STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **x**, Month 202x, pp 0–27.
Published online in International Academic Press (www.IAPress.org)

Selection of initial points using Latin Hypercube Sampling for Active Learning

Nompilo Mabaso [1,2,*], Roelof L.J. Coetzer [1], Shawn C. Liebenberg [1,2]

[1] *Focus Area for Pure and Applied Analytics, North-West University, South Africa*
[2] *School of Mathematical and Statistical Sciences, North-West University, South Africa*

Abstract Classification requires labelling large sets of data, which is often a time-consuming and expensive process. Active learning is a machine learning technique that has gained popularity in recent years due to its ability to effectively reduce the amount of labelled data required to train accurate models. The success of the active learner heavily relies on the selection of the initial points to initialise the active learning process. In this paper, we compare the performance of the traditional random sampling approach to the maximin Latin Hypercube sampling, conditioned Latin Hypercube sampling, and a modified Latin Hypercube sampling procedure for initialising active learning for the estimation of the logistic regression in binary classification problems. We show that the Latin Hypercube sampling designs outperform random sampling for all the performance measures evaluated. The results are demonstrated using simulated data sets and an actual case study. Specifically, the conditioned Latin hypercube sampling design exhibits high prediction accuracy using a smaller sample size for both heterogeneous and homogeneous classes. In contrast, the modified Latin hypercube sampling design yields the smallest variance of prediction across varying initial sample sizes for both homogeneous and heterogeneous classes. Furthermore, principal component analysis indicates that approximately 10% of the data is required to develop an accurate and precise logistic regression classifier.

Keywords Active Learning, Uncertainty Sampling, Latin Hypercube Designs, Logistic Regression

DOI: 10.19139/soic-2310-5070-2688

1. Introduction

Binary classification is a common task in machine learning where the goal is to assign binary labels to samples based on some explanatory variables. Traditionally, samples with class labels are provided for training the classifier. However, due to 4IR and the IoT, huge volumes of unlabeled data are becoming available in many industries. Consequently, the labelling of samples in big data is time-consuming and too expensive. Therefore, the labelling effort and the cost of model training must be minimised. Active learning learns from a few data points while selecting the most informative unlabeled samples for labelling to improve model performance. Active learning has gained much popularity in recent years due to its ability to effectively reduce the amount of labelled data required to train accurate models. [12] provided an excellent introduction and detailed overview of active learning and different query strategies. [45] provided a recent review of active learning query strategies for classification, regression and clustering. Over the last two decades, active learning has been successfully applied to various machine learning tasks, including binary classification, and has been shown to reduce the amount of labelled data required while achieving high model performance [27, 26, 46, 43, 77]. The most commonly used active learning criteria include query-by-committee [12], uncertainty sampling, error reduction, variance reduction, minimum

*Correspondence to: Nompilo Mabaso (Email: nompilo.mabaso@nwu.ac.za). School of Mathematical and Statistical Sciences, North-West University, 11 Hoffman street, Potchefstroom, North West Province, South Africa.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

loss increase, and maximum model change [26]. The simplest and most popular criterion is uncertainty sampling introduced by [21], where an active learner selects observations that are least certain of classifying. Sampling and labelling these least certain observations can help the model refine the decision boundary. This approach is often straightforward for probabilistic learning models. For example, when using a probabilistic model for binary classification, uncertainty sampling criteria simply select the observations whose probability of success is nearest 0.5 [21, 12]. [26] provided a benchmark comparison of active learning criteria specifically for logistic regression. They found that uncertainty sampling is a robust active learning algorithm, regardless of the complexity of the task.

Many active learning approaches have also been developed on experimental design techniques [10]. [27] provided an evaluation of various active learning heuristics and different loss functions motivated by experimental design for logistic regression. They found that the experimental design approaches never performed worse than random sampling over a wide range of different data sets. [9] presented a sequential sampling algorithm for estimating the class probability with minimum variance and bias using a penalized logistic regression model. They showed that selecting samples to minimise the mean squared error of the estimated posterior probability outperforms random selection and a variance-based active learning criterion [27]. [8] proposed a manifold active learning design criterion for minimising the variance of the parameter estimates of the active learner, while maximising the dependence of the sampled data points and their predicted values. They commented that the support vector machine approach depends on the accuracy of the initial classifier.

The success of any active learner heavily relies on the selection of the initial points to initiate the active learning process. The majority of the active learning literature utilises random sampling on the first iteration, where after an active learning criterion is employed for selecting the next and subsequent most informative samples until convergence. [43] proposed an iterative algorithm based on random sampling of labelled data points until at least one data point per category is available to initialise the active learning process. However, the targeted selection of initial points to accelerate and improve the performance of active learning criteria has not received much attention in the literature.

In this paper, we introduce Latin Hypercube sampling to initialise the active learning process for the estimation of the logistic regression classifier. The advantage of the Latin Hypercube sampling is that it selects initial points that are scattered throughout the design space of the input variables, ensuring the selection of design points in each category, which allows for a more accurate estimation of the initial classifier. We also employ conditional Latin Hypercube sampling [28], and introduce a modified Latin Hypercube sampling approach for selecting the initial points. We show that Latin Hypercube sampling designs outperform random sampling for minimising the mean squared error of the logistic regression classifier using simulated data. We also compare the performance of the initial sampling approaches using other performance measures such as precision, F1-score, generalised variance of the parameter estimates, and the mean squared error of the predicted posterior probability [9], over a wide range of the number of initial points specified. The use of Latin Hypercube sampling in initialising active learning was not communicated previously in the literature. We consider logistic regression since it is the most widely applied classifier in machine learning and in the applied sciences [26]. However, the proposed sampling strategies can also be adopted for other classifiers.

The paper is outlined as follows: Section 2 outlines the problem formulation and provides the theoretical background on logistic regression, active learning, and performance metrics. Section 3 discusses Latin hypercube designs, while Section 4 details the simulation designs and presents a discussion of the results. Section 5 explores the application of the proposed approach to a real-world example. Finally, Section 6 summarises the key contributions of the paper and offers an overview of potential future research directions.

2. Preliminaries

2.1. Problem setting and model estimation

Adopting the notation of [9], suppose there is a pool of data available denoted as $\mathcal{D} = \{\mathcal{L} \cup \mathcal{U}\}$, where $\mathcal{L} = \{\mathbf{x}_1, \cdots, \mathbf{x}_{n_0}\}$ is an initially selected sample of small size n_0 with associated labelled responses $\mathbf{Y} = \{Y_1, \cdots, Y_{n_0}\}$. The set, $\mathcal{U} = \{\mathbf{x}_{n_0+1}, \cdots, \mathbf{x}_N\}$ contains large amounts of unlabelled data. Given the initial sample $\mathcal{L}$, the objective of active learning is to find a subset $\mathcal{U}^* = \{\mathbf{x}_1^*, \cdots, \mathbf{x}_{n_1}^*\}$ that contains the most informative samples for labelling. Therefore, the final selected sample of size $n = n_0 + n_1 \leq N$ will reduce the generalisation error of the active learner the most if they are labelled and included in $\mathcal{L}$ for training. In this paper, the problem considered is two-fold; first to specify the initial sampling design that performs best in initialising the active learner as part of the training set $\mathcal{L}$, and second to recommend the final sample size n that are "optimal" according to various performance measures for classification.

Now, consider a Bernoulli distributed random variable Y, which can only take one of two possible values. Let $Y_i, i = 1, \cdots, n_0$ be the labelled value on the i-th observation. Denote the labelled value as either 1 or 0 for success and failure, respectively. The mass function can then be written as

$$\mathbb{P}(Y_i = y) = \pi_i^y (1 - \pi_i)^{1-y}, \ y = 0, 1.$$

Now suppose there are p explanatory variables which can potentially influence the outcome of the random variable, and furthermore, the interest is in specifying a model to quantify the relationship between the $\mathbb{E}(Y_i) = \mathbb{P}(Y_i = 1)$ and the explanatory variables. To this end, set the conditional probability as $\mathbb{P}(Y_i = 1|\mathbf{x}_i) = \pi_i$ and model with logistic regression,

$$\pi_i = \frac{e^{\eta_i}}{1 + e^{\eta_i}},$$

where $\eta_i = \mathbf{x}_i^\top \boldsymbol{\beta} = \beta_0 + \sum_{j=1}^p x_{ij}\beta_j$, and $\boldsymbol{\beta} : (p+1) \times 1$ are unknown parameters to be estimated from the training set $\mathcal{L}$. The logit transformation,

$$\ln\left(\frac{\pi_i}{1 - \pi_i}\right) = \eta_i, \tag{1}$$

is known as the link function or the linear predictor [7]. The predicted posterior probability is specified as $\widehat{\pi}_i = e^{\widehat{\eta}_i}/(1 + e^{\widehat{\eta}_i}) = e^{\mathbf{x}_i^\top \widehat{\boldsymbol{\beta}}}/(1 + e^{\mathbf{x}_i^\top \widehat{\boldsymbol{\beta}}})$ and the parameter estimates $\widehat{\boldsymbol{\beta}}$ are estimated by maximum likelihood [16]. From general likelihood theory, the Fisher information matrix of the parameters is

$$\mathbf{M}(\mathbf{x}, \mathbf{fi}) = -\mathbb{E}\left(\frac{\partial^2 l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^\top}\right) = \mathbf{X}^\top \mathbf{W} \mathbf{X},$$

where $l(\boldsymbol{\beta})$ is the log-likelihood function given by

$$l(\boldsymbol{\beta}) = \sum_{i=1}^{n_0} Y_i \eta_i - \sum_{i=1}^{n_0} \ln\left(1 + e^{\eta_i}\right), \tag{2}$$

and $\mathbf{W}$ is the diagonal weight matrix with $W_{ii} = \pi_i(1 - \pi_i)$. The matrix $\mathbf{X} : n_0 \times (p+1)$ is the design matrix expanded for all the terms in the linear model. Maximising (2) can be obtained by employing the Newton-Raphson method. Now, the variance-covariance matrix of the parameter estimates can be approximated by

$$\mathbb{V}(\widehat{\boldsymbol{\beta}}) = \left(\mathbf{X}^\top \widehat{\mathbf{W}} \mathbf{X}\right)^{-1},$$

where $\widehat{W}_{ii} = \widehat{\pi}_i(1 - \widehat{\pi}_i)$. Note, selecting sample points that maximise $\ln|\mathbf{M}(\mathbf{x}, \mathbf{fi})|$, minimise the generalized variance of the parameter estimates, and is referred to as the D-optimal design [10].

A further consideration is the effect of the initial sample n_0 on the variance of the predicted posterior probability, $\mathbb{V}(\widehat{\pi}_i)$ (see Appendix A for derivation) in active learning. [27] considered the selection of sample

points for minimising the average variance of prediction over the design space. Specifically, $\sum_{i=1}^{n_0} \mathbb{V}(\widehat{\pi}_i) = \sum_{i=1}^{n_0} \mathbf{g}_i(\mathbf{x})^T \mathbf{M}(\mathbf{x}, \mathbf{fi})^{-1} \mathbf{g}_i(\mathbf{x})$, where $\mathbf{g}_i(\mathbf{x})^T = \pi_i(1-\pi_i)\mathbf{x}_i^T$ and $\mathbf{x}_i$ denotes the vector of values of the explanatory variables where the prediction is made, which they refer to as A-optimality. In fact, in optimal design literature, the criterion derived by [27] is known as V-optimality. They found that the experimental design methods more often beat random selection than the heuristic methods. [27] used random selection to initialise the active learning process. Similarly, [9] employed random sampling to initialise the active learner and compared this method to selecting the most informative subjects as those with the smallest mean squared estimation error of the predicted posterior probability for penalized logistic regression. They showed that minimising the mean squared error yields greater accuracy compared to the random selection of points in active learning. Specifically, numerical results on a wide range of real-world data sets demonstrate that the proposed method achieves effective and highly stable performance with modest computational complexity compared to random sampling and several state-of-the-art alternatives.

In establishing further notation, let the variance of the linear predictor be

$$\mathbb{V}(\widehat{\eta}_i) = \mathbf{x}_i^\top \left(\mathbf{X}^\top \widehat{\mathbf{W}} \mathbf{X} \right)^{-1} \mathbf{x}_i.$$

Selecting the sample points that minimise the integrated mean squared error over the design space, $I(\widehat{\eta}_i) = \mathrm{tr}(\mathbf{M}(\mathbf{x}, \mathbf{fi})^{-1} A)$, where $A = \int_{\mathcal{X}} \mathbf{x}\mathbf{x}^\top d\mathbf{x}$, and $\mathcal{X}$ denotes the whole design space, yields an I-optimal design. However, often the average of the variance is obtained over a large grid across the feature space, which is then referred to as a V-optimal design [10].

Consider the linear predictor (1), then along the j-th axis, the decision boundary or classifier is specified as

$$x_j = \sum_{k \neq j}^{p} m_k x_k + c, \tag{3}$$

where $m_k = -\beta_k/\beta_j$ and $c = -\beta_0/\beta_j$. Equation (3) is derived from the fact that for any point on the decision boundary, the predicted posterior probability is $\widehat{\pi}_i = 0.5$ and hence $\widehat{\eta}_i = 0$. This paper takes into consideration the effect of the initial design and the active learning criterion on both the mean squared error of the estimated slope of the decision boundary,

$$\mathrm{MSE}(\widehat{m}_k) = \mathbb{V}(-\widehat{\beta}_k/\widehat{\beta}_j) + \mathrm{Bias}(-\widehat{\beta}_k/\widehat{\beta}_j)^2, \tag{4}$$

and the mean squared error of the estimated intercept of the decision boundary,

$$\mathrm{MSE}(\widehat{c}) = \mathbb{V}(-\widehat{\beta}_0/\widehat{\beta}_j) + \mathrm{Bias}(-\widehat{\beta}_0/\widehat{\beta}_j)^2. \tag{5}$$

As a special case, for $p = 2$, the derivation of the variance and bias of $\widehat{m}_k$ i.e., $\mathbb{V}(-\widehat{\beta}_1/\widehat{\beta}_2)$, $\mathrm{Bias}(-\widehat{\beta}_1/\widehat{\beta}_2)$, is given in Appendix B. Note that these expressions can easily be expanded for $p > 2$. It will be shown that the choice of the initial design affects the variance and bias of the slope and intercept of the decision boundary, and therefore the precision and accuracy of the classifier. To our knowledge, the mean-squared error of the decision boundary has not been addressed previously in the active learning literature.

2.2. Active Learning Criterion

Given an initial training set, the aim of active learning is to iteratively select the most informative samples for labelling according to some criterion to improve the performance of the classifier. The addition of samples to the training set is stopped whenever some performance threshold is reached. In this paper, we will use uncertainty sampling for the active learning criterion. However, different measures of uncertainty will lead to different variants of uncertainty sampling. These measures include Entropy [50], Least Confident [12, 20], and Margin Sampling [51]. In the context of binary classification, entropy-based sampling is equivalent to the margin and least confident strategies, as all three approaches essentially select an unlabelled observation whose class posterior is nearest to 0.5[48, 12, 49]. The margin-based sampling approach, which selects data points closest to the decision boundary,

is the most often employed when utilising classification models [55, 54]. Therefore, in this study, we employ a margin-based approach, which is based on the distance measure of uncertainty.

Intuitively, the distance between an unlabelled observation and the decision boundary is a measure of uncertainty. As an illustration, consider a binary classifier in p-dimensional space. The classifier is given by $\sum_{j=1}^{p} \beta_j x_{ij} + \beta_0 = 0$.

Consequently, the shortest distance d_i, of a point $\mathbf{x}_i$, to the nearest point on the decision boundary is given by

$$d_i = \frac{|\sum_{j=1}^{p} \beta_j x_{ij} + \beta_0|}{\sqrt{\sum_{j=1}^{p} \beta_j^2}}.$$

Therefore, given the initial sample n_0, the observation with the minimum distance to the decision boundary is selected next for labelling. A generalized active learning procedure is provided in Algorithm 1 [66].

Algorithm 1: Generalized Active Learning Loop

Input : $\mathcal{L}$: Partial training set,
$\mathcal{U}$: Pool of unlabeled examples
T: Number of examples to sample on each iteration,
n_1: Desired training set size.
Output: Updated training set.

Active Learning Loop
while *training set size is less than the desired size* **do**
    Randomly select T observations from the pool;
    Rank these examples according to an active learning rule;
    Select the top-ranked example and present it to the annotator for labelling ;
    Add the labelled example to the training set;
return *updated training set*;

Note that $T = 1$ observation is traditionally selected from the pool in the active learning context.

2.3. Performance measures for classification

Classification performance metrics play a crucial role in evaluating the effectiveness of learning methods and models. A wide range of metrics has been developed for this purpose, each categorised based on its approach of evaluating a classifier. While some metrics prioritise minimising the number of misclassifications, others adopt a probabilistic perspective, focusing on deviations from true probabilities and evaluating the reliability of classifiers. The behaviour of these metrics can vary significantly, particularly in challenging scenarios such as imbalanced datasets or multiclass classification problems. Consider the confusion matrix in Table 1.

Table 1. Confusion table for binary classification.

	Actual Positive	Actual Negative
Predicted positive	True positives (TP)	False positives (FP)
Predicted Negative	False negatives (FN)	True negatives (TN)

Accuracy is a metric for classification models that measures the number of predictions that are correct as a percentage of the total number of predictions. Accuracy is specified as,

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}.$$

[33] argued that accuracy is not a good metric to use when there is a class imbalance. Therefore, the F1-score is preferred in situations where class imbalance is present. The F1-Score combines the precision (PRC) and recall (RC) metrics into a single metric,

$$F1 = 2\left(\frac{PRC \times RC}{PRC + RC}\right),$$

where

$$PRC = \frac{TP}{TP + FP} = P(Y = 1|\widehat{Y} = 1),$$

and the recall, also referred to as the "True Positive Rate (TPR)" or "sensitivity", is defined as

$$RC = \frac{TP}{TP + FN} = P(\widehat{Y} = 1|Y = 1).$$

The Receiver Operating Characteristic (ROC) curve plots the true positive rate (TPR) against the false positive rate (FPR) at different classification thresholds. Lowering the classification threshold results in more items being classified as positive, thus increasing both False Positives and False Negatives. The area under the ROC curve (AUC), which measures the entire two-dimensional area underneath the ROC curve, is used to summarize the performance of the classifier [58]. Therefore, the AUC provides an aggregate measure of performance across all possible classification thresholds.

There is no universally optimal measure, as the most appropriate metric is determined by the specific application and the characteristics of the data. For a more in-depth discussion on performance measures, see [67], [68], and [69].

We will evaluate the effect of the initial sample selection and the active learning criterion on the aforementioned classification performance measures.

3. Sampling Designs

At the highest level, random sampling involves selecting a subset of a population where units are chosen randomly using a random number generator. Probability sampling, a type of random sampling, meets two essential criteria. First, every unit in the population has a non-zero probability of being selected, ensuring no part of the population is excluded. Second, the probability of selecting each possible sample is known. In the design-based approach, units are selected through probability sampling, and estimates are based on the selection probabilities determined by the sampling design. Numerous design-based methods have been developed to facilitate population sampling, including but not limited to, regular grid sampling, factorial-based designs [70, 71], optimal or model-based designs [10], orthogonal design [60], and space-filling designs, such as Sobol sequences [72], Latin Hypercube designs [6] and Uniform designs [2].

Sampling methods in active learning pose challenges, primarily due to the strong benchmark set by random sampling from the available data pool. This baseline set by random sampling requires that sampling methods in active learning demonstrate clear advantages over random sampling in terms of efficiency and performance to be considered justifiable. For active learning to be practical in industrial settings, it must consistently yield performance improvements that outweigh the additional costs associated with implementing a non-random sampling approach and regularly retraining the model. This challenge is further exacerbated by the fact that active learning is particularly valuable in emerging domains where labelled data is scarce.

In this paper, we evaluate the effect of probabilistic sampling as an initial sampling design on the performance of the classifier within an active learning setting. Specifically, we employ Latin Hypercube sampling and derivatives thereof to initialise the active learning process and compare the results to those obtained with random sampling. Although many studies were done previously to compare various active learning criteria to random sampling, we are not aware of any study that utilises Latin Hypercube sampling or any other space-filling design to initialise the active learning process.

3.1. Latin Hypercube Sampling

Latin Hypercube sampling (LHS) was first introduced by [6] for selecting experimental conditions to run on computer code. The aim of LHS is to sample variable settings from their multivariate distributions while ensuring that the settings are spread out uniformly along the range of each variable axis. In essence, a sample of size n is drawn from multiple variables such that for each variable, the sample is maximally stratified [28]. A sample is maximally stratified when the number of intervals equals the sample size n for each variable, and when the probability of obtaining a sample in each of the intervals is $1/n$ [28, 6].

In the general case, consider a unit hypercube in a p-dimensional space, $[0,1]^p$, which is partitioned into n intervals of equal length, $1/n$, along each axis. This partitioning results in n intervals of equal probability corresponding to $[0,1/n), [1/n,2/n), \cdots, [(n-1)/n,1]$ for each dimension. The Latin Hypercube Sampling (LHS) can be represented as an $n \times p$ sample matrix with entries $x_{ij} \in [0,1]$. Each x_{ij} in the j-th column is restricted to one of the intervals. Research efforts across various fields have sought to enhance the performance of the original LHS. Among these strategies are Orthogonal LHS [5, 4] and Optimal LHS [1]. For a comprehensive overview, readers are encouraged to consult the reviews by [62] and [63] on the state of the art in this area.

In this paper, we employ the maximin Latin hypercube sampling design for specifying the initial sample n_0. The procedure for constructing the maximin LHS is listed in Algorithm 2.

Algorithm 2: Maximin Latin Hypercube Design with Equal Probability Intervals

Input : p: Number of variables,
n_0: Number of intervals,
F_j: Cumulative distribution function for each variable $\mathbf{x}_j$,
r_{ij}: Uniform random values.

Output: $n_0 \times p$ Maximin Latin Hypercube Design (LHD) matrix.

Step 1: Divide Cumulative Distribution
for *each variable* $j = 1$ *to* p **do**
  Divide cumulative distribution of $\mathbf{x}_j$ into n_0 equally probable intervals;

Step 2: Sample Cumulative Probabilities
for *each variable* $j = 1$ *to* p *and interval* $i = 1$ *to* n_0 **do**
  Calculate sampled cumulative probability:
  $s_{ij} = r_{ij}/n_0 + (i-1)/n_0$;

Step 3: Specify Sampled Values
for *each variable* $j = 1$ *to* p *and interval* $i = 1$ *to* n_0 **do**
  Compute sampled value x_{ij} as the quantile:
  $x_{ij} = F_j^{-1}(s_{ij})$;

Step 4: Pair Values Using Maximin Distance
Pair the n_0 values of each variable by maximising the minimum Euclidean distance;
Let the distance between points $\mathbf{x}_i$ and $\mathbf{x}_k$ be defined as:
$\rho_2(\mathbf{x}_i, \mathbf{x}_k) = \left(\sum_{j=1}^{p}(x_{ij} - x_{kj})^2\right)^{\frac{1}{2}}$;

Step 5: Determine Maximin Design
Identify the design $\mathcal{L}$ that maximises the minimum distance:
$x_{ij}^{lhs} = \max_{\mathcal{L}\in\mathcal{X}} \min_{\mathbf{x}_i,\mathbf{x}_k\in\mathcal{L}} \rho_2(\mathbf{x}_i, \mathbf{x}_k)$;
return $n_0 \times p$ *Maximin LHD matrix*

For the maximin Latin hypercube sampling design, however, the specified values for the input variables do not necessarily, and most often, exist in the actual data set under consideration for the active learner. Therefore, in this case, we select the sample points in the actual data set that are closest to the specified maximin LHD points by minimising the Euclidean distance between the LHD points and the actual samples.

3.2. Conditioned Latin Hypercube Sampling

[28] proposed a conditional LHS algorithm to sample ancillary data that form a Latin hypercube in the feature space. The method relies on a search algorithm based on heuristic rules and an annealing schedule. The objective function being minimised is the weighted sum of three components. The first component is the sum of the absolute differences between the marginal stratum sample sizes and the targeted sample size of 1, across all marginal strata. The second component is the sum of the absolute differences between the correlations of covariates in the population and in the sample, across all entries of the correlation matrix. The third component, applicable when both quantitative covariates and categorical variables are present, is the sum of the absolute differences between the sample proportions and the population proportions for each class of the categorical variables [59]. Therefore, actual samples in the $\mathbf{x}_j, j = 1, 2, \ldots, p$ input variables are selected that form a LHD for the training set $\mathcal{L}$. The algorithm for generating the conditional Latin hypercube sample (cLHS) is listed in Algorithm 3.

3.3. Modified Latin Hypercube Sampling

In this section, we present a similar but new approach to the cLHS. This approach retains the selection process of LHS with the added advantage of adaptable marginal spacings of the cLHS. Specifically, the modified Latin hypercube sampling procedure (mLHS) generates a Latin hypercube sample and adjusts it according to the estimated empirical distribution provided by the data. The result is that the one-dimensional projections for every dimension (i.e., marginal distributions) are not necessarily uniformly distributed.

The Hermite polynomial approach of [64] is used to estimate the cumulative distribution functions and subsequently the quantiles. Define $F^{-1}(s) = \inf\{x \in \mathbb{R} : F(x) \geq s\}$ as the quantile function where $F(x)$ is the distribution function of a random variable X. The estimated quantiles, $\widehat{q}_s = \widehat{F}^{-1}(s)$, are obtained through iteratively evaluating the relation,

$$\widehat{q}_s^{(i+1)} = \widehat{q}_s^{(i)} - \frac{\widehat{F}_M\left(\widehat{q}_s^{(i)}\right) - s}{\hat{f}_M\left(\widehat{q}_s^{(i)}\right)}, 0 \leq s \leq 1, \tag{6}$$

where

$$\widehat{f}_M(x) = \sum_{k=0}^{M} \hat{a}_k H_k(x) Z(x),$$

is the $M+1$ term truncated Gauss-Hermite expansion with estimated coefficients, $\widehat{a}_k = (\alpha_k/m)\sum_{i=1}^{m} Z(\mathbf{x}) H_k(\mathbf{x})$, $Z(\mathbf{x})$ is the standard normal probability density function, $\alpha_k = \sqrt{\pi}/(2^{k-1}k!)$ and $H_k(\mathbf{x})$ is an explicit calculable form of Hermite polynomials. Furthermore,

$$\widehat{F}_M(x) = \begin{cases} 1 - \sum_{k=0}^{M} \hat{a}_k k! \sum_{l=0}^{\lfloor k/2 \rfloor} \frac{(-1)^l 2^{\frac{3k}{2}-3l-1} \Gamma\left(-l+\frac{k}{2}+\frac{1}{2}, \frac{x^2}{2}\right)}{l!(k-2l)!\sqrt{\pi}} & \text{if } x \geq 0 \\ \sum_{k=0}^{M} \hat{a}_k k! \sum_{l=0}^{\lfloor k/2 \rfloor} \frac{(-1)^{-l+k} 2^{\frac{3k}{2}-3l-1} \Gamma\left(-l+\frac{k}{2}+\frac{1}{2}, \frac{x^2}{2}\right)}{l!(k-2l)!\sqrt{\pi}} & \text{if } x < 0 \end{cases},$$

where $\Gamma(a, x) = \int_x^{\infty} t^{a-1} e^{-t}\, \mathrm{d}t$ is the upper incomplete Gamma function. This procedure is attractive because of its speed and accuracy, as well as its ability to incorporate observations in an incremental fashion when needed. A general algorithm for the mLHS is given in Algorithm 4.

Algorithm 3: Conditional Latin hypercube sampling

Input : p: Number of variables,
n_0: Number of intervals,
Data for each variable $\mathbf{x}_j$,
Weights w_1 and w_2.
Output: Sampled data points $\tilde{x}_{ij}^{lhs}$ forming an LHS.

Step 1: Calculate Quantiles for Each Variable
for *each variable* $j = 1$ *to* p **do**
  Divide the cumulative distribution of $\mathbf{x}_j$ into n_0 intervals;
  Calculate the quantiles $\mathbf{q}_j = (q_{1j}, q_{2j}, \ldots, q_{n_0+1,j})^\top$;

Step 2: Compute Correlation Matrix C
Calculate the correlation matrix $\mathbf{C} : p \times p$ for the quantiles of each variable;

Step 3: Select Random Sample from Data
Select n_0 random samples from data, denoted by $\tilde{\mathbf{x}}_j : n_0 \times 1$ for each variable j;

Step 4: Compute Correlation Matrix B for Sample
Calculate the correlation matrix $\mathbf{B} : p \times p$ for the sampled values $\tilde{\mathbf{x}}_j$;

Step 5: Calculate Objective Function O_1
$O_1 = \sum_{i=1}^{n_0} \sum_{j=1}^{p} |\eta_{ij}(q_{ij} \leq \tilde{x}_{ij} \leq q_{i+1,j}) - 1|$, where η_{ij} is the count of $\tilde{x}_{ij}$ values between quantiles q_{ij} and $q_{i+1,j}$;

Step 6: Calculate Objective Function O_2
$O_2 = \sum_{j=1}^{p} \sum_{k=1}^{p} |c_{jk} - b_{jk}|$, where c_{jk} and b_{jk} are elements of correlation matrices $\mathbf{C}$ and $\mathbf{B}$, respectively;

Step 7: Solve Final Objective Function
Define final objective function as:
$\tilde{x}_{ij}^{lhs} = \min_{\tilde{x}_{ij} \in \mathcal{X}} w_1 O_1 + w_2 O_2$;
Set weights $w_1 = w_2 = 1$ (for general application, can be adjusted as needed);

Step 8: Optimize Using Heuristic Algorithm
Apply a suitable heuristic optimization algorithm to minimise the objective function;
return *LHS sample points* $\tilde{x}_{ij}^{lhs}$;

Algorithm 4: Generalized Latin Hypercube Sampling with Quantile Redistribution

Input : N: Number of samples,
D: Number of dimensions,
$data$: Matrix of observed data.
Output: S: Latin Hypercube sample matrix,
G: Grid matrix for visualization.

Step 1: Generate Latin Hypercube Design
Generate sequence $Lseq = -(N-1)/2$ to $(N-1)/2$;
initialise $L \leftarrow$ matrix(N, D), and for each $d \in 1 \ldots D$: $L[, d] \leftarrow$ random permutation of $Lseq$;
Generate uniform random matrix $U \leftarrow$ matrix(N, D);
Compute $S \leftarrow (L + (N-1)/2 + U)/N$ and grid $G \leftarrow (L + (N-1)/2)/N$;
Append a row of 1s to G;

Step 2: Quantile Redistribution
for $d \leftarrow 1$ **to** D **do**
  Apply `Quantile(`$data[, d], S[, d]$`)` to $S[, d]$;
  Apply `Quantile(`$data[, d], G[, d]$`)` to $G[, d]$;
return S, G;

Similar to LHS, the sample points in the actual data set that are closest to the specified mLHS points have to be selected. This is done by minimising the Euclidean distance between the mLHS points and the actual samples. Note the estimation of the quantiles is based on the entire set of unlabelled data. Furthermore, any quantile redistribution method can be used within Algorithm 4. Thus, it should be noted that mLHS and LHS may be more computationally intensive than cLHS, owing to the need to calculate distances between LHS points and the observed points. Furthermore, mLHS incurs additional computational cost due to the iterative quantile estimation and Hermite polynomial computations

To summarise the three sampling procedures, LHS provides stratified coverage of the input space, cLHS improves realism by aligning samples with observed data distributions, and mLHS enhances efficiency by placing more points in high-density regions, which potentially reduces sampling variability. In other words, the cLHS strives to match sample and population correlations, while the mLHS aims to achieve a Latin Hypercube structure based on the estimated marginal distributions, which may be more robust for certain types of data heterogeneity.

4. Simulation study

4.1. Simulation design

In this section, we discuss the simulation study and the results for binary classification as a function of two variables. For illustration purposes, three different synthetic data sets are considered. The three data sets are shown in Figure 1 as two-dimensional scatter plots, together with the two classes indicated on each plot. The data sets are referred to as Data 1, Data 2 and Data 3, respectively, with $N_1 = 1000, N_2 = 1000$ and $N_3 = 1250$ number of observations, respectively. Data 3 is a synthetic dataset found in the *MASS* package [75]. As can be observed, the three simulated data sets differ in complexity for binary classification.

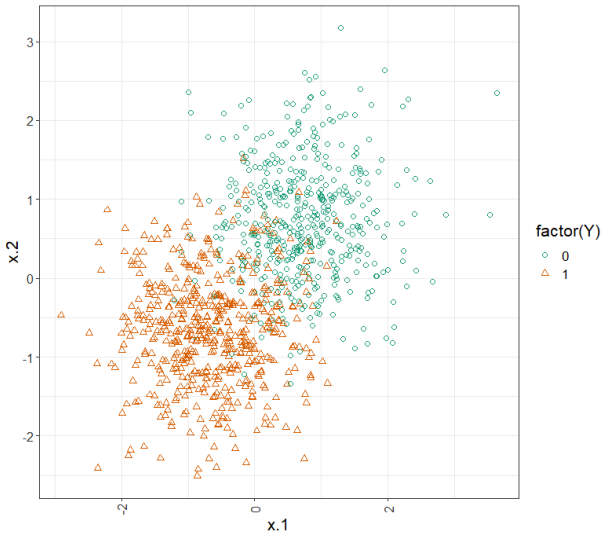
For each of the data sets, $\{\mathbf{x}_i, Y_i\}$, $i = 1, \ldots, N$, the simulation design employed involves a structured sequence of steps. First, the data is divided into test and training sets in a 50:50 ratio. An initial sample of size n_0 is selected from this training set and denoted $\mathcal{L}$. The logistic regression model is fitted to this set and the statistical properties of the model and the performance measures (given in Section 2.1 and Section 2.3) of the classifier are calculated using the test set. Using the uncertainty criterion (2.2), a new observation is selected from the unlabelled data in the training set, $\mathcal{U}$ and added to the initial set. This process of fitting the model and calculating the statistics is repeated for $150 \leq n_1 \leq 300$ iterations. The entire procedure, from splitting the data to the model fitting and evaluation, is repeated for $MC = 1000$ simulations. Finally, the mean of all the statistics is calculated over the MC simulations to assess the overall performance.

The aim of the simulation study is to evaluate the effect of the initial design and the size of the design on the performance of the classifier and the model estimation. Four types of initial designs are considered for specifying the initial training set, n_0. These are RS, LHS, cLHS and mLHS. The following R packages, *lhs* and *clhs* by [73] and [74], were used for the initial designs. The mLHS utilizes the hermiter package [76]. The initial sample sizes were varied with $n_0 = 8, 15, 50$ for all four sampling designs. The R code and data are available upon request.

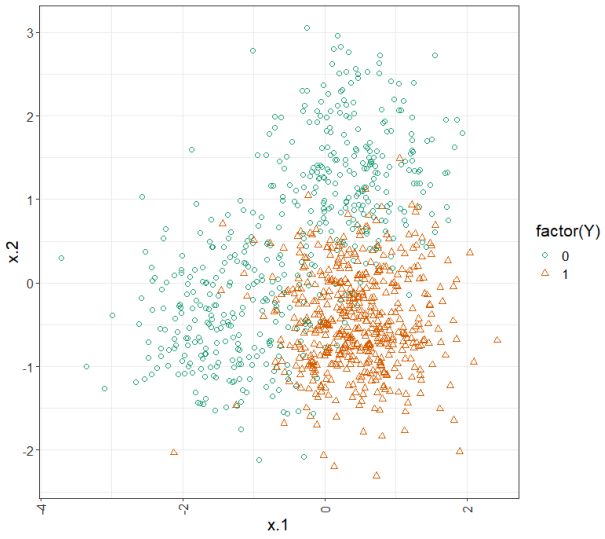
4.2. Results and discussion

Figure 2 shows the learning curves of the classification accuracy for Data 1 for initial sample sizes of $n_0 = 8$ and $n_0 = 50$, respectively. The horizontal dotted line is the maximum average accuracy that may be achieved if all the observations are used for classification. Note in cases where the accuracy of the different designs goes above the maximum accuracy for all observations, the differences in accuracy are very small and might be due to random variations in the MC simulations and round-off effects.

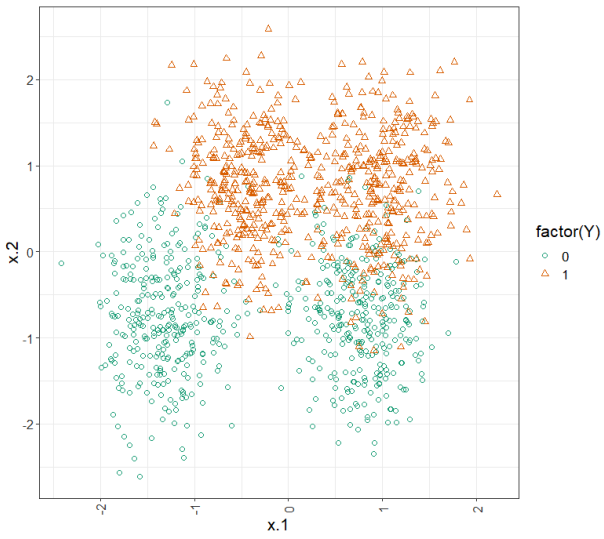
From Figure 2a, considering an initial starting design with $n_0 = 8$, it is observed that the accuracy for LHS and mLHS designs increases at the fastest rate from the initial size to about $n = 50$. From Figure 2b, for a larger initial sample size of $n_0 = 50$, the accuracy for the cLHS design increases faster compared to the other designs



(a) Data 1



(b) Data 2



(c) Data 3

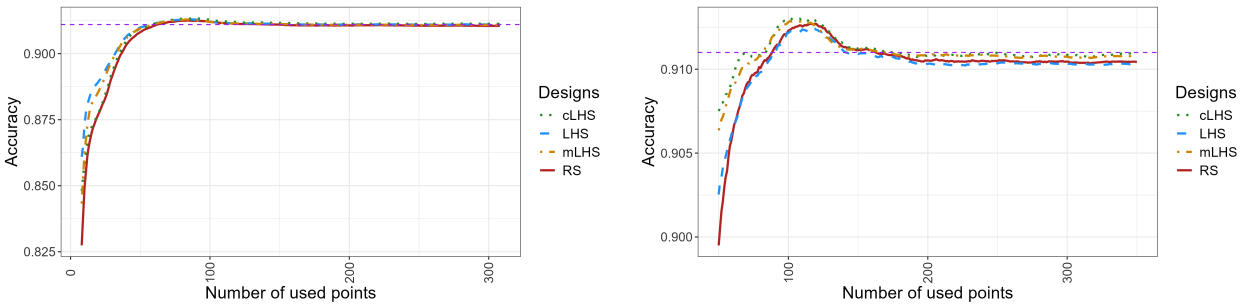
Figure 1. Scatter plots of the three synthetic datasets

and achieves the maximum accuracy soonest, about $n = 65$. Requiring a larger initial sample for the cLHS design is sensible since actual data points are selected that form a Latin hypercube sample as close as possible. Therefore, more data points are required to sample from the binary distribution of the data. The mLHS design also yields similar performance but requires slightly more points for the highest accuracy. This might be due to the Gauss-Hermite approximation requiring more observations for estimating the marginal distributions. Above about $n = 80$, the accuracy for all the designs is very similar. However, all three design approaches outperform the RS approach in terms of accuracy for initialising the active learning process.

Figure 3 shows the learning curves of the classification accuracy for Data 2 for initial sample sizes of $n_0 = 8$ and $n_0 = 50$, respectively. From Figure 3a, considering an initial starting design with $n_0 = 8$, it is observed that the accuracy for the cLHS design increases at the fastest rate from the initial size to about $n = 68$. From Figure 3b

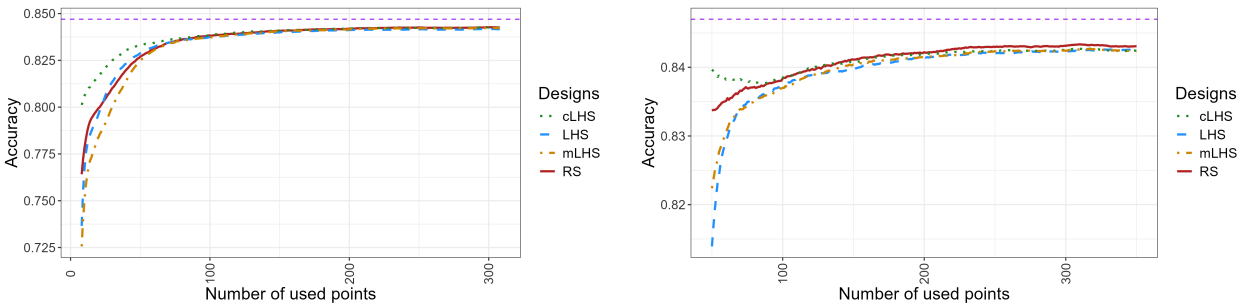
with $n_0 = 50$, it is observed that the cLHS design yields the greatest accuracy from initialization to about $n = 79$. RS outperforms mLHS and LHS in this case. Data 2 is more complex in that the two classes exhibit variance heterogeneity. The cLHS design samples the actual data points and therefore captures the differences in variability more easily. The mLHS design utilises the estimated marginal distributions, and it seems that it lacks efficiency in the estimation for heterogeneous classes. For heterogeneous classes, a greater sample size is required to achieve comparable accuracy compared to Figure 2.

Figure 4 shows the learning curves of the classification accuracy for Data 3 for initial sample sizes of $n_0 = 8$ and $n_0 = 50$, respectively. From Figure 4a, considering an initial starting design with $n_0 = 8$, it is observed that the accuracy for the cLHS design increases at the fastest rate from the initial size to about $n = 66$. Maximum accuracy is achieved at about $n = 80$ for all designs. From Figure 4b with $n_0 = 50$, it is again observed that the cLHS design achieves the greatest accuracy at about $n = 74$. Beyond $n > 74$, the accuracy either stabilises or exhibits minimal improvement across all designs. Data 3 exhibits four clusters, and it is expected that the classifier will take longer to achieve acceptable accuracy. However, from the results, it can be concluded that the cLHS design yields good accuracy early on for homogeneous and heterogeneous classes.



(a) $n_0 = 8$ (b) $n_0 = 50$

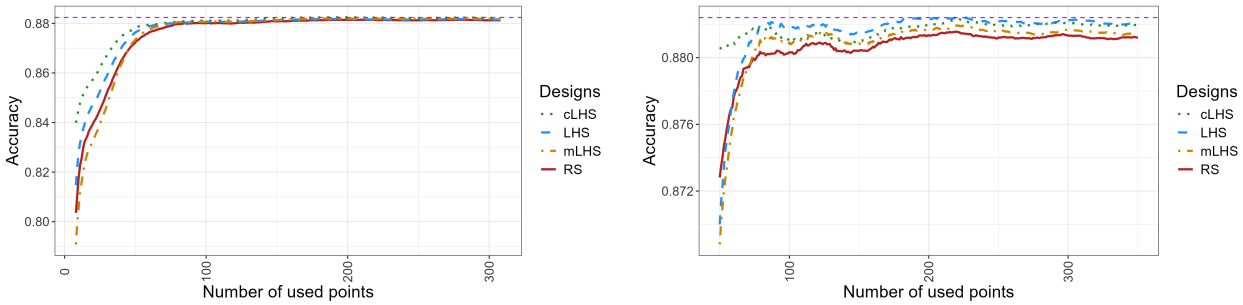
Figure 2. Accuracy for different designs plotted against increasing sample size for Data 1.



(a) $n_0 = 8$ (b) $n_0 = 50$

Figure 3. Accuracy for different designs plotted against increasing sample size for Data 2.

Figure 5 shows the average variance of the predicted posterior probability ($\mathbb{V}(\widehat{\pi}_i)$) across the design space for Data 1 for initial sample sizes $n_0 = 15$ and $n_0 = 50$, respectively. There is huge variability in the variance of prediction for $n \leq 80$. The reason is that random sampling and Latin hypercube designs, which are space-filling designs and not model-based optimal designs, can select samples that yield an information matrix that is very close to singularity and results in huge variances of the parameter estimates and variance of prediction. Therefore, we only show the results from $n = 80$ and higher. This is not a concern since the results for accuracy indicated



(a) $n_0 = 8$ (b) $n_0 = 50$

Figure 4. Accuracy for different designs plotted against increasing sample size for Data 3.

that the accuracy stabilizes from about $n = 60$ and higher for all the designs and data sets.

From Figure 5a, it is observed that the average $\mathbb{V}(\widehat{\pi}_i)$ is the smallest for the mLHS design for Data 1. The variance stabilizes at about $n = 200$ for the mLHS, cLHS, and RS designs. The LHS yields the worst average variance of prediction. Similarly, from Figure 5b it is observed that the average $\mathbb{V}(\widehat{\pi}_i)$ is the smallest for the mLHS design. The smallest variance is obtained at about $n = 200$ for the mLHS, cLHS, and RS designs. Again, the LHS yields the worst average variance of prediction. The variance of prediction stabilizes earlier when the active learning process is initialised with a smaller sample size.

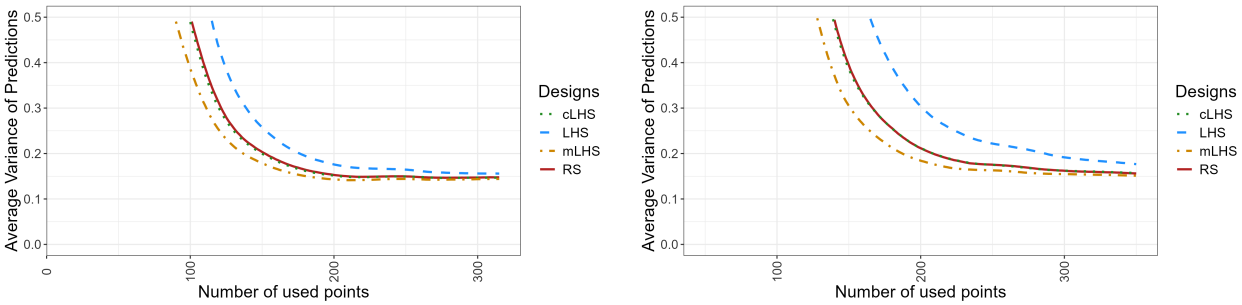
From Figure 6a for $n_0 = 15$, it is observed that the average $\mathbb{V}(\widehat{\pi}_i)$ is the smallest for the mLHS design for Data 2, up to about $n = 139$. The variance continues to decrease until $n = 308$. From Figure 6b for $n_0 = 50$, it is observed that the average $\mathbb{V}(\widehat{\pi}_i)$ is significantly smaller for the mLHS design, compared to the other designs until about $n = 243$. The smaller variance of prediction is obtained sooner when the active learning process is initialised with a smaller sample size. The LHS yields the worst average variance of prediction for both initial design sizes.

From Figure 7a for $n_0 = 15$, it is observed that the average $\mathbb{V}(\widehat{\pi}_i)$ is comparable for the three designs for Data 3. The variance continues to decrease until $n = 308$. From Figure 7b for $n_0 = 50$, it is observed that the average $\mathbb{V}(\widehat{\pi}_i)$ is smaller for the mLHS design, compared to the other designs, until about $n = 285$. Again, the LHS yields the worst average variance of prediction for both initial design sizes.

From the results for the average variance of prediction, it can be concluded that the mLHS design performs best for different initial sample sizes and for homogeneous and heterogeneous classes. Therefore, selecting samples from a grid over the design space that is based on the estimated marginal probability distributions given by the data yields the best results. In addition, the active learning process should be initialised with a smaller sample size when the variance of the predicted posterior probability is the main criterion.

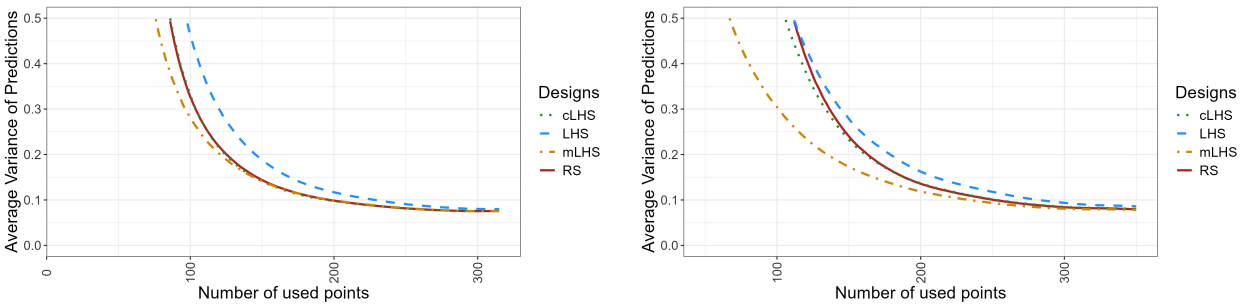
Figure 8 shows the mean squared error of the slope of the logistic classifier ($\mathrm{MSE}(\widehat{m}_1)$) across the design space for Data 1 for initial sample sizes $n_0 = 15$ and $n_0 = 50$, respectively. We only plot the results from $n = 80$ and higher due to huge variability resulting from an ill-condition information matrix for smaller sample sizes. Again, this is not a concern since the results for accuracy indicated that the accuracy stabilizes from about $n = 60$ and higher for all the designs and data sets.

From Figure 8a for $n_0 = 15$, it is observed that the $\mathrm{MSE}(\widehat{m}_1)$ is the greatest for RS for Data 1. The $\mathrm{MSE}(\widehat{m}_1)$ is very similar for the other three designs. The $\mathrm{MSE}(\widehat{m}_1)$ stabilizes at about $n = 200$ for all four designs. From Figure 8b for $n_0 = 50$, it is observed that the $\mathrm{MSE}(\widehat{m}_1)$ is the smallest for the mLHS design until about $n = 143$. The $\mathrm{MSE}(\widehat{m}_1)$ only stabilises at about $n = 200$ for all four designs. Therefore, the mLHS design yields the



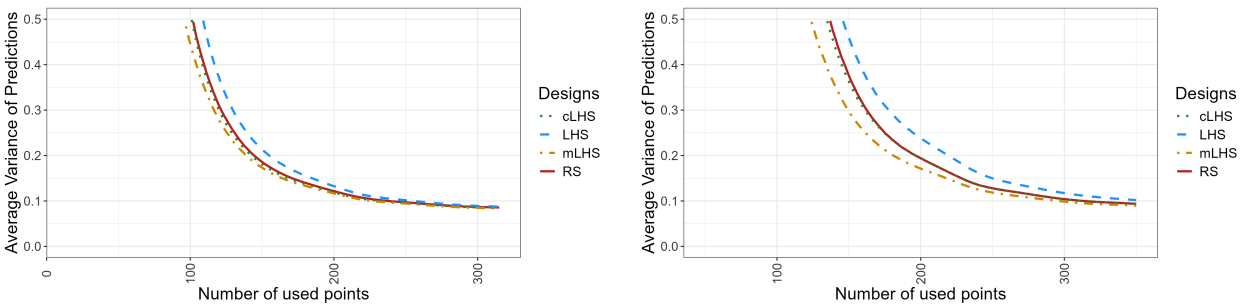
(a) $n_0 = 15$ (b) $n_0 = 50$

Figure 5. Average variance of the predicted posterior probability ($\mathbb{V}(\widehat{\pi}_i)$) for different designs plotted against increasing sample size for Data 1.



(a) $n_0 = 15$ (b) $n_0 = 50$

Figure 6. Average variance of the predicted posterior probability ($\mathbb{V}(\widehat{\pi}_i)$) for different designs plotted against increasing sample size for Data 2.



(a) $n_0 = 15$ (b) $n_0 = 50$

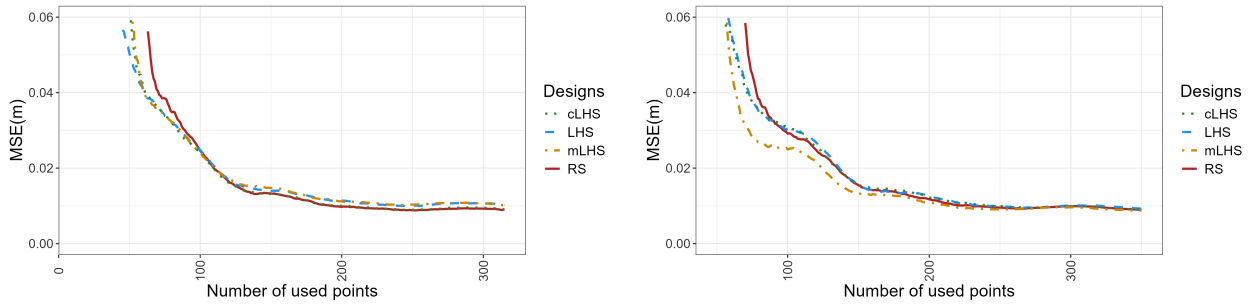
Figure 7. Average variance of the predicted posterior probability ($\mathbb{V}(\widehat{\pi}_i)$) for different designs plotted against increasing sample size for Data 3.

smallest mean squared error of the slope of the classifier for $n_0 = 50$.

From Figure 9a for $n_0 = 15$, it is observed that the $\text{MSE}(\widehat{m}_1)$ is the smallest for the LHS and cLHS designs for Data 2. The $\text{MSE}(\widehat{m}_1)$ stabilizes at about $n = 120$ for all four designs. From Figure 9b for $n_0 = 50$, it is observed that the $\text{MSE}(\widehat{m}_1)$ is the smallest for the cLHS initially until $n = 77$, but the LHS design yields the smallest $\text{MSE}(\widehat{m}_1)$ from about $n = 78$ until about $n = 151$. Therefore, the cLHS design yields the best performance for

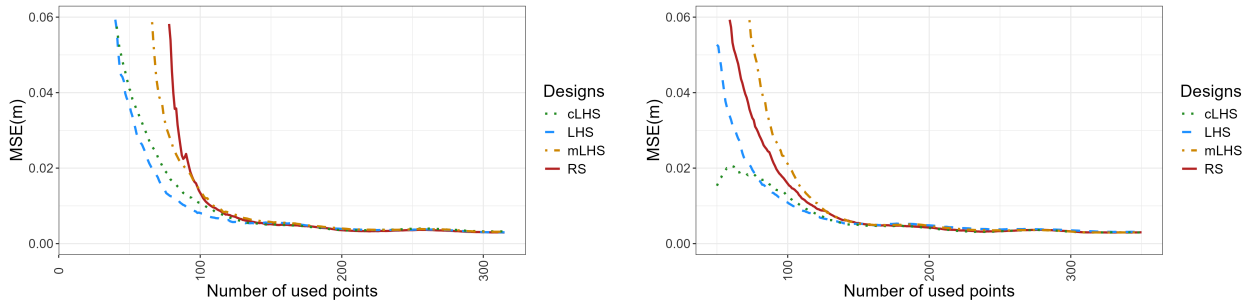
$\text{MSE}(\widehat{m}_1)$ for heterogeneous classes. Similar trends were observed for the accuracy.

From Figure 10a for $n_0 = 15$, it is observed that the $\text{MSE}(\widehat{m}_1)$ is the smallest for the cLHS and LHS designs for Data 3. The $\text{MSE}(\widehat{m}_1)$ stabilizes at about $n = 100$ for all four designs. From Figure 10b for $n_0 = 50$, it is observed that the $\text{MSE}(\widehat{m}_1)$ is the smallest for the cLHS design initially; however, the results are very similar from about $n = 73$ for all four designs. Overall, the cLHS design yields the best performance for $\text{MSE}(\widehat{m}_1)$ for Data 3. Note that the effect of the initial design and the design size on the mean square error of the classifier was not communicated before in the literature.



(a) $n_0 = 15$ (b) $n_0 = 50$

Figure 8. Mean squared error of the slope of the logistic classifier ($\text{MSE}(\widehat{m}_k)$) for different designs plotted against increasing sample size for Data 1.

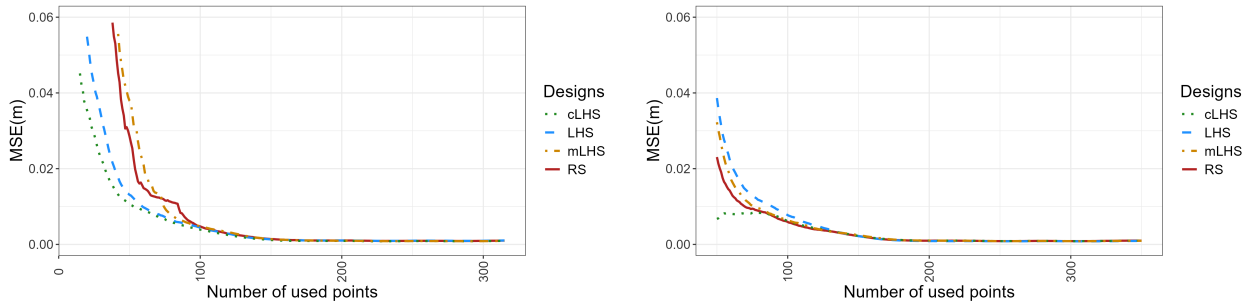


(a) $n_0 = 15$ (b) $n_0 = 50$

Figure 9. Mean squared error of the slope of the logistic classifier ($\text{MSE}(\widehat{m}_k)$) for different designs plotted against increasing sample size for Data 2.

The results discussed so far indicate that the design approaches, i.e., LHS, cLHS and mLHS, used to initialise the active learning process together with uncertainty sampling outperform RS for classification accuracy, average variance of the predicted posterior probability and the mean squared error of the slope of the classifier. Both the cLHS and mLHS designs yield very good results.

However, as discussed in Section 2.3, there are many classification performance measures, including, but not limited to, accuracy, precision, sensitivity, F1-score, and AUC. In this paper, we also consider various statistics of the logistic regression model and the binary classifier, such as the generalised variance of the parameter estimates ($|\mathbf{M}(\mathbf{x}, \mathbf{fi})^{-1}|$), the variance of the predicted posterior probability ($\mathbb{V}(\widehat{\pi}_i)$), the mean squared error of the slope of the classifier ($\text{MSE}(\widehat{m}_k)$) and the mean squared error of the intercept of the classifier ($\text{MSE}(\widehat{c})$). However, the best initial design, initial sample size, and recommended number of samples needed to train the classifier will depend



(a) $n_0 = 15$ (b) $n_0 = 50$

Figure 10. Mean squared error of the slope of the logistic classifier ($\text{MSE}(\widehat{m}_k)$) for different designs plotted against increasing sample size for Data 3.

on the criterion of interest.

Therefore, to evaluate the effect of the various scenarios on the model estimation and classifier performance measures simultaneously, as well as to assess the correlations between all the different criteria, we performed principal component analysis (PCA) and biplot visualization on the results for each data set. PCA is a dimension reduction method that creates new latent variables as a linear combination of the variables in the data. The coefficients of the linear combination are referred to as the loadings of the input variables, and the predictions of the latent variables for a set of input variables are the scores. The first principal component explains the maximum variance in the data, the second principal component explains the second most variability in the data, and is uncorrelated with the first component, etc. Typically, the first two principal components explain the greatest cumulative variance in the data. The results can be visualized on a two-dimensional PCA biplot, which shows the scores and loadings on the same plot. See [19] for a detailed discussion of PCA and biplots.

In this case, the variables in the data set are the evaluation criteria, which are nine in total. The results for all four designs were combined for each data set. The PCA biplots are presented for $80 \leq n \leq 200$. For $n \leq 80$, the substantial variability in the values of $\mathbb{V}(\widehat{\pi}_i)$, $|\mathbf{M}(\mathbf{x}, \boldsymbol{\beta})^{-1}|$, $\text{MSE}(\widehat{m}_k)$, and $\text{MSE}(\widehat{c})$ renders their interpretation less reliable. For $n > 200$, any improvements in the performance measures are expected to be marginal, as illustrated with the learning curves discussed above. Consider the PCA biplot in Figure 11. The first two principal components explain over 90% of the variability in the data, i.e., the representation of the data in two dimensions is very accurate. The rays are the loadings that indicate the direction of increasing values of the variables, and the angles between the rays approximate the correlations between the variables. We used the package *factorextra* in R to generate the biplots [3].

From the PCA biplot in Figure 11 for Data 1 and $n_0 = 15$, it is observed that there are noticeable differences in the performances of the designs and an effect of the number of training samples on the performance measures. Considering the effect of the number of points used for training, it is observed that the AUC and precision increase with increasing sample size from about $n = 120$ to $n = 200$. The cLHS and mLHS designs yield higher accuracy and AUC for lower number of training samples, and the highest accuracy at $n = 200$. The standardized variance of the parameter estimates, the average variance of prediction, and the mean squared error of the slope of the classifier are positively correlated and decrease with increasing sample size up to about $n = 120$. Beyond n=120, there is little effect of the increasing sample size on the precision of the classifier and the accuracy. Accuracy and F1-score are positively correlated with the standardized variance of the parameter estimates, the average variance of prediction, and the mean squared error of the slope of the classifier, which illustrates that higher precision of the classifier does not yield higher accuracy or F1-score of the classifier. However, precision and recall are relatively uncorrelated with the precision of the classifier, accuracy and F1-score. Therefore, although higher accuracy and

precision of the classifier are obtained for lower number of training samples, it yields average precision and recall. In summary, Figure 11 illustrates that a training sample of about $n = 120$ is sufficient for acceptable accuracy and a precise logistic classifier, achieving average classification accuracy, recall, and AUC. Moreover, the cLHS design yields the best results overall for $n = 120$.

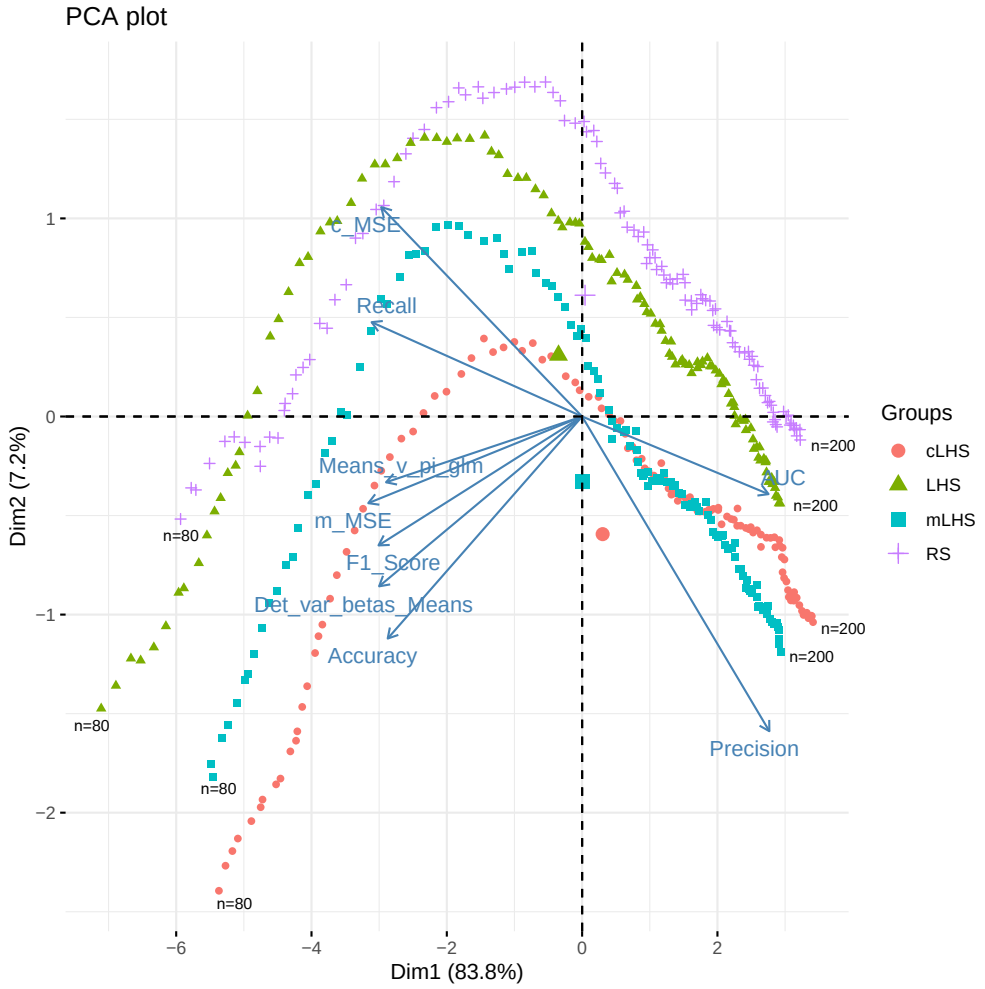


Figure 11. PCA biplot for $n_0 = 15$ for Data 1. (*Means_v_pi_glm* = $V(\widehat{\pi}_i)$, *Det_var_betas_Means* = $|\mathbf{M}(\mathbf{x}, \boldsymbol{\beta})^{-1}|$, *m_MSE* = $\text{MSE}(\widehat{m})$, *and c_MSE* = $\text{MSE}(\widehat{c})$)

Consider the PCA biplot in Figure 12 for Data 2 and $n_0 = 15$; the first two principal components explain about 94% of the variability in the data. From the biplot, it is observed that there are again differences between the designs, especially at the smaller sample size from $n = 80$ to $n = 120$. However, for the cLHS, mLHS and RS designs, the standardized variance of the parameter estimates and the mean squared error of the slope of the classifier decrease with increasing the sample size to about $n = 120$. As the AUC and F1-score increase, the standardized variance of the parameter estimates and the mean squared error of the slope decrease for increasing sample size to about $n = 120$ for the mLHS, cLHS and RS designs. For a training sample of about $n = 120$, the mLHS design yields the highest recall and average performance for all other criteria. This can be derived by projecting the data points perpendicular to each of the variable axes. In summary, Figure 12 illustrates that a training sample of about $n = 120$

is sufficient for an accurate and precise logistic classifier and average classification accuracy, recall and AUC. The mLHS design yields the best results overall.

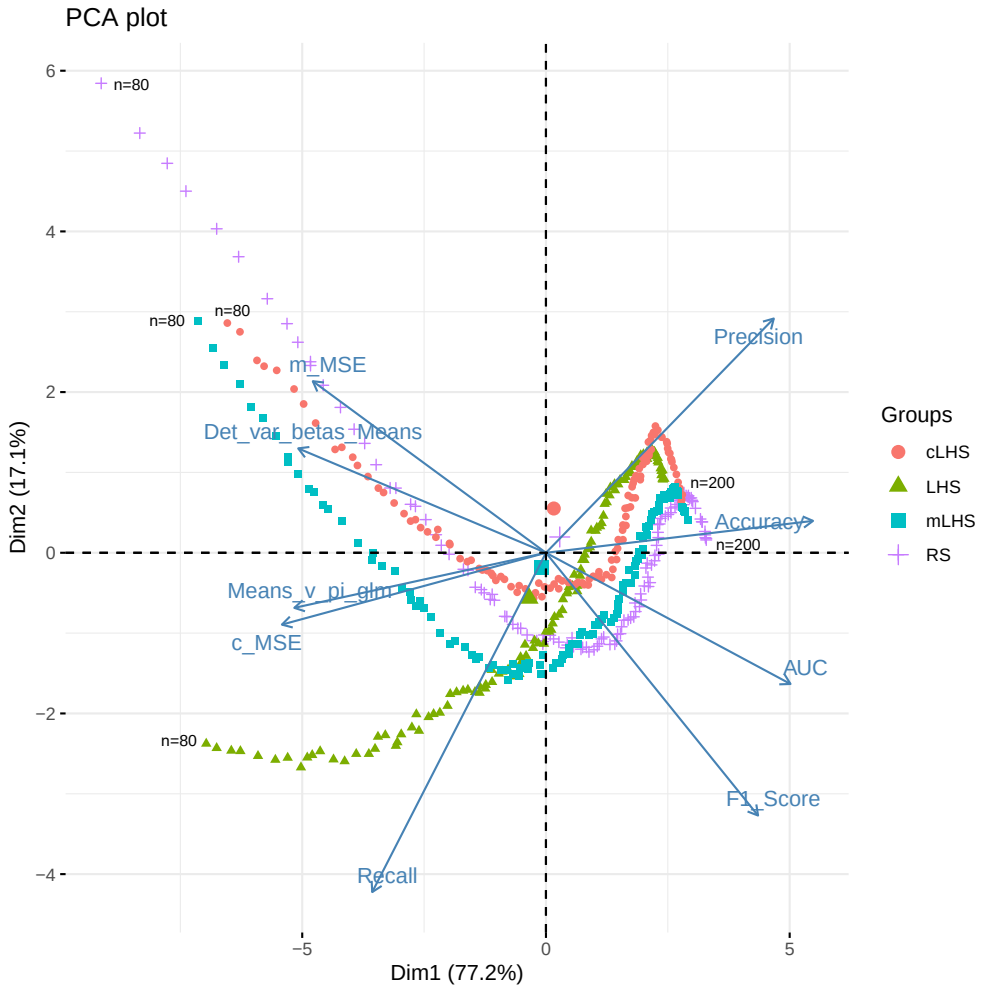


Figure 12. PCA biplot for $n_0 = 15$ for Data 2. (*Means_v_pi_glm* = $V(\widehat{\pi}_i)$, *Det_var_betas_Means* = $|\mathbf{M}(\mathbf{x}, \boldsymbol{\beta})^{-1}|$, *m_MSE* = $\text{MSE}(\widehat{m})$, *and c_MSE* = $\text{MSE}(\widehat{c})$)

Consider the PCA biplot in Figure 13 for Data 3 and $n_0 = 15$; the first two principal components explain almost 90% of the variability in the data. From the biplot, it is observed that there is a quadratic trend in the PCA scores for all four designs. Maximum recall is achieved at a sample size of about $n = 120$ for all four designs. The cLHS design yields the best precision, accuracy and F1-score irrespective of the size of the training sample. The mLHS design yields the maximum recall and minimum precision at about $n = 120$. As for Data 2, the standardized variance of the parameter estimates, the average variance of prediction and the mean squared error of the slope of the classifier decreases, and AUC increases for increasing sample size to about $n = 120$. In summary, Figure 13 clearly illustrates that a training sample of about $n = 120$ is sufficient for an accurate and precise logistic classifier and average classification accuracy, recall and AUC.

In summary, from the simulation study, Table 2 presents the recommended training sample sizes that achieves the best balance across all measures for the four different designs. It is noted that LHD requires fewer samples than random sampling across all three datasets. This trend is also evident in the PCA figures, Figure 11, Figure 12 and

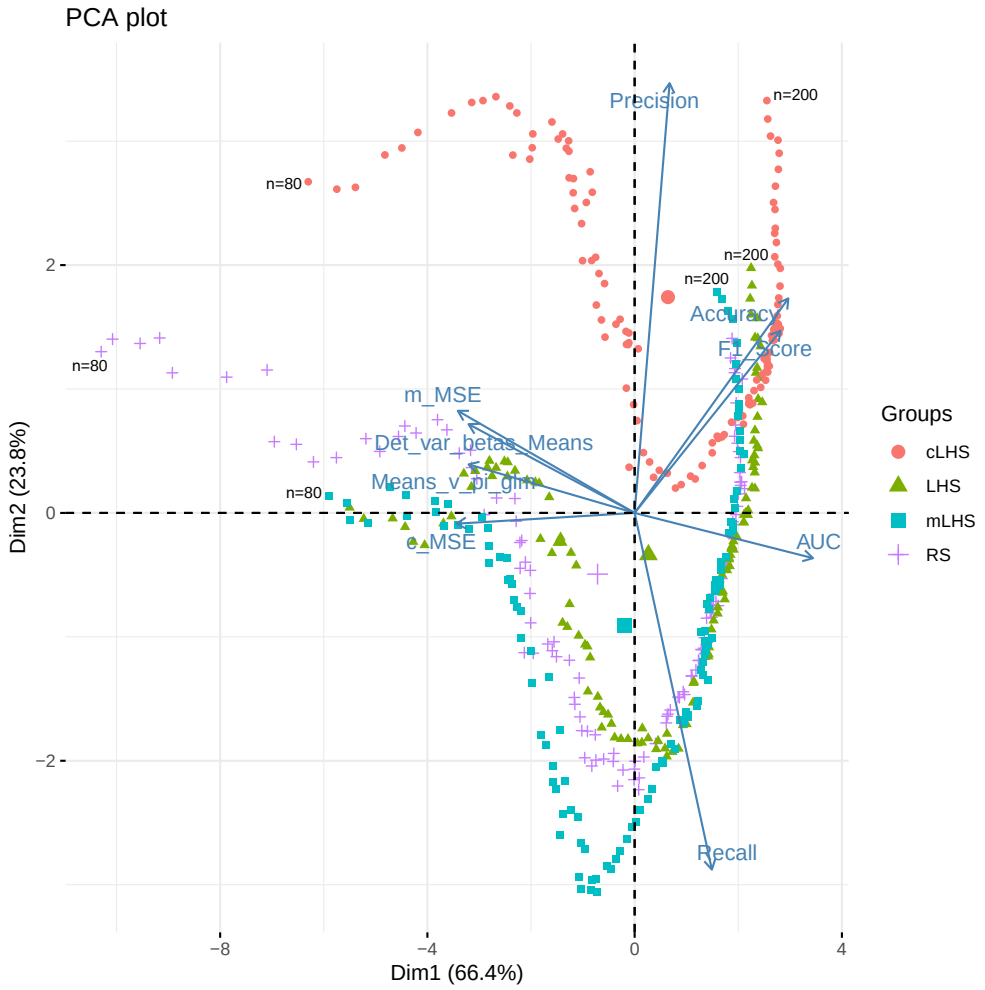


Figure 13. PCA biplot for $n_0 = 15$ for Data 3. (*Means_v_pi_glm* = $V(\widehat{\pi}_i)$, *Det_var_betas_Means* = $|\mathbf{M}(\mathbf{x}, \boldsymbol{\beta})^{-1}|$, *m_MSE* = $\mathrm{MSE}(\widehat{m})$, *and c_MSE* = $\mathrm{MSE}(\widehat{c})$)

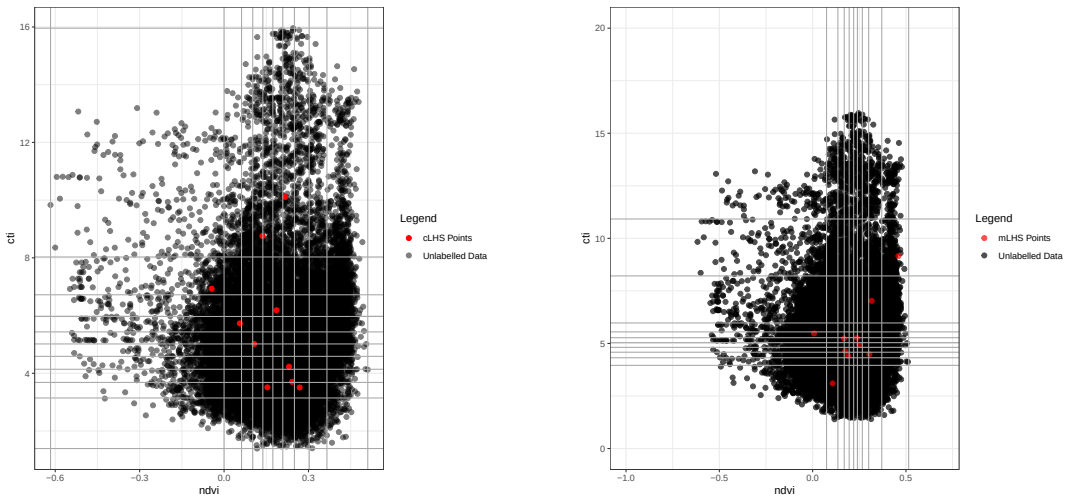
Figure 13, where the turning points can be observed approximately. The recommended training samples sizes are about 10% of the original size of the data sets.

Table 2. Recommended training sample sizes from the simulation study.

Design	Data 1	Data 2	Data 3
RS	120	133	138
LHS	110	92	135
cLHS	113	122	136
mLHS	108	120	126

5. Real world data example: Hunter Valley data

For a real-world example, we consider the Hunter Valley data from [59]. The authors used the cLHS design to sample representative data of five ancillary variables for four different land-use or vegetation classes. For illustration purposes, we will employ the logistic regression model to classify the observations into two land-use classes only, namely Native forest and viticulture, as a function of four predictor variables. The ancillary variables considered for the model are elevation, slope, compound topographic index (cti), and normalized difference vegetation index (ndvi). The dataset consists of 6710 observations, including 5509 observations from native forest and 1201 observations from viticulture. We evaluate the effect of the four different designs, i.e., RS, LHS, cLHS and mLHS on the performance of the classifier. For an illustration of the sampled points, Figure 14a show the cLHD sampled points and Figure 14b show the mLHD sampled points for $n_0 = 10$. However, in this section we will use $n_0 = 15$ and $n_0 = 50$, respectively, for evaluation.



(a) cLHD in 2-dimensions

(b) mLHD in 2-dimensions

Figure 14. Comparison of cLHD and mLHD in 2-dimensions

Figure 15a illustrates the learning curves of accuracy for $n_0 = 15$. The maximum accuracy is achieved by the mLHS design for a sample size of $n = 70$ and beyond. However, all the LHDs outperform the RS design when $n < 120$. Similarly, as shown in Figure 15b, with a larger initial design size of $n_0 = 50$, it is evident that the LHDs also outperform the RS design for $n < 120$. Specifically, the mLHS design achieves the maximum accuracy for sample sizes of around $n = 70$ and above. Since the dataset is highly class-imbalanced, Figure 16 shows the F1-score for $n_0 = 15$ and $n_0 = 50$, respectively. The F1-score and accuracy yield similar trends.

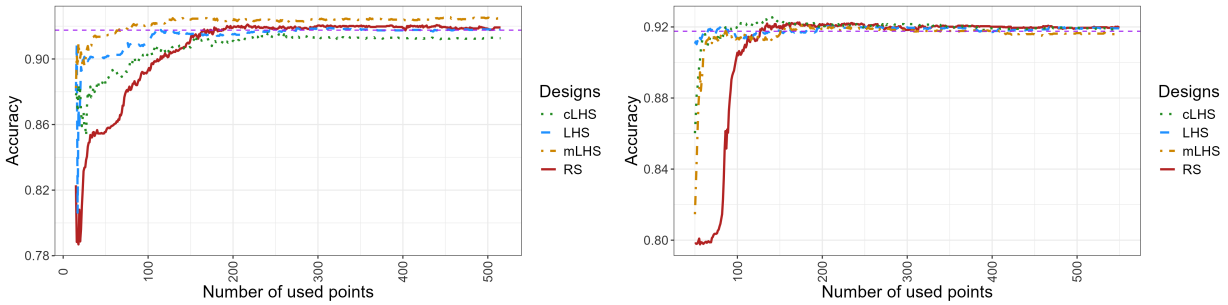
Figure 17a presents the learning curves of the AUC for $n_0 = 15$. The LHDs outperform the RS design for $n < 140$. In Figure 17b, where an initial design size of $n_0 = 50$ is used, the LHDs demonstrate superior performance until $n = 84$.

Figure 18a illustrates the learning curves of the variance of prediction for $n_0 = 15$, showing that the cLHS design results in the smallest variance of prediction. Similarly, as observed in Figure 18b, with $n_0 = 50$, the cLHS design maintains the smallest variance of prediction up to $n = 212$.

In summary, these results demonstrate that the experimental design approaches significantly outperform the RS design, particularly for smaller training sample sizes.

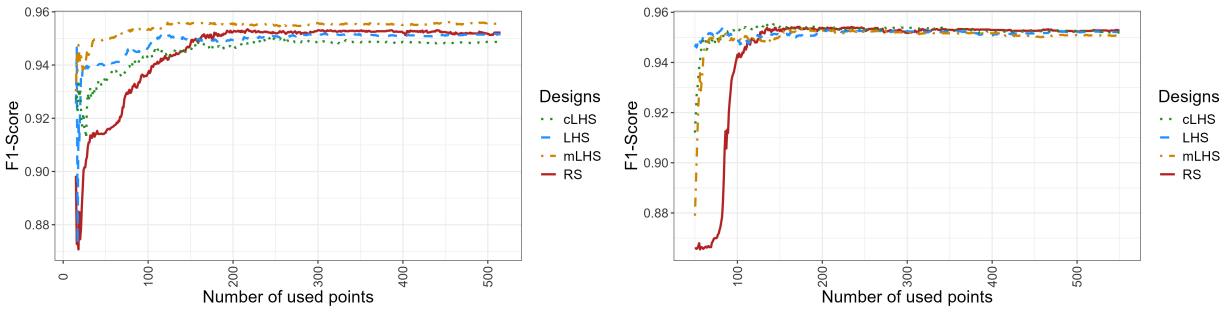
To assess the combined effects of the designs and initial sample sizes on the various evaluation criteria, Figure 19 provides the PCA biplot for $n_0 = 50$. The first two principal components account for approximately 89% of the variability in the data. The plot highlights distinct differences between the experimental designs and the RS design in terms of performance across all evaluation criteria. Notably, the RS design requires a larger number of training

samples to achieve a performance level comparable to that of the other designs. The mLHS and cLHS designs are similar in performance. The performance measures do not change much for increasing the sample size for the LHS design.



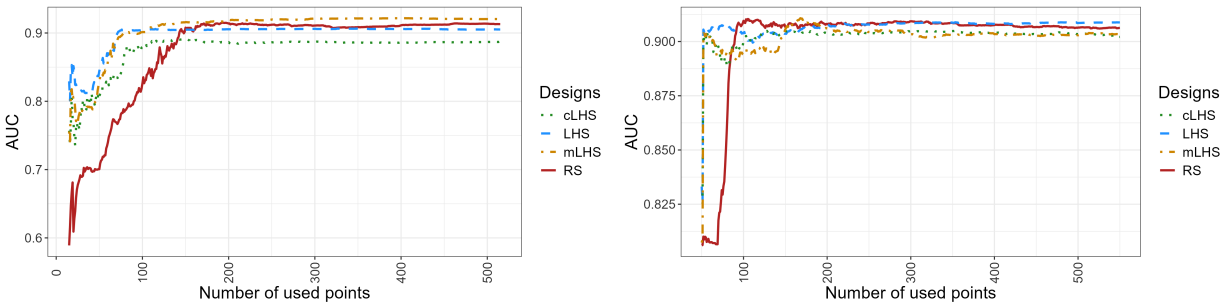
(a) $n_0 = 15$ (b) $n_0 = 50$

Figure 15. Accuracy plot for the Hunter Valley data for different initial sample sizes



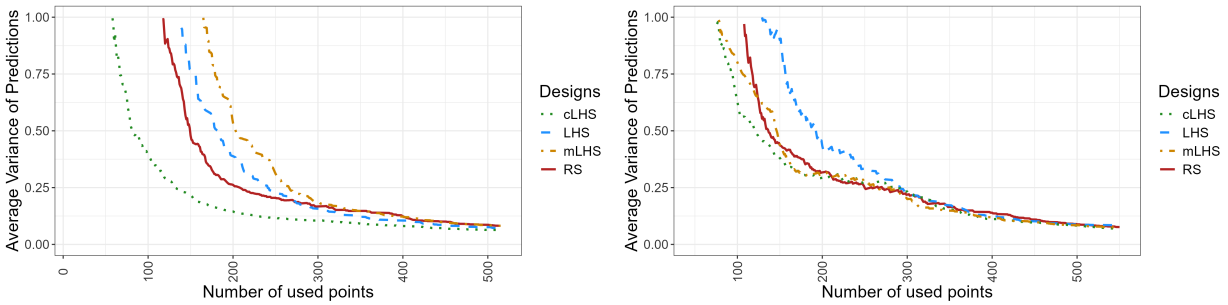
(a) $n_0 = 15$ (b) $n_0 = 50$

Figure 16. F1-Score plot for the Hunter Valley data for different initial sample sizes



(a) $n_0 = 15$ (b) $n_0 = 50$

Figure 17. AUC plot for the Hunter Valley data for different initial sample sizes



(a) $n_0 = 15$ (b) $n_0 = 50$

Figure 18. Variance of predictions plot for the Hunter Valley data for different initial sample sizes

Figure 19. PCA biplot of the Hunter Valley data for $n_0 = 50$. (*Means_v_pi_glm* = $V(\widehat{\pi}_i)$, *Det_var_betas_Means* = $|\mathbf{M}(\mathbf{x}, \boldsymbol{\beta})^{-1}|$)

6. Conclusion

In this paper, the performance of three Latin hypercube designs, namely, the maximin Latin hypercube, the conditioned Latin hypercube, and the modified Latin hypercube are compared to random sampling when selecting the initial points to initiate the active learning process. Furthermore, we demonstrated the effectiveness of active learning in reducing the number of points required to train an accurate model by employing the uncertainty-based active learning criterion. Several performance measures were utilised, including accuracy, the F1-score, the AUC, the average variance of predictions, the standardised variance of parameter estimates, and the mean squared error of the slope and intercept of the classifier. It was shown that the performance of the designs varies depending on the sample size, the specific performance measure under consideration, and the complexity of the dataset. The Latin hypercube designs yield better results compared to the random sampling for all the performance measures evaluated. It can be concluded that the cLHS design yields high accuracy, F1-score, and AUC at an early stage for homogeneous classes; however, it necessitates a larger initial sample size for heterogeneous classes. When prioritising the variance of the predicted posterior probability as the primary criterion, the mLHS design demonstrates superior performance across different initial sample sizes and for both homogeneous and heterogeneous classes. The results from the three simulated datasets, based on a combined performance assessment via the PCA, indicate that approximately 120 to 140 points out of $1,000$ to $1,250$ are sufficient to construct an accurate and precise logistic classifier, achieving high classification accuracy, recall, and AUC.

Since the seminal paper by [6], there have been many extensions and improvements for generating Latin hypercube designs with specific advantageous multivariate properties, such as orthogonal Latin hypercube designs [5, 4]. Utilising subsampling in active learning when selecting the most informative data points may reduce redundancy, enhance classification accuracy, and improve generalisation while minimising computational and labelling costs. The mLHS exhibited highly promising results in comparison with the cLHS; therefore, it would be beneficial to modify it to select actual observations directly from the training set, as is the case with the cLHS. Furthermore, future research will extend this framework by incorporating more complex classifiers, such as support vector machines and neural networks, and by comparing the proposed designs with advanced initialisation and query strategies (e.g., query-by-committee). In addition, assessing performance on highly correlated or high-dimensional datasets will provide further insight into the robustness of these sampling designs in more complex data structures.

Acknowledgement

The authors appreciate the reviewers' constructive feedback, which has contributed to enhancing the quality of this work.

Declarations

Conflict of interest: The authors have no conflict of interest to declare.

A. Derivation of $\mathbb{V}(\widehat{\pi}_i)$.

For convenience, in the following derivation, the subscripts are suppressed. The predicted probability $\widehat{\pi} = h(\widehat{\eta})$ is a nonlinear function of $\widehat{\eta}$. To approximate its variance, we linearize $h(\widehat{\eta})$ around the true η using a first-order Taylor expansion. To this end, let $\widehat{\eta} = \mathbf{x}^\intercal \hat{\boldsymbol{\beta}}$, we obtain

$$\widehat{\pi} = h(\widehat{\eta}) = \frac{e^{\widehat{\eta}}}{1 + e^{\widehat{\eta}}}.$$

The Taylor series expansion of $h(\widehat{\eta})$ in a neighbourhood of η yields,

$$\begin{aligned} h(\widehat{\eta}) &= h(\eta) + (\widehat{\eta} - \eta)\frac{\partial h}{\partial \eta} \\ &= h(\eta) + (\mathbf{x}^\intercal\widehat{\boldsymbol{\beta}} - \mathbf{x}^\intercal\boldsymbol{\beta})\pi(1-\pi) \\ &= h(\eta) + \pi(1-\pi)\mathbf{x}^\intercal(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}), \end{aligned}$$

since the derivative of $h(\eta) = \frac{e^\eta}{1+e^\eta}$ with respect to η is given by

$$\begin{aligned} \frac{\partial h}{\partial \eta} &= (e^\eta)^2(-1)(1+e^\eta)^{-2} + (1+e^\eta)^{-1}e^\eta \\ &= -\left(\frac{e^\eta}{1+e^\eta}\right)^2 + \frac{e^\eta}{1+e^\eta} \\ &= -\pi^2 + \pi \\ &= \pi(1-\pi). \end{aligned}$$

The term $\mathbf{x}^\intercal(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta})$ is a linear combination of the random vector $\widehat{\boldsymbol{\beta}}$. Its variance is given by,

$$\text{Var}\left[\mathbf{x}^\intercal(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta})\right] = \mathbf{x}^\intercal\text{Var}(\widehat{\boldsymbol{\beta}})\mathbf{x}.$$

This follows from the property of covariance matrices, $\text{Var}(\mathbf{AZ}) = \mathbf{A}\text{Var}(\mathbf{Z})\mathbf{A}^\intercal$, where $\mathbf{A}$ is a matrix (here, $\mathbf{x}^\intercal$).

The variance of $h(\widehat{\eta})$ is approximated by the variance of the linear term,

$$\text{Var}(h(\widehat{\eta})) \approx \text{Var}\left[\pi(1-\pi)\cdot\mathbf{x}^\intercal(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta})\right].$$

Since $\pi(1-\pi)$ is a scalar constant (for a fixed $\mathbf{x}$), we factor it out,

$$\text{Var}(h(\widehat{\eta})) \approx [\pi(1-\pi)]^2\cdot\text{Var}\left[\mathbf{x}^\intercal(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta})\right] = [\pi(1-\pi)]^2\cdot\mathbf{x}^\intercal\text{Var}(\widehat{\boldsymbol{\beta}})\mathbf{x}.$$

B. Derivation of $MSE(\widehat{c})$ and $MSE(\widehat{m})$

Let the intercept and slope of the classification boundary be defined as

$$\widehat{c} = q_1(\hat{\boldsymbol{\beta}}) = q_1(\hat{\beta}_0, \hat{\beta}_2) = -\frac{\hat{\beta}_0}{\hat{\beta}_2}$$

and

$$\widehat{m} = q_2(\hat{\boldsymbol{\beta}}) = q_1(\hat{\beta}_1, \hat{\beta}_2) = -\frac{\hat{\beta}_1}{\hat{\beta}_2}.$$

The first-order Taylor expansion of $\widehat{q_1}$ around the true parameters β_0 and β_2 is given by,

$$\widehat{q_1} \approx q_1 + (\widehat{\beta_0} - \beta_0)\frac{\partial q_1}{\partial \beta_0} + (\widehat{\beta_2} - \beta_2)\frac{\partial q_1}{\partial \beta_2}.$$

The variance of $\widehat{q_1}$ is,

$$\mathbb{V}(\widehat{q_1}) = \mathbb{E}\left[(\widehat{q_1} - q_1)^2\right].$$

Substituting the first-order expansion,

$$\mathbb{V}(\widehat{q_1}) = \mathbb{V}(\widehat{\beta_0})\left(\frac{\partial q_1}{\partial \beta_0}\right)^2 + \mathbb{V}(\widehat{\beta_2})\left(\frac{\partial q_1}{\partial \beta_2}\right)^2 + 2\text{Cov}(\widehat{\beta_0}, \widehat{\beta_2})\left(\frac{\partial q_1}{\partial \beta_0}\right)\left(\frac{\partial q_1}{\partial \beta_2}\right),$$

where $\mathbb{V}(\widehat{\boldsymbol{\beta}}) = (\mathbf{X}^{\intercal}\widehat{\mathbf{W}}\mathbf{X})^{-1}$, $\widehat{W} = \text{diag}(\widehat{\pi}_i(1-\widehat{\pi}_i))$, and

$$\frac{\partial q_1}{\partial \beta_0} = \frac{\partial}{\partial \beta_0}\left(-\frac{\beta_0}{\beta_2}\right) = -\frac{1}{\beta_2}$$

and

$$\frac{\partial q_1}{\partial \beta_2} = \frac{\partial}{\partial \beta_2}\left(-\frac{\beta_0}{\beta_2}\right) = \frac{\beta_0}{\beta_2^2}.$$

Therefore,

$$\mathbb{V}(\widehat{q_1}) = V(\widehat{\beta_0})\left(-\frac{1}{\beta_2}\right)^2 + V(\widehat{\beta_2})\left(\frac{\beta_0}{\beta_2^2}\right)^2 + 2\text{Cov}(\widehat{\beta_0},\widehat{\beta_2})\left(-\frac{1}{\beta_2}\right)\left(\frac{\beta_0}{\beta_2^2}\right).$$

Second-order Taylor series for q_1,

$$\begin{aligned}\widehat{q_1} \;\approx\; & q_1 + (\widehat{\beta_0}-\beta_0)\frac{\partial q_1}{\partial \beta_2} + (\widehat{\beta_2}-\beta_2)\frac{\partial q_1}{\partial \beta_2} + \frac{1}{2}(\widehat{\beta_0}-\beta_0)^2\frac{\partial^2 q_1}{\partial \beta_0^2} + \frac{1}{2}(\widehat{\beta_2}-\beta_2)^2\frac{\partial^2 q_1}{\partial \beta_2^2} \\ & +(\widehat{\beta_0}-\beta_0)(\widehat{\beta_2}-\beta_2)\frac{\partial^2 q_1}{\partial \beta_0 \partial \beta_2}.\end{aligned}$$

The bias of $\widehat{q_1}$ is,

$$\text{Bias}(\widehat{q_1}) = \mathbb{E}(\widehat{q_1}) - q_1 = \frac{1}{2}\mathbb{V}(\widehat{\beta_0})\frac{\partial^2 q_1}{\partial \beta_0^2} + \frac{1}{2}\mathbb{V}(\widehat{\beta_2})\frac{\partial^2 q_1}{\partial \beta_2^2} + \text{Cov}(\widehat{\beta_0},\widehat{\beta_2})\frac{\partial^2 q_1}{\partial \beta_0 \partial \beta_2},$$

where

$$\frac{\partial^2 q_1}{\partial \beta_0^2} = 0, \quad \frac{\partial^2 q_1}{\partial \beta_2^2} = \frac{2\beta_0}{\beta_2^3}, \quad \text{and} \quad \frac{\partial^2 q_1}{\partial \beta_0 \partial \beta_2} = \frac{1}{\beta_2^2}.$$

Therefore,

$$\text{Bias}(\widehat{q_1}) = \frac{1}{2}V(\widehat{\beta_2})\left(\frac{2\beta_0}{\beta_2^3}\right) + \text{Cov}(\widehat{\beta_0},\widehat{\beta_2})\left(\frac{1}{\beta_2^2}\right).$$

The mean square error (MSE) of $\widehat{q_1}$ is given by,

$$\text{MSE}(\widehat{q_1}) = \mathbb{V}(\widehat{q_1}) + (\text{Bias}(\widehat{q_1}))^2.$$

First-order Taylor series for q_2,

$$\widehat{q_2} \;\approx\; q_2 + (\widehat{\beta_1}-\beta_1)\frac{\partial q_2}{\partial \beta_1} + (\widehat{\beta_2}-\beta_2)\frac{\partial q_2}{\partial \beta_2}.$$

The variance follows from,

$$\begin{aligned}\mathbb{V}(\widehat{q_2}) &= \mathbb{E}\left[\widehat{q_2}-q_2\right]^2 \\ &= \mathbb{V}(\widehat{\beta_1})\left(\frac{\partial q_2}{\partial \beta_1}\right)^2 + \mathbb{V}(\widehat{\beta_2})\left(\frac{\partial q_2}{\partial \beta_2}\right)^2 + 2\text{Cov}(\widehat{\beta_1},\widehat{\beta_2})\left(\frac{\partial q_2}{\partial \beta_1}\right)\left(\frac{\partial q_2}{\partial \beta_2}\right),\end{aligned}$$

where

$$\frac{\partial q_2}{\partial \beta_1} = \frac{\partial}{\partial \beta_1}\left(-\frac{\beta_1}{\beta_2}\right) = -\frac{1}{\beta_2},$$

and

$$\frac{\partial q_2}{\partial \beta_2} = \frac{\partial}{\partial \beta_2}\left(-\frac{\beta_1}{\beta_2}\right) = \frac{\beta_1}{\beta_2^2}.$$

Therefore,

$$\mathbb{V}(\widehat{q_2}) = \mathbb{V}(\widehat{\beta_1})\left(-\frac{1}{\beta_2}\right)^2 + \mathbb{V}(\widehat{\beta_2})\left(\frac{\beta_1}{\beta_2^2}\right)^2 + 2\text{Cov}(\widehat{\beta_1},\widehat{\beta_2})\left(-\frac{1}{\beta_2}\right)\left(\frac{\beta_1}{\beta_2^2}\right).$$

The second-order Taylor expansion of $\widehat{q_2}$ is,

$$\begin{aligned}\widehat{q_2} \quad &\approx \quad q_2 + (\widehat{\beta_1}-\beta_1)\frac{\partial q_2}{\partial \beta_1} + (\widehat{\beta_2}-\beta_2)\frac{\partial q_2}{\partial \beta_2} + \frac{1}{2}(\widehat{\beta_1}-\beta_1)\frac{\partial^2 q_2}{\partial \beta_1^2} + \frac{1}{2}(\widehat{\beta_2}-\beta_2)\frac{\partial^2 q_2}{\partial \beta_2^2} \\ &\quad +(\widehat{\beta_1}-\beta_1)(\widehat{\beta_2}-\beta_2)\frac{\partial^2 q_2}{\partial \beta_1 \partial \beta_2}.\end{aligned}$$

The bias follows as,

$$\text{Bias}(\widehat{q_2}) = E(\widehat{q_2}) - q_2 = \frac{1}{2}V(\widehat{\beta_1})\frac{\partial^2 q_2}{\partial \beta_1^2} + \frac{1}{2}V(\widehat{\beta_2})\frac{\partial^2 q_2}{\partial \beta_2^2} + Cov(\widehat{\beta_1},\widehat{\beta_2})\frac{\partial^2 q_2}{\partial \beta_1 \partial \beta_2}$$

where

$$\frac{\partial^2 q_2}{\partial \beta_1^2} = 0, \quad \frac{\partial^2 q_2}{\partial \beta_2^2} = \frac{2\beta_1}{\beta_2^3}, \quad \text{and} \quad \frac{\partial^2 q_2}{\partial \beta_1 \partial \beta_2} = \frac{1}{\beta_2^2}.$$

Therefore,

$$\text{MSE}(\widehat{q_2}) = \mathbb{V}(\widehat{q_2}) + (\text{Bias}(\widehat{q_2}))^2.$$

In general, the expressions for the variances and biases for $\widehat{c}$ and $\widehat{m}$, along the p-th axis, where $(k = 1, \ldots, p)$, are as follows:

$$\mathbb{V}(\widehat{c}) = \mathbb{V}(\widehat{q_1}) = \mathbb{V}(\widehat{\beta_0})\left(\frac{\partial q_1}{\partial \beta_0}\right)^2 + \mathbb{V}(\widehat{\beta_p})\left(\frac{\partial q_1}{\partial \beta_p}\right)^2 + 2\,\text{Cov}(\widehat{\beta_0},\widehat{\beta_p})\left(\frac{\partial q_1}{\partial \beta_0}\right)\left(\frac{\partial q_1}{\partial \beta_p}\right),$$

and

$$\text{Bias}(\widehat{c}) = \text{Bias}(\widehat{q_1}) = \frac{1}{2}\mathbb{V}(\widehat{\beta_0})\frac{\partial^2 q_1}{\partial \beta_0^2} + \frac{1}{2}\mathbb{V}(\widehat{\beta_p})\frac{\partial^2 q_1}{\partial \beta_p^2} + \text{Cov}(\widehat{\beta_0},\widehat{\beta_p})\frac{\partial^2 q_1}{\partial \beta_0 \partial \beta_p}.$$

$$\mathbb{V}(\widehat{m_k}) = \mathbb{V}(\widehat{q_2}) = \mathbb{V}(\widehat{\beta_k})\left(\frac{\partial q_2}{\partial \beta_k}\right)^2 + \mathbb{V}(\widehat{\beta_p})\left(\frac{\partial q_2}{\partial \beta_p}\right)^2 + 2\text{Cov}(\widehat{\beta_k},\widehat{\beta_p})\left(\frac{\partial q_2}{\partial \beta_k}\right)\left(\frac{\partial q_2}{\partial \beta_p}\right),$$

and

$$\text{Bias}(\widehat{m_k}) = \text{Bias}(\widehat{q_2}) = \frac{1}{2}\mathbb{V}(\widehat{\beta_k})\frac{\partial^2 q_2}{\partial \beta_k^2} + \frac{1}{2}\mathbb{V}(\widehat{\beta_p})\frac{\partial^2 q_2}{\partial \beta_p^2} + \text{Cov}(\widehat{\beta_k},\widehat{\beta_p})\frac{\partial^2 q_2}{\partial \beta_k \partial \beta_p}.$$

Note that q_1 and q_2 are $-\dfrac{\hat{\beta}_0}{\hat{\beta}_p}$ and $-\dfrac{\hat{\beta}_k}{\hat{\beta}_p}$, respectively.

REFERENCES

1. N.A. Butler, *Optimal and orthogonal Latin hypercube designs for computer experiments*, Biometrika, vol. 88, no. 3, pp. 847–857, 2001.
2. K.T. Fang, D.K.J. Lin, P. Winkler, and Y. Zhang, *Uniform design: Theory and Application*, Technometrics, vol. 42, no. 3, pp. 237–248, 2000.
3. A. Kassambara, and F. Mundt, *factoextra: Extract and Visualize the Results of Multivariate Data Analyses*, R package version 1.0.7, https://CRAN.R-project.org/package=factoextra, 2020.

4. D.M. Steinberg, and D.K.J. Lin, *A construction method for orthogonal Latin hypercube designs*, Biometrika, vol. 93, no. 2, pp. 279–288, 2006.
5. K.Q. Ye, *Orthogonal column Latin hypercubes and their application in computer experiments*, Journal of the American Statistical Association, vol. 93, no. 444, pp. 1430–1439, 1998.
6. M.D. Mckay, R.J. Beckman, and W.J. Conover, *A comparison of three methods for selecting values of input variables in the analysis of output from a computer code*, Technometrics, vol. 21, pp. 239–245, 1979.
7. J.J. Faraway, *Extending the linear model, with R*, CRC Press, Boca Raton, 2016.
8. P. Li, J. Bu, C. Chen, and D. Cai, *Manifold optimal experimental design via dependence maximization for active learning*, Neurocomputing, vol. 142, pp. 429–437, 2014.
9. J. Wang, and E Park, *Active learning for penalized logistic regression via sequential experimental design*, Neurocomputing, vol. 222, pp. 183–190, 2017.
10. A. Atkinson, A. Donev, and R. Tobias, *Optimum experimental designs, with SAS*, OUP Oxford, vol. 34, 2007.
11. A. Wang, *Active learning: An exploratory study of its application in R*, Unpublished, 2021.
12. B. Settles, *Active learning literature survey*, University of Wisconsin-Madison Department of Computer Sciences, 2009.
13. T. He, S. Zhang, J. Xin, P. Zhao, J. Wu, X. Xian, C. Li, and Z. Cui, *An active learning approach with uncertainty, representativeness, and diversity*, The Scientific World Journal, Hindawi, vol. 2014, 2014.
14. A.J. Joshi, F. Porikli, and N. Papanikolopoulos, *Multi-class active learning for image classification*, 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2372–2379, 2009.
15. W.J. Conover, *On a better method of selecting values of input variables for computer codes*, Unpublished manuscript, 1975.
16. A. Diop, A. Diop and J.F. Dupuy, *Maximum likelihood estimation in the logistic regression model with a cure fraction*, Electronic Journal of Statistics, Institute of Mathematical Statistics and Bernoulli Society, vol. 5, pp. 460–483, 2011.
17. R.A. Fisher, *The use of multiple measurements in taxonomic problems*, Annals of eugenics, Wiley Online Library, vol. 7, no. 2, pp. 179–188, 1936.
18. D.W. Hosner, and S. Lemeshow, *Applied logistic regression*, New York: John Wiley & Son, vol. 581, 1989.
19. J. Gower, S. Lubbe, and N. Le Roux, *Understanding Biplots*, John Wiley & Sons, Chichester, 2011.
20. D.D. Lewis, *A sequential algorithm for training text classifiers: Corrigendum and additional data*, Acm Sigir Forum, ACM New York, NY, USA, vol. 29, no. 2, pp. 13–19, 1995.
21. D.D. Lewis, and J. Catlett, *Heterogeneous uncertainty sampling for supervised learning*, Machine learning proceedings 1994, Elsevier, San Francisco, pp. 148–156, 1994.
22. J.N. Hwang, J.J. Choi, S. Oh, R.J. Marks, and others, *Query-based learning applied to partially trained multilayer perceptrons*, IEEE Transactions on Neural Networks, vol. 2, no. 1, pp. 131–136, 1991.
23. E.B. Baum, *Neural net algorithms that learn in polynomial time from examples and queries*, IEEE Transactions on Neural Networks, vol. 2, no. 1, pp. 5–19, 1991.
24. A. McCallum, K. Nigam, and others, *Employing EM and Pool-Based Active Learning for Text Classification*, ICML, Citeseer, vol. 98, pp. 350–358, 1998.
25. H. Chai, Y. Liang, S. Wang, and H.W. Shen, *A novel logistic regression model combining semi-supervised learning and active learning for disease classification*, Scientific Reports, vol. 8, 2018.
26. Y. Yazhou, and M. Loog, *A benchmark and comparison of active learning for logistic regression*, Pattern Recognition, vol. 83, pp. 401–415, 2018.
27. A.I. Schein, and L.H. Ungar, *Active learning for logistic regression: an evaluation*, 2007.
28. B. Minasny, and A.B. McBratney, *A conditioned Latin hypercube method for sampling in the presence of ancillary information*, Computers & geosciences, Elsevier, vol. 32, no. 9, pp. 1378–1388, 2006.
29. M.D. McKay, and R.J.C. Beckman, *WJ, A comparison of three methods for selecting values of input variables in the analysis of output from a computer CodeTechnometrics*, Am Stat Assoc Am Soc Qual, vol. 21, pp. 239–245, 1979.
30. M.D. McKay, R.J. Beckman, and W.J. Conover, *A comparison of three methods for selecting values of input variables in the analysis of output from a computer code*, Technometrics, Taylor & Francis, vol. 42, no. 1, pp. 55–61, 2000.
31. Jr. F.E. Harrell and K.L. Lee, and D.B. Mark, *Multivariable prognostic models: issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors*, Statistics in medicine, Wiley Online Library, vol. 15, no. 4, pp. 361–387, 1996.
32. H.S. Seung, M. Opper, and H. Sompolinsky, *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 287–294, 1992.
33. Y. Guo, and R. Greiner, *Optimistic active-learning using mutual information*, IJCAI, vol. 7, pp. 823–829, 2007.
34. A. Holub, P. Perona, and M.C. Burl, *Entropy-based active learning for object recognition*, 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, IEEE. pp. 1–8, 2008.
35. N. Roy, and A. McCallum, *Toward optimal active learning through Monte Carlo estimation of error reduction*, ICML, Williamstown, vol. 2, pp. 441–448, 2001.
36. K. Yu, J. Bi, and V. Tresp, *Active learning via transductive experimental design*, Proceedings of the 23rd International Conference on Machine learning, pp. 1081–1088, 2006.
37. Y. Yang, and M. Loog, *A variance maximization criterion for active learning*, Pattern Recognition, Elsevier, vol. 78, pp. 358–790, 2018.
38. A. Freytag, E. Rodner, and J. Denzler, *Selecting influential examples: Active learning with expected model output changes*, Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV 13. Springer, pp. 562–577, 2014.
39. S.J. Huang, R. Jin, and Z.H. Zhou, *Active learning by querying informative and representative examples*, Advances in neural information processing systems, vol. 23, 2010.
40. D.C. Montgomery, *Design and analysis of experiments*, John Wiley and Sons, 2017.
41. V.L. Nguyen, M.H. Shaker, and E. Hüllermeier, *How to measure uncertainty in uncertainty sampling for active learning*, Machine Learning, Springer, vol. 111, no. 1, pp. 89–122, 2022.

42. Y. Fu, X. Zhu, and B. Lin, *A survey on instance selection for active learning*, Knowledge and information systems, Springer, vol. 35, no. 1, pp. 249–283, 2013.
43. Y. Yang, and M. Loog, *To Actively Initialize Active Learning*, Pattern Recognition, Elsevier, vol. 131, 2022.
44. B. Settles, and M. Craven, *An analysis of active learning strategies for sequence labeling tasks*, Proceedings of the 2008 conference on empirical methods in natural language processing, pp. 1070–1079, 2008.
45. P. Kumar, and A. Gupta, *Active learning query strategies for classification, regression, and clustering: a survey*, Journal of Computer Science and Technology, Springer, vol. 35, pp. 913–945, 2020.
46. A.I. Schein, *Active learning for logistic regression*, University of Pennsylvania, Ph.D. thesis, 2005.
47. O. Reyes, and S. Ventura, *Evolutionary strategy to perform batch-mode active learning on multi-label data*, ACM Transactions on Intelligent Systems and Technology (TIST), ACM New York, NY, USA, vol. 9, no. 4, pp. 1–26, 2018.
48. S. Tong, and D. Koller, *Support vector machine active learning with applications to text classification*, Journal of machine learning research, vol. 2, no. Nov, pp. 45–66, 2001.
49. Y. Yang, *Towards practical active learning for classification*, Delft University of Technology, Ph.D thesis, 2018.
50. S.E. Argamon, and I. Dagan, *Committee-based sample selection for probabilistic classifiers*, Journal of Artificial Intelligence Research, vol. 11, pp. 335–360, 1999.
51. T. Scheffer, C. Decomain, and S. Wrobel, *Active hidden Markov models for information extraction*, Advances in Intelligent Data Analysis: 4th International Conference, IDA 2001 Cascais, Portugal, September 13–15, 2001 Proceedings 4. Springer Berlin Heidelberg, pp. 309–318, 2001.
52. D. Lewis, and W. Gale, *A sequential algorithm for training text classifiers*, SIGIR'94: Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, organised by Dublin City University. pp. 3–12, 1994.
53. J. Lu, P. Zhao, and S.C.H. Hoi, *Online passive-aggressive active learning*, Machine Learning, Springer, vol. 103, pp. 141–183, 2016.
54. M.F. Balcan, A. Broder, and T. Zhang, *Margin-Based Active Learning*, Springer Berlin Heidelberg, pp. 35–50, 2007.
55. D. Roth, and K. Small, *Margin-Based Active Learning for Structured Output Spaces*, Springer Berlin Heidelberg, pp. 413–424, 2006.
56. M. Stephanou, and M. Varughese, *hermiter: R package for sequential nonparametric estimation*, Computational Statistics, Springer, pp. 1–37, 2023.
57. X. Guo, Y. Yin, C. Dong, G. Yang, and G. Zhou, *On the Class Imbalance Problem*, 2008 Fourth International Conference on Natural Computation. vol. 4, pp. 192–201, 2008.
58. C. Marzban, *The ROC curve and the area under it as performance measures*, Weather and Forecasting, American Meteorological Society, vol. 19, no. 6, pp. 1106–1114, 2004.
59. D.J. Brus, *Sampling for digital soil mapping: A tutorial supported by R scripts*, Geoderma, Elsevier, vol. 338, pp. 464–480, 2019.
60. A.B. Owen, *Orthogonal arrays for computer experiments, integration and visualization*, Statistica Sinica, JSTOR, pp. 439–452, 1992.
61. R. Carnell, *Package 'lhs'*, CRAN. https://cran.rproject.org/web/packages/lhs/lhs.pdf. 2016.
62. F.A.C. Viana, *Things you wanted to know about the Latin hypercube design and were afraid to ask*, 10th World Congress on Structural and Multidisciplinary Optimization. sn, vol. 19, no. 24.05, pp. 1–9, 2013.
63. J.C. Helton, and F.J. Davis, *Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems*, Reliability Engineering & System Safety, Elsevier, vol. 81, no. 1, pp. 23–69, 2003.
64. M. Stephanou, M. Varughese, and I. Macdonald, *Sequential quantiles via Hermite series density estimation*, 2017.
65. P. Roudier, C. Brugnard, D. Beaudette, and B. Louis, *Package 'clhs'*, Citeseer, 2019.
66. A.I. Schein, and L.H. Ungar, *Active learning for logistic regression: an evaluation*, Machine Learning, Springer, vol. 68, pp. 265–9, 2007.
67. C. Ferri, J. Hernández-Orallo, and R. Modroiu, *An experimental comparison of performance measures for classification*, Pattern recognition letters, Elsevier, vol. 30, no. 1, pp. 27–38, 2009.
68. D. Ballabio, F. Grisoni, and R. Todeschini, *Multivariate comparison of classification performance measures*, Chemometrics and Intelligent Laboratory Systems, Elsevier, vol. 174, pp. 33–44, 2018.
69. M. Sokolova, and G. Lapalme, *A systematic analysis of performance measures for classification tasks*, Information processing & management, Elsevier, vol. 45, no. 4, pp. 427–437, 2009.
70. G.E.P. Box, and J.S. Hunter, *The 2 k—p fractional factorial designs*, Technometrics, Taylor & Francis, vol. 3, no. 3, pp. 311–351, 1961.
71. G.E.P. Box, and J.S. Hunter, *The 2 k—p Fractional Factorial Designs Part II*, Technometrics, Taylor & Francis, vol. 3, no. 4, pp. 449–458, 1961.
72. I.M. Sobol, *On the distribution of points in a cube and the approximate evaluation of integrals*, Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki, Russian Academy of Sciences, Branch of Mathematical Sciences. vol. 7, no. 4, pp. 784–802, 1967.
73. R. Carnell, *lhs: Latin Hypercube Samples*, R package version 1.2.0. https://CRAN.R-project.org/package=lhs 2024.
74. P. Roudier, *clhs: a R package for conditioned Latin hypercube sampling*, 2011.
75. W.N. Venables, and B.D. Ripley, *Modern Applied Statistics with S*, Springer. https://www.stats.ox.ac.uk/pub/MASS4/. 2002.
76. M. Stephanou, and M. Varughese, *hermiter: R package for sequential nonparametric estimation*, Computational Statistics. https://doi.org/10.1007/s00180-023-01382-0. 2023.
77. M.W. Al-Nabki, E. Fidalgo, E. Alegre, S.J. Delany, and F. Janez-Martino, *Classifying the content of online notepad services using active learning*, Journal of Intelligent Information Systems, Springer, vol. 63, no. 2, pp. 507–533, 2025.