) \tag{B.5}$$

For the computation of (4.6), which is the pNML normalization factor, we are interested in the behaviour of the distribution $p\left(v|u, \hat{\theta}^n\right)$ for different test data points, u:

For $y_n = 1 - \alpha$:

$$\begin{aligned} \int_0^1 \sum_{v=0}^{1} p\left(v|u, \hat{\theta}^n\right) du &= \int_0^{x_n} \sum_{v \in \mathbb{V}} p\left(v|u, \hat{\theta}^n\right) du \\ + \int_{x_n}^{\hat{\theta}^{n-1}_{max}} \sum_{v \in \mathbb{V}} p\left(v|u, \hat{\theta}^n\right) du &+ \int_{\hat{\theta}^{n-1}_{max}}^{1} \sum_{v \in \mathbb{V}} p\left(v|u, \hat{\theta}^n\right) du \end{aligned} \tag{B.6}$$

For $y_n = 1 - \alpha$, if the test feature, u, satisfies $\theta^{n-1}_{max} \leq u$ then $p\left(v = 1|u, \theta\right) = 1$ and if $x_n \leq u$, then $p\left(v = 1|u, \theta\right) = 0$.

Therefore,

$$\begin{aligned} \int_0^1 \sum_{v=0}^{1} p\left(v|u, \hat{\theta}^n\right) du &= |x_n| \\ + \int_{x_n}^{\hat{\theta}^{n-1}_{max}} \sum_{v \in \mathbb{V}} p\left(v|u, \hat{\theta}^n\right) du &+ |1 - \hat{\theta}^{n-1}_{max}| \end{aligned} \tag{B.7}$$

Now we observe that when $x_n \leq u \leq \hat{\theta}^{n-1}_{max}$:

$$\begin{aligned} &p\left(v = 1|u, \hat{\theta}^n(v = 1, u, x^n, y^n)\right) = \\ &p\left(v = 0|u, \hat{\theta}^n(v = 0, u, x^n, y^n)\right) = 1 \end{aligned} \tag{B.8}$$

Thus (B.7) can be expressed as:

$$\int_0^1 \sum_{v=0}^{1} p\left(v|u, \hat{\theta}^n\right) du = |x_n| + 2|x_n - \theta^{n-1}_{max}| + |1 - \hat{\theta}^{n-1}_{max}| \tag{B.9}$$

For $y_n = \alpha$:

$$\begin{aligned} \int_0^1 \sum_{v=0}^{1} p\left(v|u, \hat{\theta}^n\right) du &= \int_0^{\hat{\theta}^{n-1}_{min}} \sum_{v \in \mathbb{V}} p\left(v|u, \hat{\theta}^n\right) du \\ + \int_{\hat{\theta}^{n-1}_{min}}^{x_n} \sum_{v \in \mathbb{V}} p\left(v|u, \hat{\theta}^n\right) du &+ \int_{x_n}^{1} \sum_{v \in \mathbb{V}} p\left(v|u, \hat{\theta}^n\right) du \end{aligned} \tag{B.10}$$

Therefore,

$$\begin{aligned} \int_0^1 \sum_{v=0}^{1} p\left(v|u, \hat{\theta}^n\right) du &= |\hat{\theta}^{n-1}_{min}| \\ + \int_{\hat{\theta}^{n-1}_{min}}^{x_n} \sum_{v \in \mathbb{V}} p\left(v|u, \hat{\theta}^n\right) du &+ |1 - x_n| \end{aligned} \tag{B.11}$$

Finally,

$$\int_0^1 \sum_{v=0}^{1} p\left(v|u, \hat{\theta}^n\right) du = |\hat{\theta}^{n-1}_{min}| + 2|\theta^{n-1}_{min} - x_n| + |1 - x_n| \tag{B.12}$$

We wish to find x_n which minimizes the following expression:

$$\max_{y_n \in \mathbb{Y}} \int_{u \in \mathbb{U}} \sum_{v \in \mathbb{V}} p\left(v|u, \hat{\theta}^n\right) du = \max\{l_0, l_1\} \tag{B.13}$$

where:

$$l_0 = |1 - \theta_{max}^{n-1}| + 2|x_n - \theta_{max}^{n-1}| + |x_n| = 1 + |x_n - \theta_{max}^{n-1}|$$

and

$$l_1 = |\theta_{min}^{n-1}| + 2|\theta_{min}^{n-1} - x_n| + |1 - x_n| = 1 + |\theta_{min}^{n-1} - x_n|$$

For (4.6) the score is averaged over all possible v and u, then:

$$\begin{aligned} &\min_{x_n \in \mathbb{X}} \max_{y_n \in \mathbb{Y}} \int_{u \in \mathbb{U}} \sum_{v \in \mathbb{V}} p\left(v|u, \hat{\theta}^n\right) du \\ &= \min_{x_n \in \mathbb{X}} \max\{|x_n - \theta_{max}^{n-1}|, |\theta_{min}^{n-1} - x_n|\} \end{aligned} \tag{B.14}$$

Therefore, the point x_n which minimizes the maximal length is the mid point of the interval $\left[\theta_{min}^{n-1}, \theta_{max}^{n-1}\right]$

In a very similar way we can prove that IAL as defined in (4.7) behaves the same. The greedy criterion defined in (4.7) can be written as:

$$C_{n|n-1} = \min_{x_n \in \mathbb{X}} \max_{y_n \in \mathbb{Y}} \int_{x \in \mathbb{X}} \log \sum_{y \in \mathbb{Y}} p\left(y|x, \hat{\theta}^n\right) dx \tag{B.15}$$

where y, x and $\hat{\theta}^n$ are the test label,feature and maximum likelihood estimation based on training and test data respectively:

$$\hat{\theta}^n = \arg\max_{\theta \in \Theta} p\left(y^n, y|x^n, x, \theta\right) \tag{B.16}$$

For the separable case, $p\left(y^n, y|x^n, x, \theta\right)$, is basically a multiplication of indicator functions resulting in a rectangle window around the correct barrier. There are multiple solutions for the maximum likelihood estimator and we select the mid point of the posterior's support, which is uniformly distributed.

We can write the likelihood for z^{n-1} as:

$$p\left(y^{n-1}|x^{n-1}, \theta\right) \propto \mathbb{1}\left(\theta \geq \theta_{min}^{n-1}\right) \mathbb{1}\left(\theta < \theta_{max}^{n-1}\right) \tag{B.17}$$

where θ_{min}^{n-1} and θ_{max}^{n-1} represent the support of the posterior on θ given x^{n-1}, y^{n-1}.

Once a new feature point x_n is selected (any point in the support of $p\left(y^{n-1}|x^{n-1}, \theta\right)$), then based on its label y_n which can be arbitrary (we train for the worst), the likelihood window function gets split again.

For $y_n = 1 - \alpha$:

$$p\left(y^n|x^n, \theta\right) \propto \mathbb{1}\left(\theta \geq x_n\right) \mathbb{1}\left(\theta < \theta_{max}^{n-1}\right) \tag{B.18}$$

For $y_n = \alpha$:

$$p\left(y^n|x^n, \theta\right) \propto \mathbb{1}\left(\theta \geq \theta_{min}^{n-1}\right) \mathbb{1}\left(\theta < x_n\right) \tag{B.19}$$

For the computation of (4.6), we are interested in the distribution $p\left(y|x,\hat{\theta}^n\right)$ for different test data points, x:

For $y_n = 1-\alpha$:

$$\begin{aligned}\int_0^1 \log\sum_{y=0}^{1} p\left(y|x,\hat{\theta}^n\right)dx &= \int_0^{x_n}\log\sum_{y\in\mathbb{Y}} p\left(y|x,\hat{\theta}^n\right)dx\\ +\int_{x_n}^{\hat{\theta}^{n-1}_{max}}\log\sum_{y\in\mathbb{Y}} p\left(y|x,\hat{\theta}^n\right)dx &+ \int_{\hat{\theta}^{n-1}_{max}}^{1}\log\sum_{y\in\mathbb{Y}} p\left(y|x,\hat{\theta}^n\right)dx\end{aligned} \tag{B.20}$$

For $y_n = 1-\alpha$, if the test feature, x, satisfies $\theta^{n-1}_{max}\le x$ then $p\left(y=1|x,\theta\right)=\alpha$ and if $x_n\ge x$, then $p\left(y=1|x,\theta\right)=1-\alpha$.

Therefore,

$$\int_0^1 \log\sum_{y=0}^{1} p\left(y|x,\hat{\theta}^n\right)dx = \int_{x_n}^{\hat{\theta}^{n-1}_{max}}\log\sum_{y\in\mathbb{Y}} p\left(y|x,\hat{\theta}^n\right)dx \tag{B.21}$$

Now we observe that when $x_n\le x\le\hat{\theta}^{n-1}_{max}$:

$$\begin{aligned}&p\left(y=1|x,\hat{\theta}^n(y=1,x,x^n,y^n)\right)\\ &= p\left(y=0|x,\hat{\theta}^n(y=0,x,x^n,y^n)\right)=1\end{aligned} \tag{B.22}$$

Thus (B.21) can be expressed as:

$$\int_0^1 \log\sum_{y=0}^{1} p\left(y|x,\hat{\theta}^n\right)dx = |x_n-\theta^{n-1}_{max}| \tag{B.23}$$

For $y_n=\alpha$:

$$\begin{aligned}\int_0^1 \log\sum_{y=0}^{1} p\left(y|x,\hat{\theta}^n\right)dx &= \int_0^{\hat{\theta}^{n-1}_{min}}\log\sum_{y\in\mathbb{Y}} p\left(y|x,\hat{\theta}^n\right)dx+\\ \int_{\hat{\theta}^{n-1}_{min}}^{x_n}\log\sum_{y\in\mathbb{Y}} p\left(y|x,\hat{\theta}^n\right)dx &+ \int_{x_n}^{1}\log\sum_{y\in\mathbb{Y}} p\left(y|x,\hat{\theta}^n\right)dx\end{aligned} \tag{B.24}$$

Finally,

$$\int_0^1 \log\sum_{y=0}^{1} p\left(y|x,\hat{\theta}^n\right)dx = |\theta^{n-1}_{min}-x_n| \tag{B.25}$$

We wish to find x_n which minimizes the following expression:

$$\max_{y_i\in\mathbb{Y}}\int_{x\in\mathbb{X}}\log\sum_{y\in\mathbb{Y}} p\left(y|x,\hat{\theta}^n\right)dx = \max\{l_0,l_1\} \tag{B.26}$$

where:

$$l_0 = |x_n-\theta^{n-1}_{max}|$$

and

$$l_1 = |\theta_{min}^{n-1} - x_n|$$

For (4.6) the score is averaged over all possible y and x, then:

$$\begin{aligned} &\min_{x_n \in \mathbb{X}} \max_{y_n \in \mathbb{Y}} \int_{x \in \mathbb{X}} \log \sum_{y \in \mathbb{Y}} p\left(y|x, \hat{\theta}^n\right) dx = \\ &\min_{x_n \in \mathbb{X}} \max\{|x_n - \theta_{max}^{n-1}|, |\theta_{min}^{n-1} - x_n|\} \end{aligned} \tag{B.27}$$

Therefore, the point x_i which minimizes the maximal length is the mid point of the interval $\left[\theta_{min}^{n-1}, \theta_{max}^{n-1}\right]$

□

B.2 Proof of Theorem 9

Proof. Based on [57], the recurring equation for the estimated model parameter, θ, after incorporating the n'th data point x_n

$$\hat{\theta}^n = \hat{\theta}^{n-1} + \frac{\left(X_{n-1}^T X_{n-1} + \lambda^{-1} I\right)^{-1}}{1 + x_n^T \left(X_{n-1}^T X_{n-1} + \lambda^{-1} I\right) x_n} \left(y - x_n^T \hat{\theta}^{n-1}\right) \tag{B.28}$$

where $\hat{\theta}^{n-1}$ is the OLS estimator for θ given $n - 1$ data points which are aggregated in the matrix X_{n-1}.

After incorporating the test feature and label, $[x, y]$, one can write $\hat{\theta}$ (which will be used in IAL):

$$\hat{\theta} = \hat{\theta}^n + \frac{\left(X_n^T X_n + \lambda^{-1} I\right)^{-1}}{1 + x^T \left(X_n^T X_n + \lambda^{-1} I\right)^{-1} x} \left(y - x^T \hat{\theta}^n\right) \tag{B.29}$$

Therefore, the genie using $\hat{\theta}$ can be expressed as:

$$p\left(y|x, \hat{\theta}\right) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}\left(y - x^T \hat{\theta}\right)^2\right) \tag{B.30}$$

The pNML normalization factor:

$$\Gamma = \int \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}\left(y - x^T \hat{\theta}\right)^2\right) dy \tag{B.31}$$

We recall that $\hat{\theta}$ is dependent on y and that is why the above integral does not equal to 1.

Plugging (B.29) in (B.31):

$$\Gamma = \int \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\left(y - x^T \hat{\theta}^n\right)^2}{2\sigma^2 \left(1 + x^T \left(X_n^T X_n + \lambda^{-1} I\right)^{-1} x\right)}\right) dy \tag{B.32}$$

Recall IAL based on (4.5):

$$C_n = \min_{x^n} \max_{y^n} \mathbb{E}_x \left(\int p\left(y|x, \hat{\theta}\right) dy \right) \tag{B.33}$$

Plugging (B.32) into (B.33):

$$C_n = \min_{x^n} \max_{y^n} \mathbb{E}_x \left(1 + x^T \left(X_n^T X_n + \lambda^{-1} I \right)^{-1} x \right) \tag{B.34}$$

Since there is no dependence on y^n and using the cyclic in-variance of the trace operator:

$$C_n = \min_{x^n} \mathrm{Tr} \left(X X^T \left(X_n^T X_n + \lambda^{-1} I \right)^{-1} \right) \tag{B.35}$$

where X is a matrix which is a concatenation of the test vectors x. □

B.3 Equivalence between EPIG and UAL

In this section we will prove that the criterion proposed by [68] is equivalent in some sense to the criterion of [27].

Proof. Assuming some prior $\pi(\theta)$, the UAL criterion is:

$$\hat{x}_i = \operatorname*{argmin}_{x_i} I(\theta; Y|X, x_i, Y_i, z^{n-1})$$

where X and Y are the test feature and label random variables.

The EPIG criterion is:

$$\hat{x}_i = \operatorname*{argmax}_{x_i} I(Y; Y_i|X, x_i, z^{n-1})$$

Using the mutual information chain rule we can write:

$$I(Y; Y_i, \theta|X, x_i, z^{n-1}) = I(Y; Y_i|X, x_i, z^{n-1}) + \\ I(Y; \theta|X, Y_i, x_i, z^{n-1})$$

But we can also use the chain rule in a different way:

$$I(Y; Y_i, \theta|X, x_i, z^{n-1}) = I(Y; \theta|X, x_i, z^{n-1}) + \\ I(Y; Y_i|X, x_i, \theta, z^{n-1})$$

We note that

$$I(Y; \theta|X, x_i, z^{n-1}) = I(Y; \theta|X, z^{n-1})$$

since there is no dependence of θ or Y on x_i without Y_i. Also, $I(Y; Y_i|X, x_i, \theta, z^{n-1}) = 0$, since given θ the test and train are independent.

Therefore, $I(Y; Y_i, \theta|X, x_i, z^{n-1})$ is not dependent on x_i and if we try to find x_i which minimizes $I(\theta; Y|X, x_i, Y_i, z^{n-1})$ (UAL), it will simultaneously maximize $I(Y; Y_i|X, x_i, z^{n-1})$ (EPIG).

□

Bibliography

[1] K. Wang, D. Zhang, Y. Li, R. Zhang, and L. Lin, "Cost-effective active learning for deep image classification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 12, pp. 2591–2600, 2016.

[2] R. M. Castro and R. D. Nowak, "Minimax bounds for active learning," *IEEE Transactions on Information Theory*, vol. 54, no. 5, pp. 2339–2353, 2008.

[3] M. Raginsky and A. Rakhlin, "Lower bounds for passive and active learning," pp. 1026–1034, 2011.

[4] S. Hanneke, *A bound on the label complexity of agnostic active learning*. Citeseer, 2007.

[5] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby, "Selective sampling using the query by committee algorithm," *Machine learning*, vol. 28, no. 2-3, pp. 133–168, 1997.

[6] M.-F. Balcan, A. Beygelzimer, and J. Langford, "Agnostic active learning," *Journal of Computer and System Sciences*, vol. 75, no. 1, pp. 78–89, 2009.

[7] M.-F. Balcan, A. Broder, and T. Zhang, "Margin based active learning," pp. 35–50, 2007.

[8] D. Cohn, L. Atlas, and R. Ladner, "Improving generalization with active learning," *Machine learning*, vol. 15, no. 2, pp. 201–221, 1994.

[9] S. Dasgupta, "Coarse sample complexity bounds for active learning," pp. 235–242, 2006.

[10] S. Dasgupta, D. J. Hsu, and C. Monteleoni, "A general agnostic active learning algorithm," pp. 353–360, 2008.

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

[12] S. Hanneke, "Theoretical foundations of active learning," CARNEGIE-MELLON UNIV PITTSBURGH PA MACHINE LEARNING DEPT, Tech. Rep., 2009.

[13] A. Beygelzimer, D. J. Hsu, J. Langford, and T. Zhang, "Agnostic active learning without constraints," pp. 199–207, 2010.

[14] V. Koltchinskii, "Rademacher complexities and bounding the excess risk in active learning," *Journal of Machine Learning Research*, vol. 11, no. Sep, pp. 2457–2485, 2010.

[15] M.-F. Balcan and P. Long, "Active and passive learning of linear separators under log-concave distributions," pp. 288–316, 2013.

[16] P. Awasthi, M. F. Balcan, and P. M. Long, "The power of localization for efficiently learning linear separators with noise," pp. 449–458, 2014.

[17] Y. Guo and R. Greiner, "Optimistic active-learning using mutual information." vol. 7, pp. 823–829, 2007.

[18] B. Settles, "Active learning literature survey," University of Wisconsin-Madison Department of Computer Sciences, Tech. Rep., 2009.

[19] N. Houlsby, F. Huszár, Z. Ghahramani, and M. Lengyel, "Bayesian active learning for classification and preference learning," *arXiv preprint arXiv:1112.5745*, 2011.

[20] M. Raginsky and A. Rakhlin, "Information-based complexity, feedback and dynamics in convex programming," *IEEE Transactions on Information Theory*, vol. 57, no. 10, pp. 7036–7056, 2011.

[21] D. J. MacKay, "Information-based objective functions for active data selection," *Neural computation*, vol. 4, no. 4, pp. 590–604, 1992.

[22] V. V. Fedorov, *Theory of optimal experiments*. Elsevier, 2013.

[23] Y. Gal, R. Islam, and Z. Ghahramani, "Deep bayesian active learning with image data," pp. 1183–1192, 2017.

[24] A. Alabduljabbar, A. Abusnaina, Ü. Meteriz-Yildiran, and D. Mohaisen, "Tldr: Deep learning-based automated privacy policy annotation with key policy highlights," in *Proceedings of the 20th Workshop on Workshop on Privacy in the Electronic Society*, 2021, pp. 103–118.

[25] N. Merhav and M. Feder, "Universal prediction," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2124–2147, 1998.

[26] S. Shayovitz and M. Feder, "Minimax active learning via minimal model capacity," in *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2019, pp. 1–6.

[27] ——, "Universal active learning via conditional mutual information minimization," *IEEE Journal on Selected Areas in Information Theory*, vol. 2, no. 2, pp. 720–734, 2021.

[28] ——, "Active learning for individual data via minimal stochastic complexity," in *2022 58th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2022, pp. 1–5.

[29] S. Shayovitz, K. Bibas, and M. Feder, “Deep individual active learning: Safeguarding against out-of-distribution challenges in neural networks,” *Entropy*, vol. 26, no. 2, p. 129, 2024.

[30] S. Shayovitz and M. Feder, “Active learning via predictive normalized maximum likelihood minimization,” *IEEE Transactions on Information Theory*, vol. 70, no. 8, pp. 5799–5810, 2024.

[31] Y. Fogel and M. Feder, “Universal batch learning with log-loss,” pp. 21–25, 2018.

[32] Y. Gal and Z. Ghahramani, “Dropout as a bayesian approximation: Representing model uncertainty in deep learning,” in *international conference on machine learning*. PMLR, 2016, pp. 1050–1059.

[33] A. Kirsch, T. Rainforth, and Y. Gal, “Test distribution-aware active learning: A principled approach against distribution shift and outliers,” *arXiv preprint arXiv:2106.11719*, 2021.

[34] R. G. Gallager, “Source coding with side information and universal coding,” 1979.

[35] D. Golovin and A. Krause, “Adaptive submodularity: Theory and applications in active learning and stochastic optimization,” *Journal of Artificial Intelligence Research*, vol. 42, pp. 427–486, 2011.

[36] A. Kirsch, J. van Amersfoort, and Y. Gal, “Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning,” pp. 7024–7035, 2019.

[37] F. Pukelsheim, *Optimal design of experiments*. SIAM, 2006.

[38] L. Chamon and A. Ribeiro, “Approximate supermodularity bounds for experimental design,” pp. 5403–5412, 2017.

[39] K. Yu, J. Bi, and V. Tresp, “Active learning via transductive experimental design,” pp. 1081–1088, 2006.

[40] C. E. Rasmussen, “Gaussian processes in machine learning,” pp. 63–71, 2003.

[41] P. Boyle, “Gaussian processes for regression and optimisation,” 2007.

[42] A. O’Hagan, “Curve fitting and optimal design for prediction,” *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 40, no. 1, pp. 1–24, 1978.

[43] D. J. MacKay *et al.*, “Introduction to gaussian processes,” *NATO ASI series F computer and systems sciences*, vol. 168, pp. 133–166, 1998.

[44] R. M. Neal, *Bayesian learning for neural networks*. Springer Science & Business Media, 2012, vol. 118.

[45] D. G. Krige, “A statistical approach to some basic mine valuation problems on the witwatersrand,” *Journal of the Southern African Institute of Mining and Metallurgy*, vol. 52, no. 6, pp. 119–139, 1951.

[46] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006, vol. 2, no. 3.

[47] C. K. Williams, "Prediction with gaussian processes: From linear regression to linear prediction and beyond," *Learning in graphical models*, pp. 599–621, 1998.

[48] T. P. Minka, "Expectation propagation for approximate bayesian inference," *arXiv preprint arXiv:1301.2294*, 2013.

[49] S. Yan and C. Zhang, "Revisiting perceptron: Efficient and label-optimal learning of halfspaces," pp. 1056–1066, 2017.

[50] P. Massart, É. Nédélec *et al.*, "Risk bounds for statistical learning," *The Annals of Statistics*, vol. 34, no. 5, pp. 2326–2366, 2006.

[51] P. Awasthi, M.-F. Balcan, N. Haghtalab, and H. Zhang, "Learning and 1-bit compressed sensing under asymmetric noise," pp. 152–192, 2016.

[52] O. Shayevitz and M. Feder, "Communication with feedback via posterior matching," pp. 391–395, 2007.

[53] M. Horstein, "Sequential transmission using noiseless feedback," *IEEE Transactions on Information Theory*, vol. 9, no. 3, pp. 136–143, 1963.

[54] O. Shayevitz and M. Feder, "Optimal feedback communication via posterior matching," *IEEE Transactions on Information Theory*, vol. 57, no. 3, pp. 1186–1222, 2011.

[55] K. Bibas, Y. Fogel, and M. Feder, "A new look at an old problem: A universal learning approach to linear regression," 2019.

[56] A. Zhou and S. Levine, "Amortized conditional normalized maximum likelihood: Reliable out of distribution uncertainty estimation," in *International Conference on Machine Learning*. PMLR, 2021, pp. 12 803–12 812.

[57] K. Bibas and M. Feder, "Distribution free uncertainty for the minimum norm solution of over-parameterized linear regression," *arXiv preprint arXiv:2102.07181*, 2021.

[58] F. E. Rosas, P. A. Mediano, and M. Gastpar, "Learning, compression, and leakage: Minimising classification error via meta-universal compression principles," in *2020 IEEE Information Theory Workshop (ITW)*. IEEE, 2021, pp. 1–5.

[59] J. Rissanen and T. Roos, "Conditional nml universal models," in *2007 Information Theory and Applications Workshop*. IEEE, 2007, pp. 337–341.

[60] T. Roos and J. Rissanen, "On sequentially normalized maximum likelihood models," *Compare*, vol. 27, no. 31, p. 256, 2008.

[61] K. Bibas and M. Feder, "The predictive normalized maximum likelihood for over-parameterized linear regression with norm constraint: Regret and double descent," *arXiv preprint arXiv:2102.07181*, 2021.

[62] F. E. Rosas, P. A. Mediano, and M. Gastpar, “Learning, compression, and leakage: Minimizing classification error via meta-universal compression principles,” *arXiv preprint arXiv:2010.07382*, 2020.

[63] J. Fu and S. Levine, “Offline model-based optimization via normalized maximum likelihood estimation,” 2021.

[64] M. Karzand and R. D. Nowak, “Maximin active learning in overparameterized model classes,” *IEEE Journal on Selected Areas in Information Theory*, vol. 1, no. 1, pp. 167–177, 2020.

[65] R. D. Nowak, “The geometry of generalized binary search,” *IEEE Transactions on Information Theory*, vol. 57, no. 12, pp. 7893–7906, 2011.

[66] C. E. Rasmussen and H. Nickisch, “Gaussian processes for machine learning (gpml) toolbox,” *The Journal of Machine Learning Research*, vol. 11, pp. 3011–3015, 2010.

[67] “Usps hand written data set,” http://www.gaussianprocess.org/gpml/data/.

[68] F. B. Smith, A. Kirsch, S. Farquhar, Y. Gal, A. Foster, and T. Rainforth, “Prediction-oriented bayesian active learning,” in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2023.

[69] D. Yoo and I. S. Kweon, “Learning loss for active learning,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019.

[70] S. Sinha, S. Ebrahimi, and T. Darrell, “Variational adversarial active learning,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[71] K. Bibas, Y. Fogel, and M. Feder, “Deep pnml: Predictive normalized maximum likelihood for deep neural networks,” *arXiv preprint arXiv:1904.12286*, 2019.

[72] W. J. Maddox, P. Izmailov, T. Garipov, D. P. Vetrov, and A. G. Wilson, “A simple baseline for bayesian uncertainty in deep learning,” *Advances in neural information processing systems*, 2019.

[73] E. Daxberger, A. Kristiadi, A. Immer, R. Eschenhagen, M. Bauer, and P. Hennig, “Laplace redux-effortless bayesian deep learning,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 20 089–20 103, 2021.

[74] A. G. Wilson, Z. Hu, R. R. Salakhutdinov, and E. P. Xing, “Stochastic variational deep kernel learning,” *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[75] C. Zhang, J. Bütepage, H. Kjellström, and S. Mandt, “Advances in variational inference,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 8, pp. 2008–2026, 2018.

[76] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, “Variational inference: A review for statisticians,” *Journal of the American statistical Association*, 2017.

[77] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley, “Stochastic variational inference,” *Journal of Machine Learning Research*, 2013.

[78] H. U. Simon, “An almost optimal pac algorithm,” in *Conference on Learning Theory*. PMLR, 2015, pp. 1552–1563.

[79] L. Deng, “The mnist database of handwritten digit images for machine learning research [best of the web],” *IEEE signal processing magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[80] G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik, “Emnist: Extending mnist to handwritten letters,” in *2017 international joint conference on neural networks (IJCNN)*. IEEE, 2017, pp. 2921–2926.

[81] A. Krizhevsky, V. Nair, and G. Hinton, “The cifar-10 dataset,” *online: http://www. cs. toronto. edu/kriz/cifar. html*, 2014.

[82] H. Xiao, K. Rasul, and R. Vollgraf, “Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms,” *arXiv preprint arXiv:1708.07747*, 2017.

[83] P. Sermanet, S. Chintala, and Y. LeCun, “Convolutional neural networks applied to house numbers digit classification,” in *Proceedings of the 21st international conference on pattern recognition (ICPR2012)*. IEEE, 2012, pp. 3288–3291.

[84] K.-H. Huang, “Deepal: Deep active learning in python,” *arXiv preprint arXiv:2111.15258*, 2021.

[85] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[86] J. v. Neumann, “Zur theorie der gesellschaftsspiele,” *Mathematische annalen*, vol. 100, no. 1, pp. 295–320, 1928.

[87] E. Telatar, “Capacity of multi-antenna gaussian channels,” *European transactions on telecommunications*, vol. 10, no. 6, pp. 585–595, 1999.

תוכן העניינים

תקציר

למידה פעילה היא פרדיגמת למידה שבה נתוני האימון נבחרים באופן אקטיבי ומכוון. המטרה העיקרית היא לייעל את ביצועי המודל על ידי מזעור מספר הדגימות המתוייגות. אסטרטגיות מובילות מבוססות על ההנחה שלמאגר האימון יש את אותה התפלגות כמו מערך הבדיקות, מה שלא יכול להיות המקרה ביישומים רגישים לפרטיות שבהם לא ניתן לתייג נתוני משתמשים.

בחלק הראשון של המחקר, אנו מניחים שהדאטא מקיים את הנחת הסטוכסטיות והדאטא מתפלג לפי פילוג כלשהו. קריטריון חדש של למידה פעילה תיאורטית מוצע בהתבסס על משפט קיבולת יתירות. קריטריון זה גורם באופן טבעי לפשרה של חקר - ניצול בבחירת תכונה ומכליל קריטריונים היוריסטיים שהוצעו בעבר. הקריטריון החדש מושווה אנליטית ואמפירית לקריטריונים אחרים של למידה אקטיבית הנפוצים.

לאחר מכן, נחשב מחלקת ההשערות היפר-מישוריות הליניאריות עם רעש תווית א-סימטרי. הביצועים הניתנים להשגה עבור הקריטריון המוצע מנותח באמצעות אלגוריתם חדשני בעל מורכבות נמוכה המבוסס על סכמת ההתאמה האחורית לתקשורת עם משוב. הוכח כי עבור רעש תווית כללי והתפלגות תכונה מוגבלת, הקריטריון התיאורטי החדש של המידע דועך במהירות אקספוננציאלית לאפס. בהתבסס על משפט הקיבול – יתירות מהחלק הקודם אז ניתן להסיק כי גם היתירות דועכת במהירות לאפס.

בחלק השני של המחקר, אנו שוקלים את ההנחה האינדיבידואלית, שאינה מניחה קשר הסתברותי בין האימון לנתוני המבחן. מונעים על ידי קידוד מקור אוניברסלי, אנו מציעים קריטריון שבוחר לתייג נקודות נתונים הממזערות את החרטה המינימלית-מקסימלית במערך הבדיקה. הוכח כי עבור סיווג בינארי ורגרסיה ליניארית, הקריטריון המתקבל עולה בקנה אחד עם קריטריונים ידועים של למידה אקטיבית ולכן מייצג גישת למידה אקטיבית תיאורטית מאוחדת עבור שיעורי השערות כלליות. לבסוף, הוכח באמצעות נתונים אמיתיים שהקריטריון המוצע עולה על קריטריונים אחרים של למידה אקטיבית מבחינת מורכבות המדגם. לבסוף, אנו רואים את מחלקת ההשערה של רשת עצבים עמוקה (DNN). על ידי החלת גרסה משוערת של הקריטריון האישי שלנו על רשתות עצביות, אנו מראים שבנוכחות נתונים מחוץ להפצה, הקריטריון המוצע מפחית את גודל מערך ההדרכה הנדרש בעד 10.4%, 15.4% ו- 12% עבור מערכי נתונים CIFAR10, EMNIST ו-MNIST בהתאמה.

עבודה זו נעשתה בהנחיית

פרופ' מאיר פדר

TEL AVIV UNIVERSITY אוניברסיטת תל-אביב
FACULTY OF ENGINEERING הפאקולטה להנדסה

הפקולטה להנדסה ע״ש איבי ואלדר פליישמן
בית הספר לתארים מתקדמים ע״ש זנדמן-סליינר

גישות אינפורמציוניות ללמידת מכונה אקטיבית

חיבור לשם קבלת התואר ״דוקטור לפילוסופיה״

שחר שיוביץ

הוגש לסנאט של אוניברסיטת תל-אביב

עבודה זו נעשתה באוניברסיטת ת״א בפקולטה להנדסה
בהנחיית פרופ׳ מאיר פדר

סיוון ה׳תשפ״ד

TEL AVIV UNIVERSITY אוניברסיטת תל-אביב
FACULTY OF ENGINEERING הפאקולטה להנדסה

הפקולטה להנדסה ע״ש איבי ואלדר פליישמן

בית הספר לתארים מתקדמים ע״ש זנדמן-סליינר

גישות אינפורמציוניות ללמידת מכונה אקטיבית

חיבור לשם קבלת התואר ״דוקטור לפילוסופיה״

שחר שיוביץ

הוגש לסנאט של אוניברסיטת תל-אביב

סיוון ה׳תשפ״ד