Dataset Structure and Active Learning
A Comparative Regression Study

Madison Stulir, Ethan Gaskin, Nick Ackerman

May 2024

1 Specific Aims

We implemented several active learning methods for performing regression on three datasets, applying each active learning method in tandem with one of three base learners: random forests, linear combinations of independent basis functions (classic linear and nonlinear regression), and neural networks. We report the performance of each method on each dataset and, importantly, attempt to explain why each method is well-suited (or not) to each dataset. Our study sheds light on the connection between dataset structure and active learning performance.

2 Background

Active learning for regression has been an active area of research. Development of new methods of active learning for regression problems has been ongoing. For example, the P-ALICE framework was published in 2009, and uses importance-weighted least-squares to choose an optimal set of data from a pool of samples [1]. Additionally, the ALIEN framework for active learning was published in 2024 and uses neural networks and a batch selection method based on determinant maximization [2]. While work has been done to create new methods to perform active learning, and the methods show promise on the data used in their development, relatively little work has

been done to characterize and compare the performance of different active learning methods for regression on different datasets. Work has been done in this field in the area of classification, specifically for a logistic regression model. *Yang et al.* [3] used 3 synthetic datasets to determine which dataset characteristics lead to optimal performance of different active learning methods. We aim to contribute knowledge to this area of research for regression problems and base learners, and choose to use real-world datasets to do so.

3 Significance

It is rarely the case that an active-learning method outperforms all other methods on every dataset. Methods are designed with specific assumptions about the data in mind. In many cases, such assumptions are mathematically formalized and used to prove performance guarantees about the method in question. However, in practice, mathematical assumptions about dataset structure may not hold true. Therefore, it is crucial to empirically measure what other characteristics of a dataset contribute to or detract from a method's performance. Our study could help active learning practitioners select the best active learning method for their dataset.

4 Experimental Design

4.1 Data

4.1.1 Logd74

The logd74 dataset consists of 1130 compounds and a variable representing each compound's lipophilicity. It was published by *Wang et al.* [4]. Each compound is defined by a SMILES string, a textual representation of molecular structure. We featurized each SMILES string by computing its Morgan fingerprint. We chose a fingerprint length of 2048, so the featurized logd74 dataset has

1130 samples and 2048 features.

In the context of drug discovery, lipophilicity is a critical property of small molecules. Molecules that are more lipophilic have a greater ability to diffuse through cell membranes and thus a greater chance of reaching targets enclosed in membranes.

The distribution of the logd74 target variable (lipophilitcy) is shown in Figure 1. It appears to be approximately normally distributed.

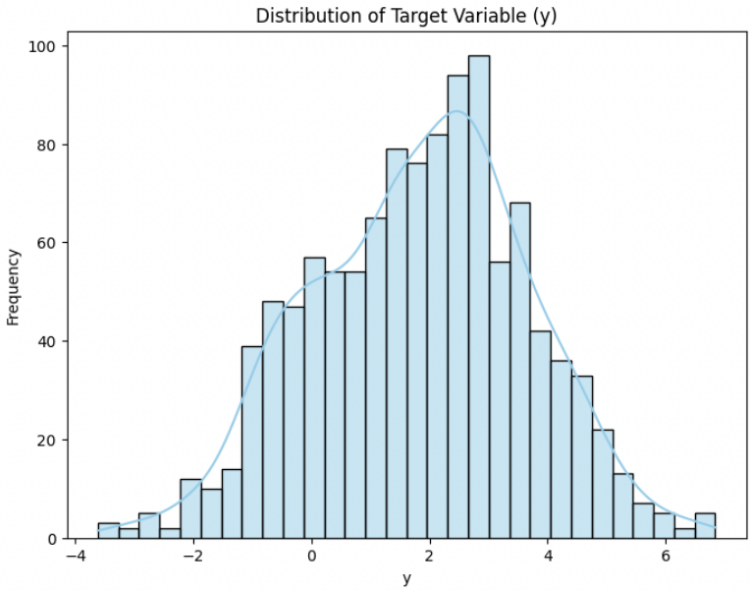

Figure 1: Distribution of the y variable for the logd74 dataset

We also performed UMAP on the logd74 dataset. The results, displayed in Figure 2, show that there is little relationship between a molecule's location on the plot and lipophilicity, suggesting this may be a challenging dataset to model.

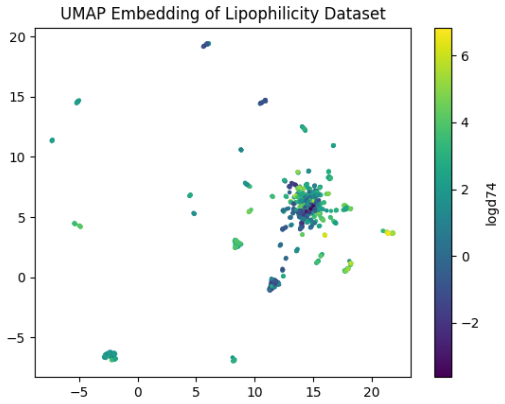

Figure 2: UMAP of the logd74 dataset

4.1.2 Abalone

The abalone age dataset is a classic UCI ML repository regression dataset. The regression target is abalone age, which is predicted from moderately accessible features of abalone such as sex, length, diameter, height and various weight measurements. The typical process for age-identification of abalone can be tedious and time-consuming, since it requires abalone experts observing cross section(s) of abalone under a microscope [5]. Therefore, predictive models performing this task would be valuable. There are 4177 samples and 8 features in this dataset.

With respect to our goal—characterizing active learning method performance on different types of real-world regression datasets—we hope that this dataset will serve as a simple benchmark given the simplicity of the relationship between target and features. This is demonstrated by the distinct structure when projected unto the first two principal components, which is displayed in Figure 3. The three different sexes form 2D-linear clusters when projecting the data unto the first two principal components. Furthermore, as seen in Figure 4, the target is approximately normally distributed and has relatively low variance. This should make the prediction task, as measured by MSE, easier

compared to regression targets that have high variance.

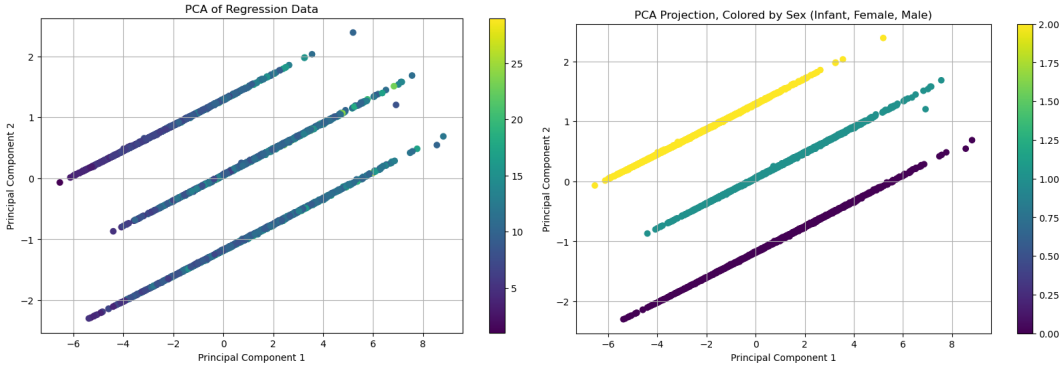

Figure 3: PCA of abalone dataset. Left is colored by target variable. Right is colored by sex-class (0=infant, 1=female, 2=male).

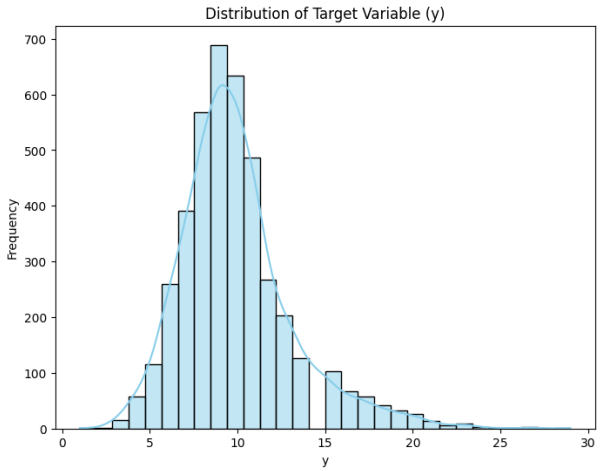

Figure 4: Distribution of dependent variable, Abalone Age dataset

4.1.3 Inhibition

The Inhibition dataset consists of 569 samples and 28 features. These samples represent kinase enzymes. The dependent variable is the inhibition effect of a particular small molecule drug on

that protein, ranging from 100 to 0: 0 indicates full inhibition of protein activity and 100 indicates no inhibition [6]. The original data consists of UniProt IDs for the kinases, as well as mutations to the original UniProt ID. We generated the amino acid sequences by retrieving the sequences from the UniProt website using an HTML request and then induced mutations to the original sequence. These sequences were then used to generate the features for the dataset. To do this, we utilized the ProtParam module of the BioPython package. Features generated were amino acid percentages, beta sheet and alpha helix frequencies, isoelectric point, molecular weight, aromaticity, and instability and GRAVY indexes. We also included features that were included in the original dataset, including if the protein was phosphorylated and the organism the protein was produced in for the experiment.

After performing exploratory data analysis, we see that the distribution of the dependent variable, shown in Figure 5, is heavily skewed.

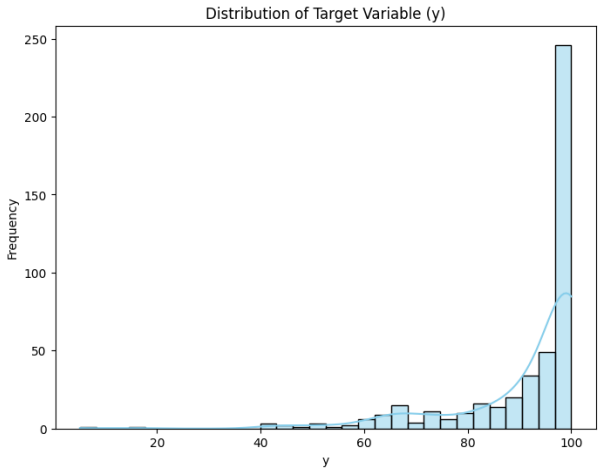

Figure 5: Distribution of the y variable for the inhibition dataset

We also performed UMAP dimensionality reduction of the dataset, as shown in Figure 6. This mapping also revealed a complex pattern, meaning the data is unlikely to be modeled well using a linear model.

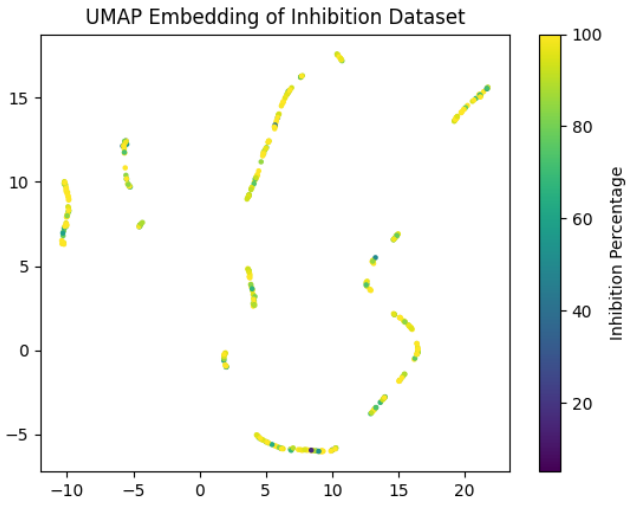

Figure 6: UMAP of the Inhibition Dataset

4.2 Methods

4.2.1 Random Forests

The first base learner used in our experiment was random forests. We then used 4 sampling techniques to perform batch-wise active learning. The random forests were implemented using scikit-learn and trained using the sklearn's default algorithm parameters and hyperparamters, without hyperparameter tuning.

Random Sampling

To perform random sampling, we selected k, where k represents the batch size, samples randomly from the total pool of data not currently used for training. We repeated this process for every iteration until 50% of the data had been sampled.

Uncertainty Sampling

To perform uncertainty sampling, we determined the uncertainty of the model by collecting

the predictions of each tree in the random forest. We then computed the variance between these predictions for each sample in the unlabeled pool. Then, we selected the k samples with the highest uncertainty to obtain labels for each iteration of the model training.

Diversity Sampling

To consider diversity in active learning sampling for the random forest, we performed clustering on the unlabeled pool of data with 2 methods. The first method was k-means clustering, where we form k clusters of the data, k being the batch size. We then compute the uncertainty of every point in the pool as we did in uncertainty sampling. We then choose the most uncertain samples from each cluster to obtain the label at each iteration.

The second clustering method we implemented was hierarchical clustering. We obtain k clusters by cutting the hierarchical tree at the point where there are k branches, and all points below the branch belong to that cluster. We then compute uncertainty as described in uncertainty sampling, and select the most uncertain sample from each cluster to obtain the label. These methods aim to balance both uncertainty and diversity to obtain a batch that does not select k uncertain points that are very similar, which would be less beneficial for model performance.

4.2.2 P-ALICE

P-ALICE stands for Pool-P-ALICE, and was developed by Sugiyama & Nakajima [1]. This method adapts the original P-ALICE [7] method from the membership query synthesis data access model to the pool-based data access model. P-ALICE is an active learning method restricted to base learners that are linear combinations of independent basis functions, as described by Equation 1, where m is the number of features for each sample and x_ℓ is the ℓth element of some sample vector $\vec{x}$. When $\varphi_\ell(x_\ell) = x_\ell$ then this is simply linear regression.

$$f(\vec{x}) = \sum_{\ell=1}^{\ell=m} \theta_\ell \varphi_\ell(x_\ell) \tag{1}$$

P-ALICE is an aggressive active learning method that corrects for resampling bias using importance weighting. These importance weights are used to sample from the pool, used in the weighted least squares estimate for the parameters to the regression, and are critical in finding $\hat{\lambda}$ that minimizes the P-ALICE criterion, $PALICE(\hat{\lambda}) = tr(\hat{U}L_{\hat{\lambda}}L_{\hat{\lambda}}^{\intercal})$ where $L_{\hat{\lambda}} = (X_{\hat{\lambda}}^{\intercal}W_{\hat{\lambda}}X_{\hat{\lambda}})^{-1}X_{\hat{\lambda}}^{\intercal}W_{\hat{\lambda}}$, $\hat{U}$ is the covariance matrix of X *after* applying the basis functions to X, and λ is a hyperparameter controlling the shape of the sample distribution (how likely a given sample will be selected from the given pool). The P-ALICE criterion is inspired by DOE (referred to as Q-optimality) and the subset of samples from the pool that minimize this quantity (correponding to $\hat{\lambda}$) is sought by the algorithm. This criterion does not have analytical solution, so multiple values of λ must be attempted.

The nonlinear base learners for Abalone Age and Kinase Inhibition were determined through exploratory data analysis, where original features, squared features, cubed features, and first order interactions were tested for significant correlation against the target under the Spearman's rank correlation coefficient test. Terms that had a p-value greater than 0.05 were included in the nonlinear model. The nonlinear base learner for Abalone used Equation 2 which includes all original features, squared features, cubed features, and first-order interactions of features since they were all significantly correlated with the target; this required a total of 45 parameters. The nonlinear base learner for logd74 used Equation 3, which we hoped would not be too computationally expensive but offer unique results compared to linear; this required 2049 parameters (the last is for a constant term). The non-dimensionality-reduced Kinase Inhibition nonlinear base learner used a subset of original features, squared features, cubed features, and first-order interactions based on the process described above; this required 87 parameters. The dimensionality-reduced Kinase Inhibition nonlinear base learner was arbitrarily set to $\varphi_{\ell}(x_{\ell}) = sin(x_{\ell})$ for all ℓ to show that even with a poorly specified model, the dimensionality-reduced version of the data allows P-ALICE to perform much better compared to the original data; this required 16 parameters, one for a constant term (y-intercept) and one for each of the 15 components of the projected data, see Section 5.3.

$$f(\vec{x}) = 1+\theta_1 x_1+\theta_2 x_2+\cdots+\theta_8 x_8+\theta_9 x_1^2+\theta_{10} x_2^2+\cdots+\theta_{17} x_1^3+\theta_{18} x_2^3+\cdots+\theta_{24} x_1 x_2+\cdots \quad (2)$$

$$f(\vec{x}) = \sum_{\ell=1}^{\ell=m} \theta_\ell \varphi_\ell(x_\ell), \quad \varphi_\ell(x_\ell) = \begin{cases} 1 & \text{if } x_\ell = 1 \\ -1 & \text{if } x_\ell = 0 \end{cases} \quad (3)$$

4.2.3 Neural Networks

The third base learner we used was neural networks. All neural networks we trained were multi-layer perceptrons (MLPs) with Leaky Rectified Linear Unit activation functions applied between each layer. Dropout was applied to each layer with probability 0.2. Because the number of features in each dataset varied, we used different MLP widths and depths for each dataset. In particular, we used, in order, layers of sizes 8, 6, 4, 2, and 1 for the Abalone dataset, layers of sizes 29, 15, 7, 3, and 1 for the inhibition dataset, and layers of sizes 2048, 512, 256, 16, and 1 for the logd74 dataset.

To perform uncertainty estimation, we used Monte Carlo dropout with 25 dropout-enabled forward passes per sample.

Random Sampling

To perform random sampling, we simply randomly selected a batch of samples to label at each iteration.

Uncertainty Sampling

To perform uncertainty sampling, we selected for labeling the k (batch size) samples with highest uncertainty as measured by Monte Carlo dropout.

COVDROP Sampling

The COVDROP method is outlined in *Bailey et al* [2]. The premise of the method is that the k samples with greatest uncertainty may be highly correlated with each other (and thus not provide

much useful information to the learner). Therefore, it may be valuable to consider correlations between samples when selecting a batch. Under certain assumptions, it can be shown that maximizing joint entropy is equivalent to maximizing the determinant of the covariance matrix. Unfortunately, identifying the batch that maximizes this determinant is NP-hard, so COVDROP uses an iterative optimization method that may not find the single, optimal batch.

Thus, COVDROP estimates the covariance matrix and searches for a batch whose covariance matrix has determinant as large as possible. COVDROP sampling is outlined in detail in *Bailey et al.*, so we will not repeat it here.

5 Results

5.1 Setup

Before we discuss results, we will outline the sampling procedure setup. We performed 5 simulations for each active learning method on each dataset. First, before doing any sampling, we randomly split the data into train (80%) and test (20%) sets. This split was performed for each trial of a simulation. All figures below display model performance on the held-out test dataset, as the training pool from which we choose samples to label and the labeled training data are biased, and metrics calculated on these datasets are not strong indicators of model generalizability.

To initialize the base learners, we started by requesting labels for a random 20% of the training data and stopped sampling once we had labels for 50% of the training data.

For the sake of consistency, we used the mean squared error (MSE) loss function throughout our study.

Note, P-ALICE, as a DOE inspired method, does not require an initially labeled set, nor does it iteratively add samples to the labeled set. To make it comparable to the other active learning methods, we allowed P-ALICE to choose its predicted optimal set of samples for all set sizes

between [20%, 50%] of the training dataset, incrementing by batch size.

5.2 Results: Abalone Age

Regression results for the Abalone Age dataset are displayed in Figure 7. For the random forest base learner, we see mostly consistent MSE across rounds of active learning. There is also consistent MSE between the sampling methods used for random forest. This indicates that the random forest base learner may have trained an optimal model with the initial 20% of data, and that the model does not need additional data sampled from active learning to obtain an optimal model, and therefore the method of sampling does not impact active learning performance. In comparing the results for P-ALICE, for the nonlinear learner, we see the MSE does decrease in the initial rounds of sampling, then stabilizes to a similar performance to the random forests. We see lower MSE similar to random forest from the beginning with the linear learner. The MLP has much higher error than the other models, and results do not change much as active learning progresses, meaning the model may have also achieved an optimal performance for this model architecture with the initial training data. The model performance is very similar across active learning sampling strategies for the MLP.

We can see that the main difference in MSE is between the base learners rather than active learning methods implemented. A random forest base learner achieves the best performance, followed by P-ALICE's base learners then the MLP.

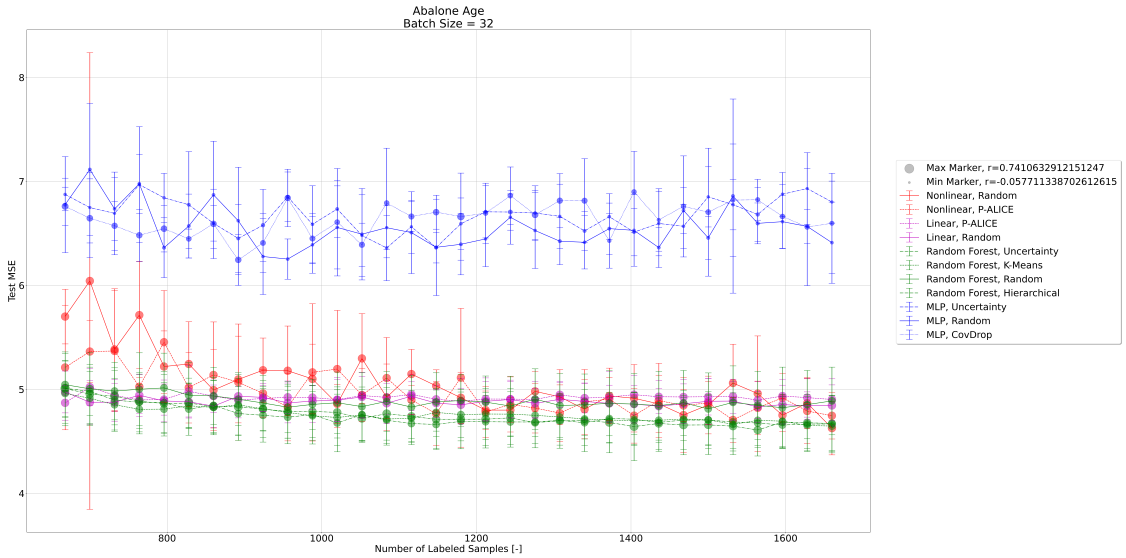

Figure 7: Average MSE on the Abalone dataset across iterations. The height of each point represents the average MSE on the held-out test set across 5 simulations. Error bars represent the standard deviation of test MSEs in that iteration across 5 simulations. The size of each point represents the average batch-wise correlation in that iteration.

5.3 Results: Kinase Inhibition

Regression results for the Kinase Inhibition dataset are displayed in Figures 8 and 9. In the results for Kinase Inhibition, we see the main differences in model performance are dependent on the base learner used, rather than the active learning method. For the random forest and P-ALICE's base learners, there is a large difference in MSE between active learning selection methods, and that MSE does not tend to decrease as more data is added to the model. This can be explained as follows: the model may be optimally trained with the initial training data, and thus additional data points may not improve performance. For the MLP base learner, we see that the test MSE does decrease as more data is used to train the model. The COVDROP method has the highest MSE across batches,

followed by random sampling and uncertainty sampling. This indicates that COVDROP may not be ideal for training a model on this dataset, although the MSE is similar between the 3 MLP sampling methods.

We also show in Figure 9 that P-ALICE's base learners originally struggled to fit a model to this dataset, resulting in a test MSE of 10^{11}. To reduce the error of this method on this dataset, we performed PCA and trained the P-ALICE model by projecting onto the first 15 principal components (the number of components needed to explain at least $\geq$90% of variance), resulting in the test MSEs shown in Figure 8. Sugiyama [1] notes that when the feature space is very high dimensional, the variance of the target tends to dominate the bias of the model, so the advantage of using P-ALICE is diminished. To remedy this, he suggests performing dimensionality reduction on the data. Based on this suggestion, we hypothesized that dimensionality reduction of the Kinase Inhibition dataset might improve P-ALICE performance. Indeed, Figure 8 shows significant improvement compared to Figure 9. The number of features is 28 for this dataset, which is significantly smaller than the 2048 features for the logd74, yet the logd74 dataset did not require dimensionality reduction for P-ALICE and its base learners. Our hypothesis for why this is the case is that the Kinase Inhibition's (non-dimensionality-reduced) distribution of target versus features had a fairly noisy relationship to the target, and that using PCA to summarize the data's variance in lower dimensions indirectly removed some of the noise/variance in the distribution of target versus features, and thus improved performance of P-ALICE.

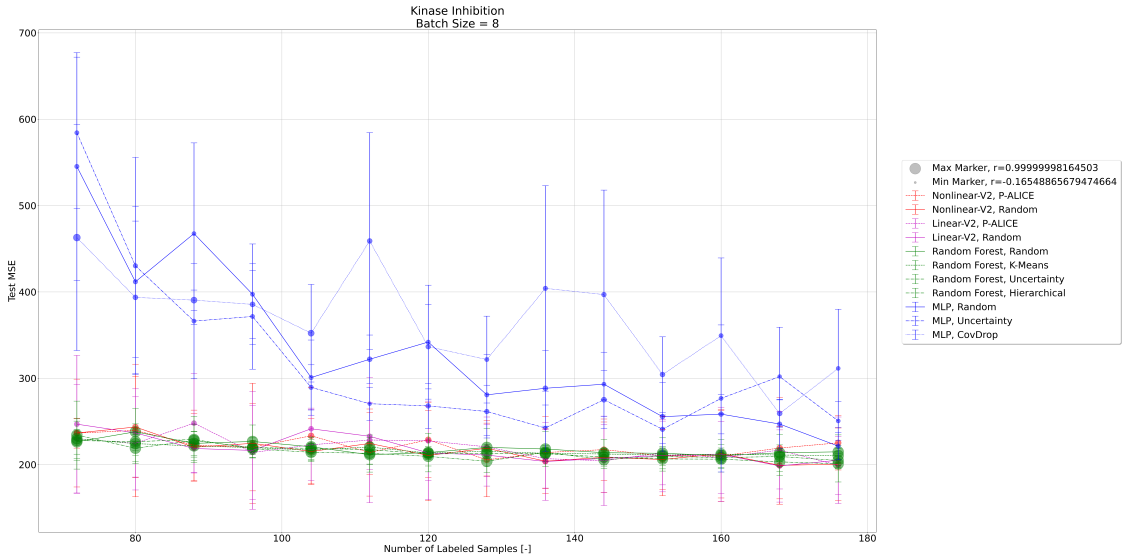

Figure 8: Average MSE on the Inhibition dataset across iterations. The height of each point represents the average MSE on the held-out test set across 5 simulations. Error bars represent the standard deviation of test MSEs in that iteration across 5 simulations. The size of each point represents the average batch-wise correlation in that iteration.

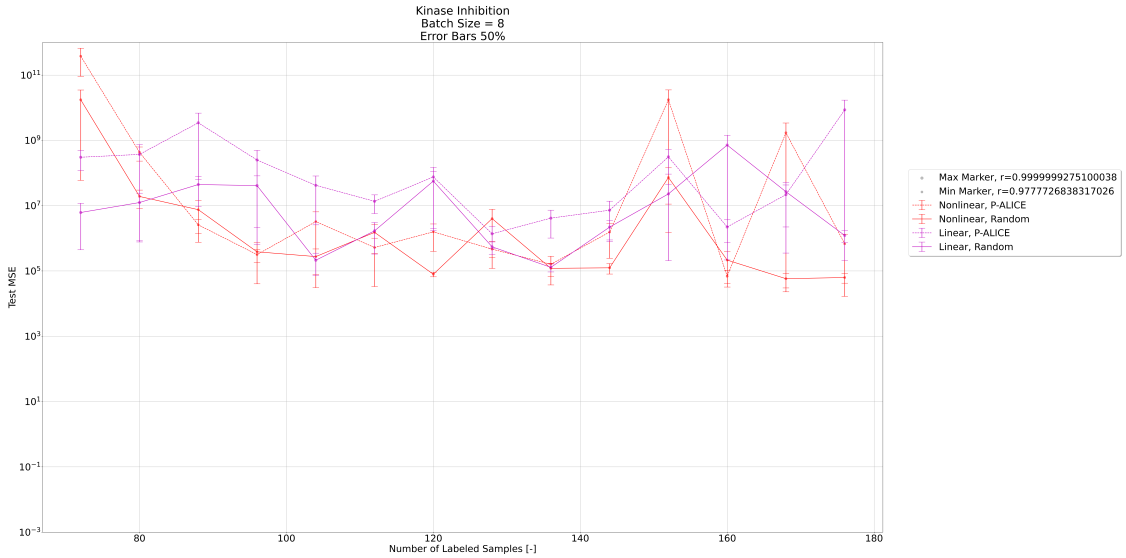

Figure 9: Average MSE on Inhibition dataset across iterations. This plot shows the especially poor performance of P-ALICE on the non-dimensionality reduced Inhibition data. Error bars represent the standard deviation of test MSEs in that iteration across 5 simulations. The size of each point represents the average batch-wise correlation in that iteration.

5.4 Results: Lipophilicity of Small Molecules

Regression results for the logd74 dataset are displayed in Figure 10. As was true in other cases, performance differences between active learning methods were dominated by performance differences between base learners. The three neural network sampling methods begin as the worst-performing methods but improve rapidly when given more data. Of the three, COVDROP performs best by the end of sampling; however, the neural network random sampling error bar encompasses COVDROP performance.

All three random forest sampling methods have comparable performance. In the case of P-ALICE, performance degrades unexpectedly in the final iterations (red and purple dotted lines $>$400

labeled samples). We hypothesize that these are due to unfavorable train/test splits under the seeds used for some of the simulations, as indicated by the disproportionately large error bars on the two points where the MSE suddenly increases. Given more compute resources, we would hope that additional simulations would demonstrate that P-ALICE tends to continuously improve as given more data. Our hypothesis is further supported by the fact that the same set of seeds was used across all base learners, and the error bars across those four base learner/active learning method combos all were all relatively large.

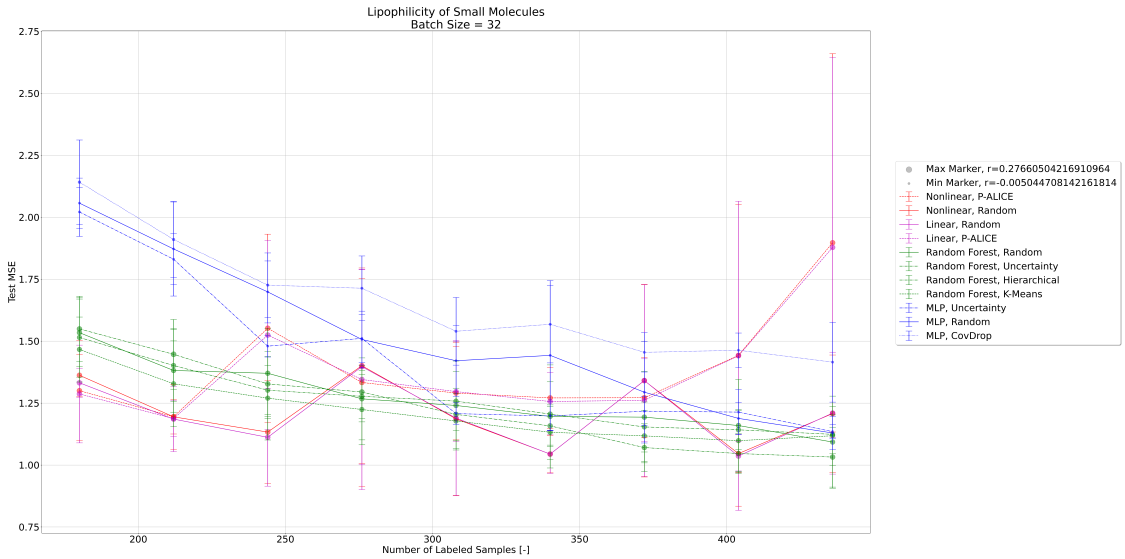

Figure 10: Average MSE on the logd74 dataset across iterations. The height of each point represents the average MSE on the held-out test set across 5 simulations. Error bars represent the standard deviation of test MSEs in that iteration across 5 simulations. The size of each point represents the average batch-wise correlation in that iteration.

6 Discussion

Our study compared the performance of several active learning methods for regression on three datasets. However, drawing definite conclusions about which aspects of dataset structure improve or degrade active learning performance proved difficult, for two main reasons.

First, as we should have anticipated, base learners were a confounding variable. For example, comparing P-ALICE sampling to COVDROP sampling amounts to comparing two different base learners *and* two different active learning methods. This makes it hard to draw conclusions about the active learning methods alone.

Second, real datasets differ in lots of ways. This makes it challenging to hone in on the most salient aspects of dataset structure. For example, when comparing the logd74 dataset to the Inhibition dataset, one may note that the logd74 target variable is approximately normally distributed, whereas the Inhibition target variable is heavily left-skewed. However, it is also the case that the featurized logd74 dataset has many more features than the Inhibition dataset (2048 vs. 28). Which difference affects active learning performance more? It is hard to tell.

As the results plots show, we also measured the average per-batch pearson correlation for all methods. Unexpectedly, we did not see a significant trend. Batch-wise correlations appear to be relatively randomly distributed. One possible reason this could be the case is that batch-wise correlations might not be a perfect proxy for information content on the datasets we chose.

If we were to redesign our study, we would control for the variables mentioned above by holding the base learner constant as we compared active learning methods and by testing these methods on synthetic data.

In spite of these challenges, conducting this study gave us an opportunity to work with real biological datasets and develop a deeper understanding of the active learning methods we implemented.

References

[1] M. Sugiyama and S. Nakajima. *Pool-based active learning in approximate linear regression*. Mach Learn 75, 249–274. 2009. URL: {DOI}:https://doi.org/10.1007/s10994-009-5100-3.

[2] Michael Bailey et al. *Deep Batch Active Learning for Drug Discovery*. eLife. 2023. URL: https://elifesciences.org/reviewed-preprints/89679.

[3] Yazhou Yang and Marco Loog. *A Benchmark and Comparison of Active Learning for Logistic Regression*. 2016. arXiv: 1611.08618 [cs.CV]. URL: https://arxiv.org/abs/1611.08618.

[4] Jian-Bing Wang et al. *In silico evaluation of logd74 and comparison with other prediction methods*. Journal of Chemometrics. 2015. URL: https://nanx.me/papers/logd.pdf.

[5] Warwick Nash et al. *Abalone*. UCI Machine Learning Repository. 1995. URL: {DOI}:https://doi.org/10.24432/C55C7W.

[6] Nathanael Gray and Jinhua Wang. *QL-XII-47 KINOMEscan (LDG-1397: LDS-1505)*. LINCS Data Portal. 2017. URL: http://identifiers.org/lincs.data/LDG-1397.

[7] Masashi Sugiyama. *Active Learning in Approximately Linear Regression Based on Conditional Expectation of Generalization Error*. Journal of Machine Learning Research. 2006. URL: https://jmlr.org/papers/v7/sugiyama06a.html.