Low Query Budget Active Learning for Classification and Regression

Bjarne Jaster[0000−0002−8362−5369](✉), Alaa Tharwat[0000−0003−4204−4506], Eiram Mahera Sheikh[0009−0000−2391−8330], Martin Kohlhase[0009−0002−9374−0720], and Wolfram Schenck[0000−0003−3300−2048]

Center for Applied Data Science (CfADS), HSBI - Bielefeld University of Applied Sciences and Arts, Bielefeld, Germany
{bjarne.jaster, alaa.othman, eiram_mahera.sheikh, martin.kohlhase, wolfram.schenck}@hsbi.de

Abstract. The labeling process for supervised learning is costly and time-consuming, and is often impractical to scale due to real-world constraints. Active learning (AL) addresses this challenge by strategically selecting representative and informative data points to reduce labeling efforts. This paper focuses on an AL scenario in which only a very limited number of labels can be acquired. We propose an algorithm operating in two phases: (1) an exploration phase that prioritizes representative and diverse data points using density-driven criteria, and (2) an exploitation phase that combines predictive uncertainty with density weighting to select informative samples from densely populated regions. This enhances both representativeness and informativeness. Our results demonstrate significant improvements in model quality compared to other algorithms typically employed for this scenario, across various scenarios involving imbalanced data in classification tasks and skewness in regression tasks. Through this work, we aim to provide a new algorithm for this scenario and investigate general principles for AL. While most AL studies focus on either classification or regression, our work applies the algorithms to both. Therefore, we can analyze the differences between classification and regression problems and their effects on AL strategies. Furthermore, we explore different categories of AL criteria and their effectiveness in the low-budget regime. These results also provide insight into the cold-start problem, which involves selecting an initial labeled set and is faced by many model-based AL methods.

Keywords: Active Learning · Classification · Regression · Low Query Budget.

1 Introduction

In the field of machine learning (ML), there are large repositories of unlabeled data that hold a considerable potential. However, to unlock this potential and use them for ML labeling is necessary. The labeling process of these datasets often

requires human effort, time, and resources. This bottleneck hinders the development of ML models in many scenarios. The challenge is therefore to maximize model performance with minimal labeling effort, a problem that active learning (AL) attempts to address by intelligently selecting the most informative data points for annotation [8,22]. The core principle of AL involves an iterative process in which a learning algorithm actively selects the most informative unlabeled data points from a larger pool and queries an oracle - often a human expert - to obtain the labels. These newly labeled samples are then used to train a model, which guides the learning process in a way that improves model accuracy (e.g. by providing uncertainty estimates). This strategic selection reduces labeling costs and accelerates training, making AL a valuable tool in many domains [8].

AL is generally applicable to both classification and regression tasks, with three typical application scenarios: pool-based sampling, where the learner selects points from a large pool of unlabeled data; stream-based selective sampling, where the decision to query a label must be made without knowing which points will arrive later in the data stream; and membership query synthesis, where the learner generates novel points for labeling [16]. This work focuses on pool-based sampling.

To select the best data points, each AL strategy has its own heuristics to determine the importance of all available unlabeled points. These heuristics are typically categorized into two general approaches: Exploration and Exploitation [22]. Exploration-based AL methods aim to select representative or diverse data points that represent the data distribution. Exploitation-based methods select data points using an ML model trained on the previously selected data points. They use the information provided by the model (e.g., the uncertainty of the predictions) to assign a value indicating the informativeness of the data points. Because of their reliance on the ML model, exploitation-based AL methods are also referred to as model-based AL methods, whereas exploration-based methods are termed model-free [6].

Most studies on AL focus on scenarios where a reasonable amount of data points are queried. In Deep AL, thousands of points are queried (e.g. [19]) and even in works that cover small datasets that have less than 50 dimensions and at most a few thousand data points, the query budget often comprises around 100 data points (e.g. [5,6,13,15]). Only very few studies deal with AL scenarios where an extremely low budget is present. One study investigating this topic in the context of Deep AL provides theoretical and empirical evidence that including model-based criteria such as uncertainty in the low-budget regime of the AL process is ineffective [3]. They claim that this phenomenon occurs because model-based criteria depend on ML models that are of low quality due to limited training data. This study focuses on Deep AL and therefore the authors consider a few hundred points to still be in the low-budget regime. Going away from Deep Learning and Big Data these budgets may not be realistic anymore. For modelling very expensive and/or time-consuming processes, potentially only 15 or 20 data points can be selected. One example of such a process is the curing of concrete, which depends on many factors, including storage conditions, and

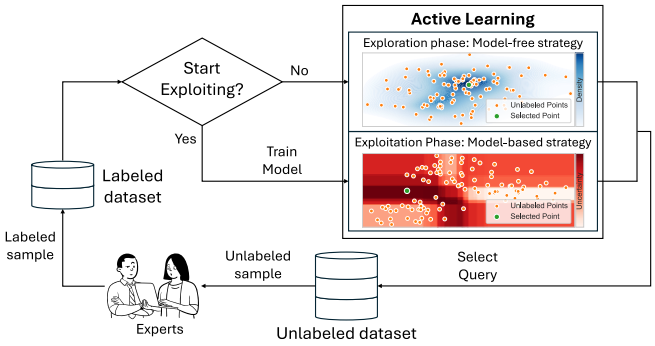

Fig. 1. Overview of the proposed AL strategy for the small data, low-budget regime.

can take up to a month [12]. A company would most likely not be able to afford to conduct a hundred experiments for this process, as it would take years and require significant storage capacity. Another example is the growth of plants. Some plants can only be harvested once a year, so again, very few experiments are realistic [2].

For these small data, low query budget scenarios, the question remains as to whether the observations made in [3] for deep AL are still valid. To investigate this question, we propose a novel AL method. This method consists of two phases: an exploration phase, which is model-free, and an exploitation phase, which incorporates model-based information into the selection process (see Fig. 1). By controlling the point at which the phase transition occurs in the AL process, we can investigate the usefulness of a model-based criterion.

In summary, the main contributions of this work are as follows: We propose and evaluate this two-phase AL strategy, specifically tailored to the small data, low-budget regime. This method is compared to several others that are typically used to initialize AL methods (see, e.g., [5,6,13]) and are therefore also used in the low-budget regime. Based on the results of this comparison, we aim to address the following research questions (RQs) in both the classification and regression settings. This will enable us to provide further insights into the differences between the two settings, a research area that has not been investigated much.

RQ1: Does the inclusion of model-based AL criteria help in small data, low-budget scenarios? If yes, at which point should the inclusion happen?
RQ2: Which types of AL criteria (density-, cluster-, distance-, informativeness-based) are generally important to consider for a high-quality, low-budget AL method?

The rest of the paper is organized as follows: Section 2 reviews related work, comparing exploration, exploitation, and hybrid approaches. Section 3 details the proposed model. Section 4 presents the experimental analysis conducted to evaluate the performance of the proposed method. Finally, Section 5 discusses

the findings, answers the RQs and outlines directions for future work. The code of our method and the experiments can be found on `https://github.com/bjaster/Low-Query-Budget-AL`.

2 Related Work

AL is a *partially-supervised* approach that combines labeled ($\mathcal{L}$) and unlabeled data ($\mathcal{U}$). It uses an iterative query strategy to find new points $\mathbf{x}^*$ for labeling, as follows:

$$\mathbf{x}^* = \underset{\mathbf{x} \in \mathcal{U}}{\operatorname{argmax}}\, \phi(\mathbf{x}, h) \tag{1}$$

Here, $\phi(\cdot)$ quantifies the informativeness of an unlabeled instance $\mathbf{x}$ relative to the current model h [23]. In this section, we will discuss how other studies handle AL in the low-budget regime. Although few studies address this issue directly, there are related challenges for which similar approaches are used.

One of these challenges is called the *cold start* or *initialization problem*, where no labeled data is initially available to train an ML-model to guide a model-based AL process. Since model-based AL methods depend heavily on the quality of the initial dataset, providing this dataset is critical, as an inaccurate initial model understanding may negatively affect the selection process [1]. For this reason, we first examine how the existing model-based AL literature approaches the creation of the initial labeled set. While this is not the primary focus of these works, their strategies offer valuable insights. We then provide a concise overview of common model-based AL strategies.

2.1 Exploration-Based Approaches

Most works on AL use a random selection to select the initial dataset (e.g. [15]). However, there are a few works that use model-free or exploration-based AL methods. The aim of these methods is to select samples that represent the data distribution and are diverse from each other, thus ensuring efficient coverage of the input space. In [5], Greedy Sampling (GSx) [26] is used for initialization to provide a deterministic and diverse initial dataset. In [6], GSx and iterative Representativeness and Diversity Maximization (iRDM) [10] are used for providing the initial labeled set. These two methods are examples for the two main paradigms that are used in model-free AL. While iRDM mainly focuses on representativeness, GSx maximizes diversity.

Representativeness-based AL methods select samples that reflect the overall data distribution, with the goal of selecting a subset that is similar to the whole dataset [27]. A common strategy to achieve representative samples is cluster-based sampling. Classical algorithms such as k-means clustering [11] have inspired several AL methods that use cluster centroids or structural information to guide sample selection [4,25]. Two notable examples, also used later in this work, are the already mentioned iRDM and TypiClust [3]. The iRDM algorithm [10] first clusters the unlabeled pool and selects samples closest to each cluster

center. It then optimizes this selection to avoid redundant queries (more details in Section 3). TypiClust is presented in the same work that investigates low-budget AL (see chapter 1) as a method specifically suited to this scenario [3]. It emphasizes coverage by targeting clusters that remain unrepresented in the labeled set. It selects the most "typical" point within each such cluster, where typicality is defined as the inverse of the average distance to its k nearest neighbors.

Diversity-based methods aim to maximize the distance between the selected points. GSx [26] starts by selecting the point closest to the centroid of the entire unlabeled pool. In subsequent rounds, it maximizes diversity by choosing the point furthest from all previously chosen samples in the input space.

2.2 Exploitation-Based Approaches

Exploitation-based approaches rely on the predictions of the current model to identify samples that are likely to improve performance when labeled. These techniques aim to fine-tune decision boundaries by focusing on regions that are either uncertain or informative. One classical approach is Query by Committee (QBC) [17], where multiple models are trained, and instances with the highest disagreement among the models are selected. Another well-known strategy is uncertainty sampling, where the the model selects samples about which it is least confident [9,18]. A more model-driven approach is expected error reduction [14,28], which estimates the reduction in generalization error that would result from labeling a candidate sample.

2.3 Hybrid Approaches

Pure exploration or exploitation may be sub-optimal. An overemphasis on exploration can lead to excessive sampling from uninformative regions. Conversely, excessive exploitation can result in to a high concentration of data points that provide only limited knowledge. This can lead to poor generalization and missed discovery of new patterns or classes. As a result, many recent approaches use hybrid strategies that dynamically balance exploration and exploitation. These techniques either begin with exploration to broadly characterize the input space and shift toward exploitation as the model becomes more reliable [21,24] or consider both exploration and exploitation criteria at all times by e.g. adding them [6,15]. For example, in [24], the authors' geometrical methods varied with each phase. First, PCA-inspired exploration investigated regions of high variance. Then, LDA-inspired exploitation focused on the boundary points between classes.

Building on this line of research, our work proposes a novel AL method for low-budget scenarios that explicitly balances exploration and exploitation. Our strategy first leverages the structural properties of the input space to later enable a model to identify regions of high uncertainty.

3 Proposed Method

We begin with the assumption that all available data are unlabeled, denoted as $\mathcal{U} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{n_u}\}$, where $\mathbf{x}_i \in \mathcal{U}$ represents an unlabeled data point and n_u is the total number of unlabeled points. The labeled dataset $\mathcal{L}$ is initially empty ($\mathcal{L} = \emptyset$). The budget B defines the number of data points that can be selected. Our method, called *Low Query Budget AL (LQBAL)*, comprises a two-phase approach:

1. **Initialization**: Our method selects m representative points from the unlabeled pool $\mathcal{U}$ using one of two strategies: **Cluster-based** or **Median-based** selection methods.
2. **Two-Phase Querying**: After initialization, a dynamic exploration-exploitation strategy queries points until the budget B is exhausted. The mode switches from exploration to exploitation depending on the hyperparameter $s \in [0, 1]$. This defines which fraction of the budget should be spent on each strategy, where $s = 0$ is pure exploitation after initialization, while $s = 1$ is pure exploration.
   - **Exploration** ($|\mathcal{L}|/B \leq s$): Focuses on dense regions and uses distance metrics to minimize redundancy.
   - **Exploitation** ($|\mathcal{L}|/B > s$): Focuses on dense and uncertain regions.

With this two-phase hybrid approach LQBAL balances broad data coverage with targeted refinement of model performance. At each iteration, the queried point is labeled, transferred from $\mathcal{U}$ to $\mathcal{L}$. When the exploitation mode is active, the model is retrained on the updated $\mathcal{L}$. The process terminates when $|\mathcal{L}| = B$, returning $\mathcal{L}$ and $\mathcal{U}$.

3.1 Initial Selection

LQBAL comprises two options to select the initial set of points. This initialization is necessary for the case $s = 1$, so that a model can be trained. The median-based method ensures representativeness and robustness to outliers [24], while the cluster-based maximizes the diversity and representativeness of the selected data [10].

Median-based initialization: The algorithm first removes outliers from $\mathbf{x}_i \in \mathcal{U}$ using the interquartile range (IQR) method:

$$\text{Lower Bound} = Q_1 - 1.5 \cdot \text{IQR}, \quad \text{Upper Bound} = Q_3 + 1.5 \cdot \text{IQR}, \tag{2}$$

where $\text{IQR} = Q_3 - Q_1$. The cleaned set $\mathcal{U}_c$ is then reduced via PCA [20] to retain components explaining at least τ variance (e.g., $\tau = 0.9$), yielding $\mathcal{U}_{c,r}$. In this space, the median, Q_1, and Q_3 are computed for each dimension and the nearest point is selected:

$$x_j^* = \underset{\mathbf{x}_i \in \mathcal{U}_{c,r}}{\operatorname{argmin}} \|\mathbf{x}_i - \mathbf{z}_j\|_2, \quad \mathbf{z}_j \in \{\mathbf{x}_{median}, \mathbf{x}_{Q_1}, \mathbf{x}_{Q_3}\}. \tag{3}$$

These three points form the initial labeled set (see Fig. 2 left), covering central and quartile regions while avoiding outliers.

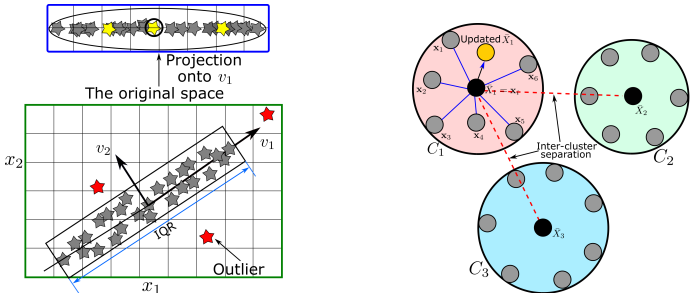

Fig. 2. Illustration of the two initialization methods used in our model. **(Left):** The median-based method removes outliers (red stars), projects the data onto the first PCA components, and queries the median, Q_1, and Q_3. **(Right):** The cluster-based method iteratively finds points that minimize intra-class density while maximizing inter-class separation. The black points represent the initial centroids; the blue point indicates the updated $\bar{X}_1$, while $\bar{X}_2$ and $\bar{X}_3$ remain fixed.

***Cluster-based initialization**:* The iRDM strategy [10] balances two objectives: *representativeness* (selecting samples near cluster centroids to avoid outliers) and *diversity* (selecting from different clusters to cover the input space). First, K-means clustering is applied to the unlabeled set $\mathcal{U}$ with m clusters (default $m = 3$), yielding centroids $\{\bar{X}_1, \ldots, \bar{X}_m\}$. From each cluster C_j, the point closest to $\bar{X}_j$ is chosen: This selection is then iteratively refined ($c_{\max} = 5$ iterations) using *Intra-cluster Density* ($R(\mathbf{x}_i)$, illustrated by the blue lines in FIg. 2 right), which is defined as the mean distance to other points in the same cluster (lower is more representative) and *Inter-cluster Separation* ($\delta(\mathbf{x}_i)$, dashed lines in Fig. 2), which is the distance to the nearest centroid of another cluster (higher is more diverse). The optimal point per cluster maximizes

$$\mathbf{x}_j^* = \underset{\mathbf{x}_i \in \mathcal{U}_{C_j}}{\operatorname{argmax}} \left(\delta(\mathbf{x}_i) - R(\mathbf{x}_i)\right) \tag{4}$$

Refining stops when the selected candidates do not change or $c_{\max}$ is reached. The resulting labeled set $\mathcal{L}$ balances representativeness and diversity, though it may still include noisy points.

3.2 Exploration Phase

This phase aims to provide a representative and diverse dataset to minimize the potentially negative effects of the strategy used to select the initial m points. The points are selected from high-density areas of the unlabeled data to ensure the selection is representative. For each unlabeled point $x_i \in \mathcal{U}$, the average distance to its k-nearest neighbors (default $k = 10$) is computed (similar to [3]):

$$\text{AvgDist}(\mathbf{x}_i) = \frac{1}{k} \sum_{\mathbf{x}_j \in \mathcal{N}_k(\mathbf{x}_i)} \|\mathbf{x}_i - \mathbf{x}_j\|_2, \tag{5}$$

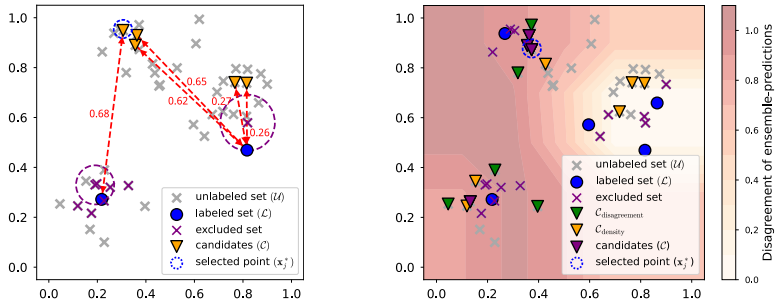

Fig. 3. (Left) Exploration phase strategy: The $t = 5$ most dense points are selected as candidates (orange triangles). The point with the highest distance (red dashed arrows) to already labeled points (blue dots) is chosen for labeling (blue dashed circle). Purple crosses indicate unlabeled data points that are excluded from the selection, because they have labeled points nearer than their AvgDist. For two excluded points the spheres with radius AvgDist are shown (purple dashed circles). **(Right)** Exploitation phase strategy: The intersection (purple triangles) between the points that are most dense (orange triangles) and the points with the highest disagreement (green triangles) represents the candidates (because the intersection was empty for $t = 5$ the $t = 10$ most dense and disagreed points are considered). From those the one with the highest nearest neighbor distance to the labeled points is selected (blue dashed circle).

where $\mathcal{N}_k(\mathbf{x}_i)$ represents the set of the nearest k neighbors to the data point $\mathbf{x}_i$. To avoid redundant queries we exclude all unlabeled points $\mathbf{x}_i$ that have labeled points within a hypersphere with radius AvgDist($\mathbf{x}_i$) from further consideration (see purple circles in Fig. 3 (left) for two examples). The top t (default $t = 5$) points with the lowest AvgDist(x_i) form the candidate set $\mathcal{C}$. From those candidates, the point with the largest nearest neighbor distance to the labeled set $\mathcal{L}$ is queried (see Fig. 3 (left) red arrows):

$$\mathbf{x}_j^* = \underset{\mathbf{x}_i \in \mathcal{C}}{\operatorname{argmax}} \min_{\mathbf{x}_k \in \mathcal{L}} \|\mathbf{x}_i - \mathbf{x}_k\|_2 \tag{6}$$

By excluding points from dense regions that have already been explored and creating a candidate set based on maximum distance to labeled points, we ensure representativeness and diversity for a high-quality dataset that can be used in the exploitation phase.

3.3 Exploitation Phase

In this phase, the strategy includes a model-based exploitation criterion. For that we calculate the disagreement of a committee which is trained on $\mathcal{L}$ (corresponds to the QBC-criterion). The disagreement $\sigma(x_i)$ of the predictions for an unlabeled point is calculated as the standard deviation (regression) or the vote-entropy (classification) of the ensemble-predictions. However, rather than selecting points based solely on disagreement, we also consider the density of the

points (Eq. 5) and exclude points near labeled points, as previously explained for the exploration phase. We identify the top t points (initially $t = 5$) with the highest disagreement and the top t points with the highest density. These candidate sets are denoted as $\mathcal{C}_{\text{disagreement}}$ and $\mathcal{C}_{\text{density}}$, respectively. The query selection is then done on the intersection of those candidate sets, which allows to select the points that are from both dense and uncertain regions:

$$\mathcal{C} = \mathcal{C}_{\text{disagreement}} \cap \mathcal{C}_{\text{density}}. \tag{7}$$

Out of $\mathcal{C}$, the point with the largest nearest neighbor distance to the labeled set $\mathcal{L}$ is queried (see Eq. 6). If $\mathcal{C} = \emptyset$, t is incremented by 5 and Eq. 7 and 6 are computed repeatedly until $\mathcal{C}$ is no longer empty and a selection can be made.

This strategy ensures that the selected points are not only informative, but also representative and diverse, enhancing the quality of the results.

4 Experimental Analysis

This section presents the results that help answer the RQs and evaluate the quality of our proposed method LQBAL. For that, we first perform a small hyperparameter study to evaluate which switch point and initialization method is best for classification and regression, respectively (RQ1). We then present the comparison of our method with other AL methods typically used for initial dataset creation (RQ2).

4.1 Experimental Settings

The comparison is performed on six regression and classification datasets. These datasets are mostly taken from the UCI repository[1] [7] with the exception of the PM10[2], Boston Housing[3] and Fish[4] datasets. The datasets are mostly small datasets with less than 1000 samples and less than 10 features, with a few exceptions like Pendigit and the Superconductivity which have a few thousand instances and up to 81 features. All classification datasets are multiclass datasets with a minimum of three classes and a maximum of ten classes. While the Iris, Pendigit and Segmentation datasets are balanced with roughly the same amount of points per class, the Fish and Glass datasets are imbalanced with a class-ratios of $[56, 34, 20, 17, 14, 11, 6]$ and $[76, 70, 29, 17, 13, 9]$ respectively. A detailed overview of the datasets is given in Table 1. These datasets were chosen because they are diverse in their number of features, classes and instances and allow to investigate the quality of different AL methods in different small data settings and to identify strength and weaknesses of each method.

[1] original names (if not already used): Slump Strength → Concrete Slump Test, Superconductivity → Superconductivity Data, Wheat → Seeds, Segmentation → Image Segmentation, Pendigit → Pen-Based Recognition of Handwritten Digits, Glass → Glass Identification

[2] https://lib.stat.cmu.edu/datasets/PM10.dat

[3] https://lib.stat.cmu.edu/datasets/boston

[4] https://www.utstat.utoronto.ca/brunner/data/legal/fish.txt

Table 1. Characteristics of third-party datasets

Classification				Regression		
Dataset	Classes	Feats.	Size	Dataset	Feats.	Size
Fish	7	6	159	Airfoil Self-Noise	5	1,503
Glass	6	9	214	Boston Housing	13	452
Iris	3	4	150	Real Estate Valuation	6	414
Pendigit	10	16	10,992	Slump Strength	7	103
Segmentation	7	19	2,310	Superconductivity	81	21,263
Wheat	3	7	210	Yacht Hydrodynamics	6	308

A total of 100 independent runs of every AL method are performed on each dataset to obtain statistically meaningful results. First, for each run, the dataset is randomly split into an AL and test set, each containing 50% of the dataset. Importantly, this split is kept identical across all active learners, ensuring that each method operates under the same conditions for a fair comparison. The AL set represents the unlabeled pool from which the active learners can query data points. The test set is used for independent evaluation of the ML models trained on the queried points. Each active learner is allowed to query 15 data points. After the AL process, we evaluate the selection by training a Random Forest (RF) on the selected data points and computing the score (macro F1-score for classification, R2-score for regression) on the test set. The hyperparameters of the RF are set to default (scikit-learn) except for the ensemble size, which is increased to 200.

For the main benchmark of the classification setting we also investigate how many classes were covered by the active learner. For that we compute the percentage of classes covered after the selection of 15 data points and average it over the 100 runs. We investigate this metric since it is an important attribute of an AL method to find all classes in the dataset. While low-budget AL may not the right choice for datasets with more than five classes, we still wanted to provide this information to show the strengths and weaknesses of different AL methods.

4.2 Experimental Results

Hyperparameter study: To determine the effect of incorporating a model-based AL criterion into low-budget AL, we evaluate the optimal value of the switch point hyperparameter s. Additionally, we compare the two presented initialization strategies, cluster-based and median-based. The classification results (see Table 2) indicate that the cluster-based initialization strategy consistently outperformed the median-based strategy, achieving peak average performance at $s = 0.0$ and $s = 0.3$. Generally, one can observe that performance decreases as s increases, showing that a brief exploration is sufficient for classification and that exploitation is more important. To enable at least a little exploration, we choose cluster-based initialization with $s = 0.3$ for our main benchmark experiments.

Table 2. Hyperparameter study for different initialization strategies and switch points s (classification). Results are given as macro F1-score.

Init. strategy	Cluster-based					Median-based				
Switch point (s)	0.0	0.3	0.5	0.7	1.0	0.0	0.3	0.5	0.7	1.0
Fish	**0.594**	0.583	0.551	0.535	0.502	0.590	0.586	0.566	0.558	0.523
Glass	0.398	0.404	**0.431**	0.394	0.323	0.402	0.418	0.413	0.386	0.328
Iris	0.944	0.942	0.941	0.938	0.939	0.940	0.938	0.943	0.941	**0.945**
Pendigit	0.495	0.491	0.432	0.378	0.281	**0.496**	0.474	0.420	0.359	0.244
Segmentation	0.645	**0.663**	0.656	0.654	0.643	0.643	0.628	0.640	0.640	**0.663**
Wheat	0.890	0.885	0.883	0.882	0.878	**0.891**	0.888	0.884	0.880	0.878
Mean	**0.661**	**0.661**	0.649	0.630	0.594	0.660	0.655	0.644	0.627	0.597

The results for the regression setting can be seen in Table 3. As with classification, we evaluate our method with both cluster- and median-based initialization and the same switching points. Similar behavior is observed between cluster- and median-based initialization, with cluster-based performing better overall. For both cluster- and median-based initialization, performance initially improves with increasing exploration, peaking at $s = 0.5$, before declining subsequently. Specifically, cluster-based initialization achieved its highest average R2-score at this point. This shows that exploration is more important for regression than classification, though exploitation is still crucial for good results. For this reason, we selected cluster-based initialization with $s = 0.5$ for our main benchmark experiments.

Table 3. Hyperparameter study for different initialization strategies (cluster- and median-based) and switch points s (regression). Results are given as R2-score. Best results marked in bold.

Init. strategy	Cluster-based					Median-based				
Switch point (s)	0.0	0.3	0.5	0.7	1.0	0.0	0.3	0.5	0.7	1.0
Airfoil Self-Noise	0.071	0.122	**0.164**	0.147	0.104	0.116	0.127	0.158	0.133	0.106
Boston Housing	0.588	**0.591**	0.586	0.565	0.540	0.534	0.545	0.569	0.558	0.507
Real Estate Val.	0.444	**0.470**	0.465	0.427	0.460	0.279	0.310	0.305	0.292	0.307
Slump Strength	0.291	0.286	0.295	0.307	**0.312**	0.282	0.276	0.277	0.293	0.295
Superconductivity	0.450	0.460	**0.480**	0.456	0.408	0.267	0.311	0.312	0.305	0.224
Yacht Hydro.	0.717	0.740	0.694	0.721	0.371	0.787	**0.802**	0.799	0.784	0.616
Mean	0.427	0.445	**0.447**	0.437	0.366	0.378	0.395	0.403	0.394	0.343

Comparison with State-of-the-Art: We compare our method with several other methods that are suitable for the low-budget regime and applicable to both classification and regression. In this category are mostly model-free AL

methods that are mentioned in chapter 2. We chose the methods in a way that we can answer RQ2 and analyze which types of criteria, namely cluster-, distance-, density- and informativeness-based, have which impact on the results. GSx has a distance-based criteria, while TypiClust and iRDM are chosen as cluster-based methods. The later two also include another criteria, typicality/density-based for TypiClust and distance-based for iRDM. Additionally, we provide the quality of random selection (RS) as a baseline that every suitable AL method should outperform. Lastly, we include QBC into the comparison. Also it is typically not seen as a suitable AL method for the low-budget scenario, we can evaluate the impact that pure exploitation has. Since QBC needs an initial set to train the model (in our case an RF with the same hyperparameters as for the evaluation), we provide it with the same initial three data points as our method LQBAL, which are sampled cluster-based (iRDM).

The results for the classification setting can be found in Table 4. As shown, GSx rarely outperforms the baseline RS for both F1-score and classes found. All other methods, however, achieve results that are at least equal to and often better than RS on all datasets, except QBC on the Segmentation dataset. The cluster-based methods iRMD and TypiClust provide similar results, with TypiClust performing slightly better in terms of both the F1-score and the number of classes identified. TypiClust also is the best method on four datasets (F1-score) with the Iris dataset being a tie. Both are the best for one dataset in finding the most classes on average[5]. Surprisingly, QBC performs also well with being the best on three datasets (F1-score), although all three results are ties. Additionally, it is only slightly better than the baseline RS in percentage of classes found. Our method LQBAL is best on three datasets, again all being ties, but it is able to find the most classes on two datasets. The mean values demonstrate how well the AL methods generally perform: TypiClust and LQBAL perform best with equal F1-scores. Although iRDM has a slightly lower F1 score, it identifies the same proportion of classes as the aforementioned methods. QBC is the fourth-best method and RS and GSx are the two worst.

The results for regression are shown in Table 5. They show a different picture than for classification which is mostly shown by the fact that no AL method is able to consistently outperform the RS baseline, with the exception of our method LQBAL. Another difference lies in performance, with ties being common in classification tasks and multiple methods often achieving similar results. In contrast, regression tasks often demonstrate performance differences of 0.2 or more, as demonstrated by the Yacht Hydrodynamics dataset. The methods GSx, iRDM, TypiClust and QBC perform similar on average. All are outperformed on four out of six datasets by RS, with the datasets Airfoil and Boston Housing being the ones where none of those methods achieves an improvement over the baseline. TypiClust performs really well on the Real Estate Valuation and Slump Strength set, which compensate for its very poor performance on the Yacht

[5] We do not mention the Iris and Wheat dataset in the analysis of percentage of classes found here and in the following, because all methods perform equally perfect in this regard.

Table 4. Results for classification (F1-score and percentage of classes found) averaged over 100 runs. Best result marked in bold.

Dataset	RS	GSx	iRDM	TypiClust	QBC	LQBAL
Fish	0.49 (0.81)	0.45 (0.74)	0.52 (0.8)	0.49 (0.78)	**0.59** (0.85)	**0.59 (0.87)**
Glass	0.34 (0.78)	0.34 (0.84)	0.39 (0.84)	**0.45 (0.9)**	0.39 (0.78)	0.37 (0.8)
Iris	0.92 (1)	**0.94** (1)	0.93 (1)	**0.94** (1)	**0.94** (1)	**0.94** (1)
Pendigit	0.40 (0.79)	0.33 (0.73)	0.52 **(0.93)**	**0.53** (0.87)	0.41 (0.77)	0.45 (0.86)
Segment.	0.56 (0.91)	0.56 (0.92)	0.61 (0.95)	**0.67** (0.96)	0.53 (0.92)	0.65 **(0.98)**
Wheat	0.87 (1)	0.87 (1)	0.88 (1)	0.88 (1)	**0.89** (1)	**0.89** (1)
Mean	0.60 (0.88)	0.58 (0.87)	0.64 **(0.92)**	**0.66 (0.92)**	0.62 (0.89)	**0.66 (0.92)**

Hydrodynamics dataset, where QBC is able to be the best method. Importantly, while all other methods have at least one dataset with a very poor performance[6] our method is able to perform best or relatively close to the best (one exception: Yacht Hydrodynamics), which is also shown by the best average result among all active learners. In terms of the average ranks, significantly LQBAL obtained the best results.

Table 5. Results for regression (R2-Score) averaged over 100 runs. Best result marked in bold.

Dataset	RS	GSx	iRDM	TypiClust	QBC	LQBAL
Airfoil Self-Noise	0.145(2)	0.126(3)	0.101(4)	0.048(5)	-0.029(6)	**0.163**(1)
Boston Housing	0.488(2)	0.376(6)	0.45(5)	0.475(4)	0.476(3)	**0.586**(1)
Real Estate Valuation	0.442(4)	0.333(6)	0.441(5)	**0.567**(1)	0.501(2)	0.466(3)
Slump Strength	0.23(5)	0.294(3)	0.281(4)	**0.313**(1)	0.191(6)	0.295(2)
Superconductivity	0.444(2)	0.125(4)	0.053(6)	0.415(3)	0.102(5)	**0.48**(1)
Yacht Hydrodynamics	0.783(4)	0.851(2)	0.791(3)	0.352(6)	**0.854**(1)	0.69(5)
Mean (AvRks.)	0.42(3.2)	0.35(4.0)	0.35(4.5)	0.36(3.3)	0.35(3.8)	**0.45**(2.2)

5 Discussion and Future Work

In this work, we present a novel active learning (AL) method tailored for scenarios with a very low query budget. Applicable to both regression and classification datasets, our method comprises two phases that combine model-free and model-based criteria to consistently produce high-quality datasets. While other methods are also applicable to regression and classification, to the best of our

[6] GSx and iRDM: Superconductivity, TpyiClust: Yacht Hydrodynamics, QBC: Airfoil and Superconductivity

knowledge, none explicitly investigate the distinctions between these settings. To address this gap, we introduced a hyperparameter that defines the transition between phases and estimated values that are generally appropriate for both classification and regression.

On average, our method (LQBAL) outperforms other commonly used methods in low-budget scenarios, such as the initialisation process, for regression and matches the performance of the best methods for classification. We now use our experimental results to address the two RQs formulated in the introduction:

RQ1: Does the inclusion of model-based AL criteria help in small data, low-budget scenarios? If yes, at which point should the inclusion happen?

- Yes, incorporating model-based criteria is advantageous. For classification, although our method does not significantly outperform existing methods, our hyperparameter analysis indicates that an early exploitation phase outperforms a later, shorter one. For regression, the results indicate that model-based criteria is critical to consistently improve on the RS baseline. Interestingly, the optimal transition point to the model-based phase varies slightly: around 30% of the budget for classification and 50% for regression. We hypothesize that classification problems are often easier to model, as a single sample per class may suffice to provide acceptable results, particularly when classes are well-clustered. In contrast, regression outputs fluctuate more, making them harder to model with limited data. This aligns with the results of [3], who suggest delaying model-based criteria until the model is sufficiently fitted to the problem.

RQ2: Which types of AL criteria (e.g. cluster-, distance-based) are generally important to consider for a high-quality, low-budget AL method?

- For classification, TypiClust and LQBAL perform best, highlighting the value of density-based AL criteria. The method iRDM also performs equally well in the covered classes found, this shows the importance of clustering in the classification setting. Model-based criteria, as evidenced by the unexpectedly solid performance of QBC, provide important information, whereas distance-based criteria, as used in GSx, alone provide no advantage. However, the top three methods all use distance measures to prevent redundant sampling, so its inclusion in this way is helpful. For regression, all methods except LQBAL are worse than RS on average, emphasizing the necessity of combining model-free and model-based criteria. Further investigation into whether cluster-, distance- or density-based criteria are more helpful may be speculative due to the minor differences, but TypiClust and iRDM again slightly outperform the others, indicating that cluster- and density-based criteria are valuable. QBC and GSx are the two worst methods, implying that distance- and model-based criteria are only effective when paired with other approaches.

For future work, we plan to investigate methods to automatically determine the switching point s, to make the performance less dependent on this hyperparameter choice. Additionally, we want to investigate the quality of the compared methods when they are not used for the whole AL process with limited bud-

get but for creating the initial set for other more advanced AL methods. This addresses the cold-start problem which is often ignored or overseen in the development of new AL methods.

Acknowledgments. This work was funded by the project *SAIL: SustAInable Life cycle of Intelligent Socio Technical Systems* (Grant ID NW21-059B), which is funded by the Ministry of Culture and Science of the State of North Rhine-Westphalia, Germany.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Attenberg, J., Provost, F.: Inactive learning? difficulties employing active learning in practice. ACM SIGKDD Explorations Newsletter **12**, 36–41 (3 2011). https://doi.org/10.1145/1964897.1964906
2. Fourcaud, T., Zhang, X., Stokes, A., Lambers, H., Körner, C.: Plant growth modelling and applications: The increasing importance of plant architecture in growth models. Annals of Botany **101**(8), 1053–1063 (04 2008). https://doi.org/10.1093/aob/mcn050
3. Hacohen, G., Dekel, A., Weinshall, D.: Active learning on a budget: Opposite strategies suit high and low budgets. In: Proceedings of the 39th International Conference on Machine Learning. vol. 162, pp. 8175–8195. PMLR (7 2022). https://doi.org/10.48550/arXiv.2202.02794
4. He, D., Yu, H., Wang, G., Li, J.: A two-stage clustering-based cold-start method for active learning. Intelligent Data Analysis **25**(5), 1169–1185 (2021)
5. Jaster, B., Kohlhase, M.: Active learning for regression problems with ensemble methods. In: Proceedings - 33. Workshop Computational Intelligence. pp. 9–29. Karlsruher Institut für Technologie (KIT) (11 2023). https://doi.org/10.5445/KSP/1000162754
6. Jose, A., de Mendonça, J.P.A., Devijver, E., Jakse, N., Monbet, V., Poloni, R.: Regression tree-based active learning. Data Mining and Knowledge Discovery (8 2023). https://doi.org/10.1007/s10618-023-00951-7
7. Kelly, M., Longjohn, R., Nottingham, K.: The UCI machine learning repository, https://archive.ics.uci.edu
8. Kumar, P., Gupta, A.: Active learning query strategies for classification, regression, and clustering: a survey. Journal of Computer Science and Technology **35**, 913–945 (2020). https://doi.org/10.1007/s11390-020-9487-4
9. Lewis, D.D., Gale, W.A.: A Sequential Algorithm for Training Text Classifiers, pp. 3–12. Springer London (1994). https://doi.org/10.1007/978-1-4471-2099-5_1
10. Liu, Z., Jiang, X., Luo, H., Fang, W., Liu, J., Wu, D.: Pool-based unsupervised active learning for regression using iterative representativeness-diversity maximization (irdm). Pattern Recognition Letters **142**, 11–19 (2 2021). https://doi.org/10.1016/j.patrec.2020.11.019
11. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics. vol. 5, pp. 281–298. University of California press (1967)

12. Rezazadeh, F., Abrishambaf, A., Dürrbaum, A., Zimmermann, G., Kroll, A.: Holistic modeling of ultra-high performance concrete production process: Synergizing mix design, fresh concrete properties, and curing conditions. In: Proceedings - 33. Workshop Computational Intelligence. pp. 215–238. Karlsruher Institut für Technologie (KIT) (11 2023). https://doi.org/10.5445/KSP/1000162754
13. Riis, C., Antunes, F., Boe, F., Carlos, H., Azevedo, L., Pereira, F.C.: Bayesian active learning with fully bayesian gaussian processes. In: Advances in Neural Information Processing Systems. pp. 12141–12153. Curran Associates, Inc. (2022)
14. Roy, N., McCallum, A.: Toward optimal active learning through sampling estimation of error reduction. In: ICML. vol. 1, p. 5. Citeseer (2001)
15. Schöne, M., Jaster, B., Bültemeier, J., Kösters, J., Holst, C.A., Kohlhase, M.: Pool-based active learning with decision trees: Incorporate the tree structure to explore and exploit. In: 2025 IEEE Symposium on Trustworthy, Explainable and Responsible Computational Intelligence (CITREx). pp. 1–9. IEEE (3 2025). https://doi.org/10.1109/CITREx64975.2025.10974940
16. Settles, B.: Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin-Madison Department of Computer Sciences (2009)
17. Seung, H.S., Opper, M., Sompolinsky, H.: Query by committee. In: Proceedings of the fifth annual workshop on Computational learning theory. pp. 287–294 (1992)
18. Sharma, M., Bilgic, M.: Evidence-based uncertainty sampling for active learning. Data Mining and Knowledge Discovery **31**, 164–202 (2017)
19. Shui, C., Zhou, F., Gagné, C., Wang, B.: Deep active learning: Unified and principled method for query and training. In: Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics. pp. 1308–1318. PMLR (2020), https://proceedings.mlr.press/v108/shui20a.html
20. Tharwat, A.: Principal component analysis-a tutorial. International Journal of Applied Pattern Recognition **3**(3), 197–240 (2016)
21. Tharwat, A., Schenck, W.: Balancing exploration and exploitation: A novel active learner for imbalanced data. Knowledge-Based Systems **210**, 106500 (12 2020). https://doi.org/10.1016/j.knosys.2020.106500
22. Tharwat, A., Schenck, W.: A survey on active learning: State-of-the-art, practical challenges and research directions. Mathematics **11**, 820 (2 2023). https://doi.org/10.3390/math11040820
23. Tharwat, A., Schenck, W.: A survey on active learning: state-of-the-art, practical challenges and research directions. Mathematics **11**(4), 820 (2023)
24. Tharwat, A., Schenck, W.: Using methods from dimensionality reduction for active learning with low query budget. IEEE Transactions on Knowledge and Data Engineering **36**(8), 4317–4330 (2024)
25. Wang, M., Min, F., Zhang, Z.H., Wu, Y.X.: Active learning through density clustering. Expert systems with applications **85**, 305–317 (2017)
26. Wu, D., Lin, C.T., Huang, J.: Active learning for regression using greedy sampling. Information Sciences **474**, 90–105 (2019). https://doi.org/10.1016/j.ins.2018.09.060
27. Yu, K., Bi, J., Tresp, V.: Active learning via transductive experimental design. In: Proceedings of the 23rd international conference on Machine learning. pp. 1081–1088 (2006)
28. Zhao, Z., Jiang, Y., Chen, Y.: Direct acquisition optimization for low-budget active learning. arXiv preprint arXiv:2402.06045 (2024)