

Article

A Novel Active Learning Regression Framework for Balancing the Exploration-Exploitation Trade-Off

Dina Elreedy * , Amir F. Atiya  and Samir I. Shaheen 

Computer Engineering Department, Cairo University, Giza 12613, Egypt

* Correspondence: dinaelreedy@eng.cu.edu.eg

Received: 31 May 2019; Accepted: 28 June 2019; Published: 1 July 2019



Abstract: Recently, active learning is considered a promising approach for data acquisition due to the significant cost of the data labeling process in many real world applications, such as natural language processing and image processing. Most active learning methods are merely designed to enhance the learning model accuracy. However, the model accuracy may not be the primary goal and there could be other domain-specific objectives to be optimized. In this work, we develop a novel active learning framework that aims to solve a general class of optimization problems. The proposed framework mainly targets the optimization problems exposed to the exploration-exploitation trade-off. The active learning framework is comprehensive, it includes exploration-based, exploitation-based and balancing strategies that seek to achieve the balance between exploration and exploitation. The paper mainly considers regression tasks, as they are under-researched in the active learning field compared to classification tasks. Furthermore, in this work, we investigate the different active querying approaches—pool-based and the query synthesis—and compare them. We apply the proposed framework to the problem of learning the price-demand function, an application that is important in optimal product pricing and dynamic (or time-varying) pricing. In our experiments, we provide a comparative study including the proposed framework strategies and some other baselines. The accomplished results demonstrate a significant performance for the proposed methods.

Keywords: active learning; exploration-exploitation; regression; optimization; mutual information; Kullback–Leibler divergence; entropy; query synthesis; demand learning; exploration-exploitation; sequential decision problems

1. Introduction

Recently, active learning has received a substantial growing interest in literature. With the abundant amounts of unlabeled data, the cost of data labelling is, generally, expensive. Thus, active learning is used for selecting the most informative “beneficial” training samples for the learning model in order to achieve high model accuracy using as few examples as possible [1]. Active learning has proved its superiority in diverse applications such as natural language processing [2] and image processing [3]. The active learning process basically proceeds as follows: first, an initial learning model is trained using a few training samples. Then, additional samples are sequentially added to the training data according to a certain querying strategy. This process repeats until a certain stopping criterion is satisfied [4].

Generally, most of the active learning research mainly focuses on querying data labels to optimize the learning model’s accuracy. Only a few contributions utilize active learning for achieving other objectives. However, in many applications, the data labeling process is costly and the ultimate goal is to optimize a domain-specific objective function, other than minimizing the learning model’s predictive error. Accordingly, in this work, we propose a comprehensive active learning framework which consists of several novel querying strategies for handling general optimization problems

where the objective could be some general utility function, not necessarily the learning model's accuracy. The problem can be framed as selecting the right trade-off for the exploration-exploitation concept. In other words, we encounter a trade-off between minimizing the uncertainty of the target objective function, known as exploration and maximizing the underlying objective function given the available function estimates, which is known as exploitation. The exploration-exploitation trade-off is encountered in machine learning [5] and optimization algorithms [6]. Furthermore, this class of optimization problems experiencing a trade-off between exploration and exploitation is prevalent in many real-world applications of various fields, such as recommender systems [7] and dynamic pricing [8].

In this paper, we provide a comprehensive analysis of active learning from the point of view of the exploration-exploitation trade-off. Our focus is on having a general optimization function, rather than prediction accuracy. For example, the user may like to select a query point that maximizes his profit. As a case study, we consider the application of the proposed active learning framework to some real-world application, namely dynamic pricing for revenue maximization in case of unknown behavior of the customers' demand [9]. Specifically, firms offering a certain good or service seek to adjust prices in a way that maximizes the obtained revenue. However, the price-demand curve which controls the relation between the price and the corresponding behavior of customers, is usually not known beforehand and has to be inferred. Generally, companies learn the price-demand curve through price experimentation by testing a number of prices and obtaining their corresponding demands from actual selling situations. On the other hand, choosing prices for revealing the price-demand relation could yield revenue losses since such prices are not designed to maximize the achieved revenue [10,11].

Therefore, we are dealing with two conflicting goals: exploration in the form of choosing prices that minimize the uncertainty of the learned demand model and exploitation in the form of setting prices to maximize the objective function, that is, the obtained revenue. The former is accomplished in a framework of active learning: what price should we suggest next to gain the most knowledge of the demand-price function?

The aforementioned problem of revenue maximization with demand learning represents a case study which can be considered an application of our proposed framework. However, the presented active learning framework is general and it can be applied to any objective optimization problem incurring a trade-off between exploration and exploitation.

The proposed active learning framework consists basically of three main active learning approaches: exploration-based, exploitation-based and balancing strategies that handle both exploration and exploitation. For the exploration-based methods, we propose several novel information-theoretic strategies with the aim of minimizing the learning model uncertainty. On the other hand, the exploitation-based methods are designed merely to optimize the target objective function, without taking into consideration the model accuracy. Finally, we present several active learning strategies specifically designed to address the exploration-exploitation trade-off by combining both objectives of optimizing the target objective and obtaining an accurate learning model.

We apply a set of experiments to evaluate the performance of our proposed active learning methods in terms of both aspects: exploitation in terms of the gained utility and exploration by measuring the regression model's accuracy. In these experiments, we compare the performance of our proposed methods to some standard baselines.

Active learning has been extensively studied in classification problems [4]. However, only few studies investigate applying active learning to regression tasks [12–14]. In this work, our presented active learning framework mainly targets regression problems. However, it could be easily adapted to handle classification problems, as well.

Active learning is generally classified into sequential and batch mode settings. In the sequential setting, one query sample is selected per iteration. On the other hand, for the batch mode, a group of samples are simultaneously selected for labeling. In this work, we adopt the sequential active learning approach.

Another scheme used for classifying the active learning methods is based on the query generation process. Specifically, active learning is classified into: pool-based and query synthesis approaches. The pool-based approach is the conventional method which is most commonly used in the active learning literature [4]. In the pool-based scheme, at each iteration, one or more query samples are selected from an unlabeled pool of existing data according to a certain querying criterion and labeling is carried out for these selected samples. On the other hand, the membership query synthesis approach selects one or more synthetic samples from the whole space. In this paper, we apply both approaches—pool-based and query synthesis. Moreover, we perform a comparative study between the two methods. From the experimental results, this work essentially elucidates the significance and the superiority of employing the query synthesis approach over the commonly used pool-based approach. More detailed results will be discussed in Sections 7 and 8.

The goal of this work is not to provide a group of active learning strategies, instead, we aim to introduce a comprehensive active learning framework including novel strategies, for handling a wide class of objective optimization problems confronting the exploration-exploitation dilemma.

The main contributions of this paper are summarized as follows:

- Provide a comprehensive active learning framework for a general objective optimization, analyzing it from the point of view of the exploration-exploitation trade-off.
- Propose several novel information-theoretic active learning strategies, designed for minimizing the learning model uncertainty.
- Design active learning methods for regression tasks.
- Present a less-myopic active learning method focusing on exploitation or target optimization.
- Develop query synthesis and pool based variants of the proposed active learning strategies and compare the two approaches.
- Apply the proposed active learning framework to a real-world application, namely dynamic pricing with demand learning, as a case study.

The paper is organized as follows: Section 2 presents a literature review. Section 3 presents the problem formulation. Section 4 briefly describes the Bayesian formulation of linear regression model that is applied in our experiments. Then, our proposed active learning strategies are represented in Section 5. After that, Section 7 presents experimental results. Section 8 discusses the main findings. Finally, Section 9 concludes the paper.

2. Related Work

In this section, we briefly review the related work.

2.1. Active Learning

A comprehensive active learning literature survey can be found in the work by Settles in Reference [4]. Mostly, active learning research is designed to query data samples which enhance the predictive power of the learning model. One of the popular active learning methods is uncertainty sampling [15], which selects a sample that the learning model is most uncertain about. The label uncertainty is often measured using the predictive label variance [16] or the label entropy [17].

Another commonly used active learning strategy is query by committee (QBC) [18]. The QBC strategy hinges on minimizing the version space [4]. A committee of learning models, generally formed using ensemble learning, are trained using the training data acquired so far. Then, the QBC strategy chooses the most controversial data sample, about which the learning models disagree the most. Roy et al. propose an active learning strategy that targets minimizing the generalization error of the learning model [19]. However, their method is computationally intensive.

2.2. Active Learning for Regression

Unlike the classification domain, there is limited work that considers active learning for the regression domain. In this work, we mainly focus on regression tasks. However, our proposed active

learning framework is general enough and could be applied to classification tasks. In this subsection we briefly overview the main methods of active learning for regression.

Several popular active learning methods have been extended and applied to regression such as query by committee (QBC) in Reference [13]. In addition, Cai et al. propose an active learning method named, Expected Model Change Maximization (EMCM). Their presented querying method selects the data samples leading to the maximum model change. In their work, they estimate the model change as the gradient of the loss function, typically squared error, with respect to the query sample under consideration.

Wu proposes an active learning approach that considers representativeness and diversity in initial data collection and sequential query selection [14]. The presented approach typically applies k-means clustering to ensure representativeness by choosing data samples that are close to clusters' centroids. Furthermore, diversity is satisfied by favoring clusters having no labeled data so far. Another work seeking to enhance diversity of data samples is presented in Reference [20].

The pool-based active learning chooses training data points without assuming a prior knowledge of the test distribution. On the other hand, the population-based active learning assumes that the test distribution of data points is known and it seeks to estimate the optimal training input density from which training data points are sampled. Sugiyama et al. develop a population-based active learning approach using weighted least-squares linear regression in Reference [21]. Their proposed method, named ALICE, aims to minimize the conditional expectation of the generalization error given the training data samples.

To our knowledge, applying information-theoretic approaches to active learning for the regression domain is limited, unlike the classification domain. In this work, as demonstrated in Section 5.2, we propose several information-theoretic based active learning querying strategies for regression.

2.3. Information-Theoretic Active Learning

In this section, we briefly describe some information-theoretic based active learning methods in literature, that are mainly designed for classification, aiming to enhance the learning model predictive performance.

Guo and Greiner exploit the potential information of the unlabeled data in their proposed active learning strategy [17]. The authors develop their active learning method based on maximizing the mutual information between the underlying query and the conditional labels of the unlabeled pool given the training data. In their method, since the true label is not known, the authors use an optimistic label for the candidate query sample, which is the label leading to the maximum mutual information about the labels of the unlabeled pool samples.

The authors of Reference [22] develop an entropy-based active learning for object recognition. The presented method seeks to minimize the expected entropy of the labels for the unlabeled pool of samples, given the training data acquired so far.

In Reference [23], the authors develop an information-theoretic active learning framework in batch setting mode. Their proposed framework seeks to maximize the mutual information between the candidate sample and the unlabeled pool of samples. The authors propose pessimistic and optimistic approximations of the mutual information by choosing the label minimizing or maximizing the conditional entropy of the labels of the unlabeled samples.

Another information-theoretic metric used in Reference [24], for active learning classification, is the Fisher information ratio (FIR). A major advantage of using the Fisher information metric specifically, is that it accounts for the diversity among the query samples. The proposed method obtains a probability mass function (PMF) over the unlabeled pool by maximizing the FI using semi-definite programming, then the chosen queries are drawn according to the optimized PMF.

In this paper, we propose novel active learning strategies for regression tasks that utilize information-theoretic concepts including: mutual information, Kullback–Leibler divergence and learning model entropy, as described in Section 5.2.

2.4. Query Synthesis versus Pool-Based AL

As previously mentioned in Section 1, active learning can be classified into pool-based and query synthesis approaches [4]. The pool-based approach is prevalent in active learning literature, however the query synthesis approach could potentially outperform the pool-based method, since unlike the pool-based, the query sample is chosen from the whole input space and not restricted by a certain pool of samples, that could not be representative for the whole input space or could not contain the optimal query samples. However, the main limitation of the query synthesis approach is that it could not be applicable for tasks requiring human annotation such as image processing and natural language processing, since the synthetically generated samples could be meaningless to the human annotator [4]. Consequently, the query synthesis could mainly be used whenever the query oracle is a scientific experiment or when the underlying input space is continuous such as: the considered dynamic pricing application [8] and some robotics applications [25].

There are a few contributions applying query synthesis for active learning querying. Query synthesis was first introduced in Reference [26]. In Reference [25], the authors approximate the version space by solving a convex optimization problem. Then, the synthetic query is generated by extracting the principal component that would shrink the version space.

In this work, we implement our proposed active learning strategies in both ways: query synthesis and pool-based. The experimental results indicate that the query synthesis approach has superior performance compared to the pool-based approach as discussed in Section 7. This is intuitively logical because the query synthesis approach optimizes the query criteria over the whole input space, so the returned solution is optimal since it is not restricted to be in a certain pool of samples. Moreover, the query synthesis approach is significantly more computationally efficient than the pool-based approach since for each iteration, the former optimizes the underlying query strategy one time, while the latter evaluates the querying strategy over all the pool samples and chooses the best sample of them to query, which is computationally intensive, especially that the pool size (the number of available unlabeled samples) is used to be large.

2.5. Active Learning for Objective Optimization

As mentioned in the introduction, Section 1 and as discussed so far, most of the active learning work in literature aims to enhance the predictive accuracy of the learning model. There are only limited research contributions that use active learning for achieving general real-world objectives other than the model predictive power. In this section, we discuss the main contributions that utilize active learning for achieving a general objective other than the learning model accuracy.

Saar and Provost design an active learning querying method named Goal-Oriented Active Learning (GOAL) for decision making. In their paper, the authors apply their proposed active learning method to customer targeting campaigns [27]. They typically consider binary decision (classification) problem, which is whether to target a specific customer or not, given that customer targeting incurs some cost. Their proposed method queries data samples that are close to decision threshold to enhance decision learning. However, the GOAL method does not consider the trade-off between learning optimal decisions and profit maximization.

Garnett et al. adopt active learning for two binary classification problems, active search and active surveying in Reference [28]. The authors utilize the Bayesian decision theory and they propose less-myopic approximations to the optimal policy by considering multiple step look-ahead of the underlying utility functions of both problems.

Marcela et al. develop an active learning approach for solving multi-objective optimization, named ϵ -Pareto Active Learning (ϵ -PAL) [29]. Their approach assumes that the considered objectives follow a Gaussian process distribution, so they use Bayesian optimization framework. However, their work does not focus on the exploration-exploitation trade-off that may exist among the underlying different objectives, which is the main concern of our presented work.

Another active learning scheme for sequential decision making is the knowledge-gradient (KG) method [30]. The KG method is an exploitation-based strategy that considers several alternatives and chooses the alternative maximizing the expected improvement of a certain utility function. The knowledge-gradient method maintains a Bayesian predictive distribution for each alternative's utility and these posterior distributions are updated upon acquiring new observations. However, the KG method could be computationally expensive for large number of alternatives.

Unlike the KG method, our framework considers the distribution of a certain utility function that is evaluated using a learning model, specifically the Bayesian linear regression. Another difference between the KG method and our proposed methods is that the KG method is inherently designed in pool-based setting where the selection is performed from a finite set of alternatives. On the other hand, our proposed approaches are general to be applied in pool-based or query synthesis setting as indicated in Section 5.1. In addition, the KG method is a pure exploitation method that does not explicitly consider exploration. However, in this work, we provide several less-myopic methods balancing between exploitation and exploration described in Section 5.4.

The mean objective cost of uncertainty (MOCU) method proposed in References [31,32] handles model uncertainty in a novel way. The MOCU method essentially studies the impact of the model uncertainty on performance degradation in terms of some incurred cost. Specifically, the MOCU criterion evaluates model uncertainty by measuring the differential cost between the current estimated model and the optimal model which minimizes the expected cost.

2.6. The Exploration-Exploitation Trade-Off

The exploration and exploitation trade-off has been extensively studied in many contexts including: reinforcement learning [5], multi-armed bandit problems [33] and evolutionary optimization [6].

Krause and Guestrin handle the trade-off between exploration and exploitation in their active learning method for handling spatial phenomena such as river monitoring [34]. The authors use Gaussian Processes (GPs) in their model, with unknown kernel parameters. They propose a non-myopic active learning approach for handling the trade-off between exploration, which aims to decrease the uncertainty about the model parameters and exploitation, which seeks to near-optimal observations using the estimated model parameters so far. However, they use static split between exploration and exploitation as two separate phases and they derive some bounds for the length of the exploration phase. On the other hand, our proposed methods described in Section 5.4 make probabilistic transitions/balance between exploration and exploitation. A dynamic balance between exploration and exploitation that is performed probabilistically could be more powerful than static balance, especially for real world applications that could have noisy observations. In such a case it is hard to predict a predefined period of exploration.

The multi-armed bandit (MAB) context is a class of sequential decision making problems originally developed in Reference [35]. The objective is to maximize rewards but under uncertainty and incomplete feedback about rewards, so there is a trade-off between performing an action that gathers information regarding reward (exploration) and making a decision that maximizes the immediate reward given the information gathered so far (exploitation). In our experiments, we apply the upper confidence bound algorithm (UCB) [36], a popular algorithm developed in the context of MAB, as a baseline to compare with.

Although the primary objective of reinforcement learning is to maximize the cumulative rewards, which is typically exploitation, exploration plays a significant role in reinforcement learning as demonstrated in Reference [5], since without exploration, the agent could simply derive sub-optimal plans. So, achieving the balance between exploration and exploitation is the core issue in reinforcement learning. However, reinforcement learning is generally computationally expensive, so it is not scalable for large state spaces. Furthermore, reinforcement learning requires a considerable amount of training data, unlike active learning which is designed for limited data requirements. The work of Reference [37] relates the concept of exploration-exploitation trade-off with bias-variance trade-off.

The exploration-exploitation trade-off has been extensively addressed in the context of evolutionary optimization. In such context, exploration is defined as visiting new regions of the search space, while exploitation denotes visiting regions of the search space within the neighborhood of previously explored points. A comprehensive review of the exploration-exploitation trade-off in evolutionary optimization is presented in Reference [6].

3. Problem Formulation

As mentioned in the introduction, this work focuses on regression tasks since it is prominent in different applications such as energy consumption prediction [38] and price-demand elasticity estimation [8,39]. Specifically, in this work, we apply linear regression model but our proposed active learning framework is general and can be applied to any other regression model. Furthermore, the proposed strategies can be adapted to classification models as well.

We consider the following linear regression problem:

$$y = \beta^T x + \epsilon \quad (1)$$

where x is the input feature vector such that $x \in \mathcal{R}^d$, where d is the dimensionality of the feature vector, y denotes the regression response variable $y \in \mathcal{R}$ and ϵ is a random error term such that $\epsilon \sim \mathcal{N}(0, \sigma^2)$ and $\beta \in \mathcal{R}^d$ denotes the regression model coefficients.

This work particularly tackles the class of optimization problems which have a certain utility function u to be optimized, for any regression task. However, the utility function u incurs some uncertainty which can be estimated using a probabilistic regression model. Such problems pose the challenging problem of how to strike a balance between maximizing the objective function u (exploitation) and minimizing the uncertainty about the utility function (exploration). In this work, we develop a novel active learning framework consisting of various strategies to interactively seek a balance between exploitation and exploration.

Notation

In this section, we introduce the adopted notation used in the proposed active learning framework.

First, the training data acquired so far is denoted as $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, the training data term \mathcal{D} is expressed in terms of a set of pairs of input data samples x_i and their corresponding labels y_i , where N is the number of data samples acquired so far.

The matrix of input data points is denoted as $X \in \mathbb{R}^{N \times d}$, such that each row x represents one data sample and d is the dimensionality of the data point x . For $Y \in \mathbb{R}^{N \times 1}$, it represents the vector of the corresponding output variables. The matrix of data samples whose outputs require prediction is denoted as $X^* \in \mathbb{R}^{m \times d}$, where m is the size of data samples to be predicted. In addition, Y^* represents the vector of the corresponding output variables and $Y \in \mathbb{R}^{m \times 1}$. Similarly, in case of predicting a single data point, the data sample is denoted as x^* and y^* is its corresponding output.

In the adopted linear regression algorithm described in Section 4, the regression coefficients are denoted as β . In addition, μ_β and Σ_β are the mean and covariance matrix of β , respectively.

In the proposed active learning framework, \mathcal{U} denotes the unlabeled pool of data samples and Y_U represents the responses of the samples in the pool. The utility function u represents the objective function to be optimized using active learning as defined in Sections 5.3 and 5.4.

4. Preliminaries: Bayesian Linear Regression

In this section, we briefly describe the Bayesian linear regression model used in the proposed active learning framework. We adopt the Bayesian linear regression model due to several reasons. First, the class of optimization problems that we handle involves uncertainty of the utility function, which can be estimated using probabilistic regression models such as the Bayesian linear regression. Moreover, most active learning querying strategies depend on the uncertainty of predictions, so it

is compelling that we use a regression model providing not only predictions but also uncertainty of the obtained predictions and Bayesian linear regression provides such information. Finally, in active learning settings, the initial data points available for training is essentially limited which could result in over-fitting, especially for noisy data, so applying Bayesian linear regression helps to combat the potential over-fitting.

The underlying regression problem is formulated as indicated in Equation (1), in Section 1. According to Equation (1), we have two major parameters in the regression model, the regression model coefficients β and the noise variance σ^2 , so we adopt Bayesian linear regression with conjugate prior of (β, σ^2) .

Since the noise variance parameter σ^2 is a key parameter in the model and we have some prior knowledge about it, for example it must be positive, we can use a conjugate prior distribution for both parameters β and σ^2 . We assume an Inverse Gamma prior distribution for σ^2 , $\sigma^2 \sim \mathcal{IG}(a_\sigma, b_\sigma)$.

$$p(\sigma^2) = \frac{(b_\sigma)^{a_\sigma}}{\Gamma(a_\sigma)} \sigma^{-2(a_\sigma+1)} e^{-\frac{b_\sigma}{\sigma^2}} \quad (2)$$

where $a_\sigma > 1$, $b_\sigma > 0$ and $\sigma^2 > 0$. The conjugate prior $p(\beta, \sigma^2)$ can be expressed as a Normal Inverse Gamma (NIG) distribution as follows:

$$p(\beta, \sigma^2) = p(\beta|\sigma^2)p(\sigma^2) = \mathcal{N}(\mu, \sigma^2 \Sigma) \mathcal{IG}(a_\sigma, b_\sigma) = \mathcal{NIG}(\mu, \Sigma, a_\sigma, b_\sigma) \quad (3)$$

- Conjugate Posterior Distribution:

According to Reference [40], the conjugate posterior distribution $p(\beta, \sigma^2|\mathcal{D})$ is a Normal Inverse Gamma (NIG) distribution as follows:

$$p(\beta, \sigma^2|\mathcal{D}) = \mathcal{NIG}(\mu_{\beta|\mathcal{D}}, \Sigma_{\beta|\mathcal{D}}, a_{\sigma|\mathcal{D}}, b_{\sigma|\mathcal{D}}) \quad (4)$$

Let μ and Σ be the prior expectation and covariance matrix of parameters β , respectively. The posterior mean $\mu_{\beta|\mathcal{D}}$ is evaluated as follows:

$$\mu_{\beta|\mathcal{D}} = (X^T X + \Sigma^{-1})^{-1} (\Sigma^{-1} \mu + X^T y) \quad (5)$$

The posterior covariance $\Sigma_{\beta|\mathcal{D}}$ is calculated as follows:

$$\Sigma_{\beta|\mathcal{D}} = (X^T X + \Sigma^{-1})^{-1} \quad (6)$$

The posterior updates of noise distribution parameters a_σ and b_σ parameters are given by:

$$a_{\sigma|\mathcal{D}} = a_\sigma + \frac{N}{2} \quad (7)$$

$$b_{\sigma|\mathcal{D}} = b_\sigma + \frac{1}{2} (y^T y + \mu^T \Sigma^{-1} \mu - \mu_{\beta|\mathcal{D}}^T \Sigma_{\beta|\mathcal{D}}^{-1} \mu_{\beta|\mathcal{D}}) \quad (8)$$

As derived in Reference [41], the marginal posterior distribution for β , denoted as $\beta|\mathcal{D}$, is a multivariate Student-t distribution as follows:

$$\beta|\mathcal{D} \sim t_{2a_{\sigma|\mathcal{D}}}(\mu_{\beta|\mathcal{D}}, \frac{b_{\sigma|\mathcal{D}}}{a_{\sigma|\mathcal{D}}} \Sigma_{\beta|\mathcal{D}}) \quad (9)$$

For a random variable Z that follows a multivariate Student-T distribution, defined as $t_v(\mu_0, \Sigma_0)$, the expectation and the covariance matrix of Z are calculated, respectively, as follows:

$$\mathbb{E}[Z] = \mu_0 \quad (10)$$

$$\Sigma_Z = \frac{v}{v-2} \Sigma_0 \quad (11)$$

where v is the number of degrees of freedom for the Student-T distribution $t_v(\mu, \Sigma)$.

According to Equations (9), (18) and (11), the expectation and the covariance matrix of the marginal $\beta|\mathcal{D}$ distribution are evaluated as follows:

$$\mathbb{E}[\beta|\mathcal{D}] = \mu_{\beta|\mathcal{D}} \quad (12)$$

$$\text{Cov}[\beta|\mathcal{D}] = \frac{b_{\sigma|\mathcal{D}}}{a_{\sigma|\mathcal{D}-1}} \Sigma_{\beta|\mathcal{D}} \quad (13)$$

- Posterior predictive distribution of Y :

As derived in Reference [40], the posterior predictive distribution $p(Y|\mathcal{D})$ is evaluated as follows:

$$p(Y|\mathcal{D}) = \int p(Y|\beta, \sigma^2) p(\beta, \sigma^2|\mathcal{D}) \quad (14)$$

$$= \mathcal{N}(X\beta, \sigma^2 \mathbb{I}_m) \times \text{NIG}(\mu_{\beta|\mathcal{D}}, \Sigma_{\beta|\mathcal{D}}, a_{\sigma|\mathcal{D}}, b_{\sigma|\mathcal{D}}) \quad (15)$$

$$= t_{2a_{\sigma|\mathcal{D}}}\left(X\mu_{\beta|\mathcal{D}}, \frac{b_{\sigma|\mathcal{D}}}{a_{\sigma|\mathcal{D}}}(\mathbb{I}_m + X\Sigma_{\beta|\mathcal{D}}X^T)\right) \quad (16)$$

To predict a vector of output responses Y^* , corresponding to a matrix of data points X^* , the posterior predictive distribution of the output vector Y^* is defined as follows:

$$p(Y^*|X^*, \mathcal{D}) \sim t_{2a_{\sigma|\mathcal{D}}}(\mathbb{E}[Y^*|X^*, \mathcal{D}], A_{Y^*|X^*, \mathcal{D}}) \quad (17)$$

The posterior expectation of the predicted responses Y^* is calculated as:

$$\mathbb{E}[Y^*|X^*, \mathcal{D}] = X^* \mu_{\beta|\mathcal{D}} \quad (18)$$

where $A_{Y^*|X^*, \mathcal{D}}$ is calculated as:

$$A_{Y^*|X^*, \mathcal{D}} = \frac{b_{\sigma|\mathcal{D}}}{a_{\sigma|\mathcal{D}}}(\mathbb{I}_m + X^* \Sigma_{\beta|\mathcal{D}} X^{*T}) \quad (19)$$

However, the covariance matrix for a multivariate Student-T distribution $t_v(\mu, A)$ is estimated as:

$$\Sigma = \frac{v}{v-2} A \quad (20)$$

Consequently, from Equation (17) and substituting from Equation (19) into Equation (20), the posterior variance of the predicted responses Y^* is calculated as:

$$\Sigma_{Y^*|X^*, \mathcal{D}} = \frac{b_{\sigma|\mathcal{D}}}{a_{\sigma|\mathcal{D}}-1}(\mathbb{I}_m + X^* \Sigma_{\beta|\mathcal{D}} X^{*T}) \quad (21)$$

To predict a single label y^* , the predictive posterior distribution $p(y^*|x^*, \mathcal{D})$ is evaluated as:

$$p(y^*|x^*, \mathcal{D}) \sim t_{2a_{\sigma|\mathcal{D} \cup \{x^*, y^*\}}}(\mathbb{E}[y^*|x^*, \mathcal{D}], \sigma_{y^*|x, \mathcal{D}}) \quad (22)$$

According to Equations (17) and (18), the posterior expectation of the predicted label y^* is calculated as:

$$\mathbb{E}[y^* | x^*, \mathcal{D}] = x^{*T} \mu_{\beta | \mathcal{D}} \tag{23}$$

Similarly, using Equations (17) and (21), the posterior variance of the predicted value y^* is defined as:

$$\sigma_{y^* | x^*, \mathcal{D}}^2 = \frac{b_{\sigma | \mathcal{D}}}{a_{\sigma | \mathcal{D}} - 1} (1 + x^{*T} \Sigma_{\beta | \mathcal{D}} x^*) \tag{24}$$

In this section, we have provided the final formulations for Bayesian linear regression model. The interested readers can find more details in References [40,42].

5. Proposed Active Learning Framework

In this section, we present our proposed active learning framework for handling optimization problems, encountering an exploration-exploitation trade-off.

First, we describe the general active learning settings. Then, we introduce our proposed active learning strategies which are mainly classified into: exploration-based, exploitation-based and strategies that balance exploration and exploitation. Figure 1 shows the proposed active learning framework.

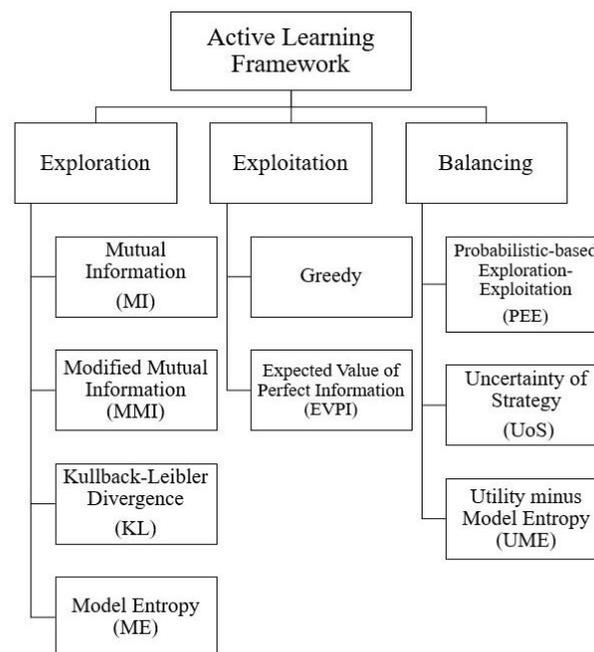


Figure 1. The Proposed Active Learning Framework.

5.1. Active Learning Schemes

Active learning can be applied in different modes that define how a new query point is generated. We describe three different schemes, the first two methods are generally known in literature and we define the third one because we incorporate it into some of our proposed strategies.

- Pool-based

This is the conventional approach that is mostly used in the active learning literature. In the pool-based approach, there exists an unlabeled pool of data samples X_U and at each iteration, one or more query example(s) x^* is selected from the pool according to a certain querying criterion. Algorithm 1 describes the pool-based active learning approach.

- Membership Query Synthesis

Unlike the pool-based approach, the membership query synthesis scheme is not commonly used in the active learning literature. In contrast with the pool-based active learning, the membership query synthesis does not select data samples out of a certain pool of unlabeled data. Alternatively, this approach essentially generates and queries synthetic data samples of the entire input space. Algorithm 2 explains the query synthesis approach.

This approach is very efficient and is not computationally intensive compared to the pool-based approach. The reason for the query synthesis's computational efficiency is that instead of iterating over the large unlabeled pool of samples and evaluating a certain selection criterion such as mutual information, the query synthesis approach directly generates a synthetic data sample to achieve a certain objective. For example, our proposed query synthesis approach optimizes the underlying querying metric using optimization algorithms. The query synthesis approach is not only computationally efficient, it could be more compelling than the pool-based approach since the generated query sample is not restricted to be part of an unlabeled pool, so the synthetically generated query sample could be more informative and beneficial than the examples in the pool.

- Membership Query Synthesis without a Predefined Pool

The query synthesis approach does not need to have a pool of samples. However, some active learning strategies exploit the potential information in the unlabeled data to guide the sample selection such as mutual information strategy defined subsequently in Equation (28) and the KL divergence strategy defined in Equation (47). Consequently, such strategies rely on the existence of some unlabeled data to estimate how useful or how representative a certain query point is. However, for some applications, the unlabeled data could not exist or if they exist, they may not be a representative sample for the input space. In such cases, one could generate a representative and diverse sample of unlabeled data using the domain knowledge of the feature space. Another way for generating unlabeled representative data could be to apply any reasonable clustering algorithm using the available training data and the cluster centroids can be used as representatives of the unobserved data. Algorithm 3 elucidates this approach.

Algorithm 1 Pool-based Active Learning

Input: A dataset $\mathcal{D} = \{(x_j, y_j)\}_{j=1}^N$, a general active learning strategy S , a utility function u , number of iterations T , a discount factor γ and a generation method for creating synthetic queries $GenerateQueryPoint()$.

Output: A Learned model θ_T and a cumulative gained utility u_T .

$\mathcal{D}_L \leftarrow N_{init}$ labeled data samples randomly chosen out of \mathcal{D} .

Train the regression model using the initial training data to obtain initial model θ_0 .

$\mathcal{D}_U \leftarrow \mathcal{D} \setminus \mathcal{D}_L$

repeat

for each $x_k \in \mathcal{D}_U$ **do**

$S(x_k) \leftarrow$ Apply a certain active learning strategy S to x_k , using current model estimate θ_i .

end for

$x^* = \arg \max_{x_k} S(x_k) \forall k, k \in \{1 \dots |\mathcal{D}_U|\}$.

$y^* \leftarrow$ the true label for the query sample x^* .

 Add the acquired data point (x^*, y^*) to the training data: $\mathcal{D}_L \leftarrow \mathcal{D}_L \cup (x^*, y^*)$.

 Evaluate the utility u_i using the new acquired point: $u_i \leftarrow u(x^*, y^*)$.

 Update the regression model θ_i using the new acquired point (x^*, y^*) .

until T iterations executed

return The learned model θ_T and the cumulative discounted utility $u_T = \sum_{i=1}^T \gamma^{i-1} u_i$.

In our experiments, we develop several novel active learning strategies and apply them in the pool-based and query synthesis schemes. For the strategies that use the unlabeled data samples for guiding its selection such as mutual information (MI), modified mutual information (MMI) and Kullback–Leibler divergence (KL), we apply the three aforementioned schemes. More details are provided in the experiments section, Section 7.

Algorithm 2 Query Synthesis Active Learning

Input: A dataset $\mathcal{D} = \{(x_j, y_j)\}_{j=1}^N$, a general active learning strategy S , a utility function u , number of iterations T , a discount factor γ and a generation method for creating synthetic queries $GenerateQueryPoint()$.

Output: A Learned model θ_T and a cumulative gained utility u_T .

$\mathcal{D}_L \leftarrow N_{init}$ labeled data samples randomly chosen out of \mathcal{D} .

Train the regression model using the initial training data to obtain initial model θ_0 .

repeat

$x^* = GenerateQueryPoint(S, \mathcal{D}_L, \theta_i)$.

$y^* \leftarrow$ the true label for the query sample x^* .

Add the acquired data point (x^*, y^*) to the training data: $\mathcal{D}_L \leftarrow \mathcal{D}_L \cup (x^*, y^*)$.

Evaluate the utility u_i using the new acquired point: $u_i \leftarrow u(x^*, y^*)$.

Update the regression model θ_i using the new acquired point (x^*, y^*) .

until T iterations executed

return The learned model θ_T and the cumulative discounted utility $u_T = \sum_{i=1}^T \gamma^{i-1} u_i$.

Algorithm 3 Query Synthesis Active Learning without a predefined pool

Input: A small dataset of N_{init} points $\mathcal{D} = \{(x_j, y_j)\}_{j=1}^{N_{init}}$, a general active learning strategy S , a utility function u , number of iterations T , a discount factor γ and a generation method for creating synthetic queries $GenerateQueryPoint()$.

Output: A Learned model θ_T and a cumulative gained utility u .

$\mathcal{D}_L \leftarrow N_{init}$ labeled data samples randomly chosen out of \mathcal{D} .

Train the regression model using the initial training data to obtain initial model θ_0 .

$\mathcal{U} \leftarrow$ Construct a representative sample of unlabeled data using for example, domain knowledge or clustering.

repeat

$x^* = GenerateQueryPoint(S, \mathcal{U}, \theta_i)$.

$y^* \leftarrow$ the true label for the query sample x^* .

Add the acquired data point (x^*, y^*) to the training data: $\mathcal{D}_L \leftarrow \mathcal{D}_L \cup (x^*, y^*)$.

Evaluate the utility u_i using the new acquired point: $u_i \leftarrow u(x^*, y^*)$.

Update the regression model θ_i using the new acquired point (x^*, y^*) .

until T iterations executed

return The learned model θ_T and the cumulative discounted utility $u_T = \sum_{i=1}^T \gamma^{i-1} u_i$.

5.2. Exploration-Based Strategies

In this section, we describe our novel proposed exploration-based active learning strategies for regression. The exploration-based strategies mainly target enhancing the regression model predictive performance. The presented strategies are not limited to a certain application or a class of problems, they are quite general and could be applied in any settings where the objective is to boost the regression model accuracy. The most popular active learning methods such as uncertain sampling [15] and Query by Committee [18] seek to query the most “uncertain” sample, that is, the data sample about

which the learning model is the most uncertain. Although this seems helpful for the learning model either classification or regression, the uncertain sampling approach does not consider the potential information of the unlabeled pool of examples. Thus, the uncertain sampling could select noisy patterns or outliers. On the other hand, querying samples not only based on the query sample but also on the unlabeled samples of the pool ([17,23]) is more promising since such approach is less myopic and it utilizes the information of the plentiful unlabeled pool.

The following proposed exploration strategies are mainly based on information theory [43]. To our knowledge, it is the first time that information theoretical concepts (such as mutual information, Kullback–Leibler divergence and model entropy) are applied in active learning for regression. Some information-theoretic metrics such as predictive label entropy, Fisher information and mutual information have been employed for active learning in classification problems [17,22,23]. However, such information theoretic metrics have not been considered yet for regression problems.

Depending solely on a single query sample information could lead to choosing noisy samples or outliers [19]. It is well-known that an outlier does more damage than help. Consequently, our proposed exploration-based active learning strategies exploit the potential information existing in the unlabeled pool of samples and the learning model uncertainty. Moreover, incorporating the information of the unlabeled pool such as mutual information, into the selection strategy, advocates querying representative samples.

5.2.1. Mutual Information (MI)

The mutual information criterion aims to query the sample x^* which effectively holds a substantial amount of information about the labels of the unlabeled pool. Thus, this strategy chooses the sample x^* that maximizes the mutual information between its label y^* and the labels of the remaining unlabeled samples of the pool excluding x^* , denoted as Y_U .

The mutual information between the query sample x^* and the labels of the unlabeled pool Y_U is defined as:

$$I(x^*, Y_U) = H(Y_U | \mathcal{D}) - H(Y_U | x^*, \mathcal{D}) \quad (25)$$

where \mathcal{D} denotes the labeled training data acquired so far.

The first term $H(Y_U | \mathcal{D})$ represents the prior entropy (or uncertainty) of all the labels of the unlabeled pool of samples. Similarly, the second term $H(Y_U | x^*, \mathcal{D})$ denotes the entropy of the labels of unlabeled pool of samples but after acquiring the new query point x^* . From Equation (25), it can be noted that maximizing $I(x^*, Y_U)$ is equivalent to minimizing the conditional entropy $H(Y_U | x^*, \mathcal{D})$, which is defined as follows:

$$H(Y_U | x^*, \mathcal{D}) = \int_{y^*} p(y^* | x^*, \mathcal{D}) H(Y_U | x^*, y^*, \mathcal{D}) dy^* \quad (26)$$

To simplify computations, Equation (26) could be approximated by eliminating the integration over all the possible labels of y^* and using the expected value of it $\mathbb{E}[y^*]$. Other approximations are made in literature [22,23], using the optimistic or the pessimistic label. However, we found that employing the expectation could be more reasonable. Accordingly:

$$H(Y_U | x^*, \mathcal{D}) = H(Y_U | (x^*, y^*), \mathcal{D}) \quad (27)$$

where y^* is the expected predicted label of the data point x^* , which is calculated using Equation (23).

$$H(Y_U | (x^*, y^*), \mathcal{D}) = \int_Y p(Y | X, \mathcal{D} \cup (x^*, y^*)) \log(p(Y | X, \mathcal{D} \cup (x^*, y^*))) dY \quad (28)$$

As mentioned in Section 4, the posterior predictive distribution of the predictive labels vector Y , $p(Y | X, \mathcal{D} \cup (x^*, y^*))$ is a multivariate Student-T distribution which is defined as follows:

$$p(Y|X, \mathcal{D} \cup (x^*, y^*)) = t_{2a_{\sigma|\mathcal{D} \cup (x^*, y^*)}}(Y, \mathbb{E}[Y|X, \mathcal{D} \cup (x^*, y^*)], \Sigma_{Y|X, \mathcal{D} \cup (x^*, y^*)}) \tag{29}$$

The posterior expectation $\mathbb{E}[Y|X, \mathcal{D} \cup (x^*, y^*)]$ and the covariance matrix $\Sigma_{Y|X, \mathcal{D} \cup (x^*, y^*)}$ are evaluated using the Bayesian linear regression model formulations described in Section 4, using Equations (18) and (21), respectively. However, this method and all our proposed methods are general and can be applied using any regression model that provides uncertainty of its predictions.

According to Reference [41], the final formulation of the entropy of a random variable Z following a Student-t distribution t_{v_Z} is given by:

$$H(Z) = \frac{1}{2} \log \det(R_Z) + \phi(v_Z, d) + \frac{v_Z + d}{2} M(v_Z, d, \Delta) \tag{30}$$

where R_Z denotes the correlation matrix of Z , d is the dimensionality of Z and v_Z represents the number of degrees of freedom for the Student-t distribution. In addition, $\phi(v_Z, d)$ is a constant depending on v_Z and d and $\Delta_Z = \mu_Z^{*T} R_Z^{*-1} \mu_Z^*$.

$$M(a_Z, d, \Delta) = e^{-\frac{\Delta_Z}{2}} \sum_{j=0}^{\infty} \frac{1}{j!} \left\{ \Psi\left(\frac{d + v_Z + 2j}{2}\right) - \Psi\left(\frac{v_Z}{2}\right) \right\} \tag{31}$$

where Ψ is the digamma function which is defined as:

$$\Psi(x) = \frac{d}{dx} \ln(\Gamma(x)) \tag{32}$$

Accordingly, the conditional entropy of Y_U , $H(Y_U|(x^*, y^*))$, is calculated using Equation (30) as follows:

$$H(Y_U|(x^*, y^*), \mathcal{D}) = \frac{1}{2} \log \det(R_Y^*) + \phi(2a^*, m) + \frac{2a^* + m}{2} M(2a^*, m, \Delta_Y) \tag{33}$$

where m is the number of data points to be predicted, that is, it is the length of the predicted output vector Y_U . To simplify notation, let $a^* = a_{\sigma|\mathcal{D} \cup (x^*, y^*)}$, $\mu_Y^* = \mu_{Y|\mathcal{D} \cup (x^*, y^*)}$, $\Sigma_Y^* = \Sigma_{Y|\mathcal{D} \cup (x^*, y^*)}$ and $R_Y^* = R_{Y|\mathcal{D} \cup (x^*, y^*)}$. For Δ_Y , it is evaluated as follows:

$$\Delta_Y = \mu_Y^{*T} R_Y^{*-1} \mu_Y^* \tag{34}$$

such that R_Y^* denotes the correlation matrix of the unlabeled samples Y after acquiring the query sample x^* .

The term $M(2a^*, m, \Delta_Y)$ can be evaluated using Equation (31):

$$M(2a^*, m, \Delta_Y) = e^{-\frac{\mu_Y^{*T} R_Y^{*-1} \mu_Y^*}{2}} \sum_{j=0}^{\infty} \frac{1}{j!} \left\{ \Psi\left(\frac{m + 2a^* + 2j}{2}\right) - \Psi(a^*) \right\} \tag{35}$$

Using algebraic manipulations, the summation in Equation (35) converges as follows:

$$\sum_{j=0}^{\infty} \frac{1}{j!} \left\{ \Psi\left(\frac{m + a^* + 2j}{2}\right) - \Psi\left(\frac{a^*}{2}\right) \right\} = e \times (\Psi(a^* + \frac{m}{2} + 1)) - \Psi(a^*) \tag{36}$$

Accordingly, substituting from Equation (36) into Equation (35) results in:

$$M(2a^*, d, \Delta_Y) = (\Psi(a^* + \frac{m}{2} + 1)) - \Psi(a^*) \times e^{-\frac{\mu_Y^{*T} R_Y^{*-1} \mu_Y^*}{2} + 1} \tag{37}$$

Then, after substituting from Equation (37) into Equation (33), the conditional entropy $H(Y_U|(x^*, y^*), \mathcal{D})$ can be evaluated as:

$$H(Y_U|(x^*, y^*), \mathcal{D}) = \frac{1}{2} \log \det(R_Y^*) + \phi(2a^*, m) + (2a^* + \frac{m}{2})(\Psi(a^* + \frac{m}{2} + 1)) - \Psi(a^*) \times e^{-\frac{\mu_Y^{*T} R_Y^{*-1} \mu_Y^*}{2} + 1} \quad (38)$$

Finally, the query sample x^* that maximizes the mutual information essentially minimizes the conditional entropy of the unlabeled pool of samples as indicated in Equation (25). Consequently, the query sample x^* minimizing the conditional entropy $H(Y_U|(x^*, y^*), \mathcal{D})$ can be evaluated as follows:

$$x_{MI} = \arg \min_{x^*} \frac{1}{2} \log \det(R_{Y|\mathcal{D} \cup (x^*, y^*)}) + \phi(2a_{\sigma|\mathcal{D} \cup (x^*, y^*)}, m) + (a_{\sigma|\mathcal{D} \cup (x^*, y^*)} + \frac{m}{2}) \times (\Psi(a_{\sigma|\mathcal{D} \cup (x^*, y^*)} + \frac{m}{2} + 1)) - \Psi(a_{\sigma|\mathcal{D} \cup (x^*, y^*)}) e^{-\frac{\mu_{Y|\mathcal{D} \cup (x^*, y^*)}^T R_{Y|\mathcal{D} \cup (x^*, y^*)}^{-1} \mu_{Y|\mathcal{D} \cup (x^*, y^*)}}{2} + 1} \quad (39)$$

Simplifying Equation (39) by eliminating the term $\phi(2a_{\sigma|\mathcal{D} \cup (x^*, y^*)}, m)$ since it is a constant that does not depend on the query sample, because $a_{\sigma|\mathcal{D} \cup (x^*, y^*)}$ basically depends on the number of data being observed, as indicated in Equation (7). Thus:

$$x_{MI} = \arg \min_{x^*} \frac{1}{2} \log \det(R_{Y|\mathcal{D} \cup (x^*, y^*)}) + (a_{\sigma|\mathcal{D} \cup (x^*, y^*)} + \frac{m}{2})(\Psi(a_{\sigma|\mathcal{D} \cup (x^*, y^*)} + \frac{m}{2} + 1)) - \Psi(a_{\sigma|\mathcal{D} \cup (x^*, y^*)}) \times e^{-\frac{\mu_{Y|\mathcal{D} \cup (x^*, y^*)}^T R_{Y|\mathcal{D} \cup (x^*, y^*)}^{-1} \mu_{Y|\mathcal{D} \cup (x^*, y^*)}}{2} + 1} \quad (40)$$

For computational efficiency purposes, we evaluate the log determinant of the correlation matrix R_Y and its inverse using Cholesky decomposition since the correlation matrix is a symmetric positive semi-definite matrix.

We apply three variants of this active learning strategy: pool-based, query synthesis and query synthesis without pool, which are described in Section 5.1.

5.2.2. Modified Mutual Information (MMI)

The modified mutual information strategy is basically akin to the aforementioned strategy. This method maximizes the mutual information defined in Equation (25) but it evaluates the first term of that equation, $H(Y_U|\mathcal{D})$, which represents the entropy of the labels of the unlabeled samples and does not ignore it. The intuition of this querying strategy is to account for the impact of the query sample (x^*, y^*) on reducing the joint entropy of the unlabeled samples $H(Y_U|\mathcal{D})$. In other words, if the first term is ignored and we just focus on minimizing the conditional entropy given the underlying query sample $H(Y_U|(x^*, y^*), \mathcal{D})$, we may choose a sample x^* that is redundant and not informative in case the entropy before acquiring x^* , $H(Y_U|\mathcal{D})$ is inherently negligible.

Accordingly, the modified mutual information equation is defined using Equation (25) but without ignoring the first term.

$$I(x^*, Y_U) = H(Y_U|\mathcal{D}) - H(Y_U|(x^*, y^*), \mathcal{D}) \quad (41)$$

Similar to the mutual information strategy, the second term of Equation (41) can be evaluated using Equation (38). As for the first term $H(Y_U|\mathcal{D})$, similar to Equation (38), it can be computed as follows:

$$H(Y_U|\mathcal{D}) = \frac{1}{2} \log \det(R_Y) + \phi(2a, m) + (a + \frac{m}{2})(\Psi(a + \frac{m}{2} + 1)) - \Psi(a) e^{-\frac{\mu_Y^T R_Y^{-1} \mu_Y}{2} + 1} \quad (42)$$

where $a = a_{\sigma|\mathcal{D}}$, $\mu_Y = \mu_{Y|\mathcal{D}}$, $\Sigma_Y = \Sigma_{Y|\mathcal{D}}$, and $R_Y = R_{Y|\mathcal{D}}$. For Δ_Y it is evaluated as follows:

$$\Delta_Y = \mu_Y^T R_Y^{-1} \mu_Y \quad (43)$$

where R_Y denotes the correlation matrix of the unlabeled samples Y , given the training data acquired so far \mathcal{D} .

Therefore, substituting from Equations (28) and (42) into Equation (41) results in:

$$I(x^*, Y_U) = \frac{1}{2} \log \det(R_{Y|\mathcal{D} \cup (x^*, y^*)}) + \phi(2a_{\sigma|\mathcal{D} \cup (x^*, y^*)}, m) + (a_{\sigma|\mathcal{D} \cup (x^*, y^*)} + \frac{m}{2}) \times (\Psi(a_{\sigma|\mathcal{D} \cup (x^*, y^*)} + \frac{m}{2} + 1) - \Psi(a_{\sigma|\mathcal{D} \cup (x^*, y^*)})) e^{-\frac{\mu_Y^T R_{Y|\mathcal{D} \cup (x^*, y^*)}^{-1} R_{Y|\mathcal{D} \cup (x^*, y^*)} \mu_Y}{2} + 1} - \frac{1}{2} \log \det(R_Y) + \phi(2a, m) + (a + \frac{m}{2})(\Psi(a + \frac{m}{2} + 1) - \Psi(a)) e^{-\frac{\mu_Y^T R_Y^{-1} \mu_Y}{2} + 1} \tag{44}$$

$$x_{MMI} = \arg \max_{x^*} \frac{1}{2} (\log \det(R_{Y|\mathcal{D} \cup (x^*, y^*)}) - \log \det(R_Y)) + (a_{\sigma|\mathcal{D} \cup (x^*, y^*)} + \frac{m}{2}) \times (\Psi(a_{\sigma|\mathcal{D} \cup (x^*, y^*)} + \frac{m}{2} + 1) - \Psi(a_{\sigma|\mathcal{D} \cup (x^*, y^*)})) e^{-\frac{\mu_Y^T R_{Y|\mathcal{D} \cup (x^*, y^*)}^{-1} R_{Y|\mathcal{D} \cup (x^*, y^*)} \mu_Y}{2} + 1} - (a + \frac{m}{2})(\Psi(a + \frac{m}{2} + 1) - \Psi(a)) e^{-\frac{\mu_Y^T R_Y^{-1} \mu_Y}{2} + 1} \tag{45}$$

Similar to the previous strategy, we apply three variants of this active learning method using the different active learning schemes: pool-based, query synthesis and query synthesis without pool.

5.2.3. Kullback–Leibler Divergence (KL)

So far, the previously mentioned strategies select the sample revealing the most amount of information for the labels of the other samples. However, this strategy addresses a different aspect. The Kullback–Leibler divergence strategy seeks to acquire samples having the greatest impact on the posterior predictive distribution of the unlabeled samples $p(Y_U|X, \mathcal{D})$. So, this method considers the influence of the query sample on the “distribution” of the unlabeled samples. To achieve that, this method maximizes the difference in posterior predictive distributions of unlabeled pool Y_U before and after querying the query point (x^*, y^*) . The distribution difference is evaluated using the Kullback–Leibler divergence (KL) metric [44]. The Kullback–Leibler divergence metric is an asymmetric distance measure that evaluates the distance between two probability distributions P and Q . In other words, $D_{KL}(P||Q)$ measures the information lost when Q is used to approximate P [44]. The $D_{KL}(P||Q)$ is defined as follows:

$$D_{KL}(P||Q) = \int_{-\infty}^{\infty} p(x) \log \frac{p(x)}{q(x)} dx$$

where $p(x)$ and $q(x)$ are the probability density functions to be compared.

It is worth noting that the KL divergence has been employed as a powerful method in Bayesian analysis. For example, Lopez et al. apply the KL divergence to influence analysis [45]. The authors use the KL metric to study the impact of removing one or several observations from data set on the inferences.

In our proposed active learning method, $p(x)$ denotes the posterior predictive distribution of unlabeled example given the query sample (x^*, y^*) , whereby $q(x)$ is the posterior predictive distribution of unlabeled example prior to acquiring the query example (x^*, y^*) . The Kullback–Leibler divergence $D_{KL}(U|\mathcal{D}, x^*)$ is defined as:

$$D_{KL}(U|\mathcal{D}, x^*) = D_{KL}(p(Y_U|\mathcal{D}, x^*, y^*), p(Y_U|\mathcal{D})) \tag{46}$$

We approximate $D_{KL}(U|\mathcal{D}, x^*)$ by evaluating the average Kullback–Leibler divergence over all the unlabeled examples of the pool Y_U .

$$D_{KL}(U|\mathcal{D}, x^*) = \frac{1}{|U|} \sum_{k \in U} D_{KL}(p(y_k|\mathcal{D}, (x^*, y^*)), p(y_k|\mathcal{D})) \tag{47}$$

Since the true label y^* of the query sample x^* is unknown, we use the expectation of y^* denoted as $\mathbb{E}[y^*|\mu_\beta, x^*]$.

As indicated in Section 4, both predictive distributions $p(y_{k|\mathcal{D},(x^*,y^*)})$ and $p(y_{k|\mathcal{D}})$ follow Student-t distributions. Let $p(y_{k|\mathcal{D},(x^*,y^*)}) \sim t_{2a_{\sigma|\mathcal{D}\cup(x^*,y^*)}}(y_k, \mathbb{E}[y_k|x_k, \mathcal{D} \cup (x^*, y^*)], \sigma^2_{y_k|x_k, \mathcal{D}\cup(x^*,y^*)})$. Similarly, the posterior predictive distribution after acquiring y^* is denoted as $p(y_{k|\mathcal{D}}) \sim t_{2a_{\sigma|\mathcal{D}}}(y_k, \mathbb{E}[y_k|x_k, \mathcal{D}], \sigma^2_{y_k|x_k, \mathcal{D}})$.

To simplify notation, let $D_{KL}(k|\mathcal{D}, (x^*, y^*))$ denote the Kullback–Leibler divergence between the two predictive distributions $D_{KL}(p(y_{k|\mathcal{D},x^*,y^*}), p(y_{k|\mathcal{D}}))$, which is calculated as:

$$D_{KL}(k|\mathcal{D}, x^*) = \int_{-\infty}^{\infty} p(y_{k|\mathcal{D},x^*,y^*}) \log \frac{p(y_{k|\mathcal{D},x^*,y^*})}{p(y_{k|\mathcal{D}})} dx \tag{48}$$

Substituting with the Student-t distribution formulation Equation (22) into Equation (48):

$$D_{KL}(k|\mathcal{D}, x^*) = \int_{-\infty}^{\infty} t_{2a_{\sigma|\mathcal{D}\cup(x^*,y^*)}}(y_k, \mathbb{E}[y_k|x_k, \mathcal{D} \cup (x^*, y^*)], \sigma^2_{y_k|x_k, \mathcal{D}\cup(x^*,y^*)}) \times \log \frac{t_{2a_{\sigma|\mathcal{D}\cup(x^*,y^*)}}(y_k, \mathbb{E}[y_k|x_k, \mathcal{D} \cup (x^*, y^*)], \sigma^2_{y_k|x_k, \mathcal{D}\cup(x^*,y^*)})}{t_{2a_{\sigma|\mathcal{D}}}(y_k, \mathbb{E}[y_k|x_k, \mathcal{D}], \sigma^2_{y_k|x_k, \mathcal{D}})} dx \tag{49}$$

where the means and variances of the posterior distributions can be given using the regression equations in Section 4, Equations (23) and (24), respectively.

After substituting from Equation (49) into Equation (47), the Kullback–Leibler divergence $D_{KL}(U|\mathcal{D}, x^*)$ is evaluated as:

$$D_{KL}(U|\mathcal{D}, x^*) = \frac{1}{|U|} \sum_{k \in U} \int_{-\infty}^{\infty} t_{2a_{\sigma|\mathcal{D}\cup(x^*,y^*)}}(y_k, \mathbb{E}[y_k|x_k, \mathcal{D} \cup (x^*, y^*)], \sigma^2_{y_k|x_k, \mathcal{D}\cup(x^*,y^*)}) \times \log \frac{t_{2a_{\sigma|\mathcal{D}\cup(x^*,y^*)}}(y_k, \mathbb{E}[y_k|x_k, \mathcal{D} \cup (x^*, y^*)], \sigma^2_{y_k|x_k, \mathcal{D}\cup(x^*,y^*)})}{t_{2a_{\sigma|\mathcal{D}}}(y_k, \mathbb{E}[y_k|x_k, \mathcal{D}], \sigma^2_{y_k|x_k, \mathcal{D}})} dx \tag{50}$$

Finally, the query sample x^* that maximizes the Kullback–Leibler divergence between the posterior predictive distributions of unlabeled pool Y_{UL} before and after querying the query point (x^*, y^*) is evaluated as follows:

$$x_{KL} = \arg \max_{x^*} \frac{1}{|U|} \sum_{k \in U} \int_{-\infty}^{\infty} t_{2a_{\sigma|\mathcal{D}\cup(x^*,y^*)}}(y_k, \mathbb{E}[y_k|x_k, \mathcal{D} \cup (x^*, y^*)], \sigma^2_{y_k|x_k, \mathcal{D}\cup(x^*,y^*)}) \times \log \frac{t_{2a_{\sigma|\mathcal{D}\cup(x^*,y^*)}}(y_k, \mathbb{E}[y_k|x_k, \mathcal{D} \cup (x^*, y^*)], \sigma^2_{y_k|x_k, \mathcal{D}\cup(x^*,y^*)})}{t_{2a_{\sigma|\mathcal{D}}}(y_k, \mathbb{E}[y_k|x_k, \mathcal{D}], \sigma_{y_k|x_k, \mathcal{D}})} dx \tag{51}$$

Like the two aforementioned active learning methods, we apply the three variants of active learning settings described in Section 5.1, along with this active learning method.

5.2.4. Model Entropy (ME)

The aforementioned strategies, the two variants of mutual information and Kullback–Leibler divergence, exploit the potential information of the unlabeled pool to guide the query selection process. However, this novel active learning strategy, named model entropy, considers a different aspect. The model entropy method targets the ultimate objective for the exploration, as mentioned in Section 1, which is minimizing the learning model uncertainty. In order to achieve this target, this method emphasizes reducing the learning model uncertainty in terms of the model entropy. Thus, this method queries the data sample that minimizes the model entropy in order to reveal the uncertainty of the underlying model and obtain better estimates of the learning model parameters.

In general, the entropy has been used in several applications such as biological systems [46], financial applications [47] and model selection [48]. However, to the best of our knowledge, the use of the model entropy minimization strategy in the active learning field is novel.

According to Reference [4], the existing work in active learning literature so far mainly addresses the following: minimizing the approximate generalization error [19] and reducing the model

uncertainty indirectly either by choosing the example about which the model is most uncertain [15] or by querying the example that produces the maximum model change [12].

For the general regression problem formulation presented in Section 3, using Equation (1), the model entropy of the regression model parameters β can be formulated as follows:

$$H(\beta|x^*, \mathcal{D}) = - \int_{\beta} p(\beta|x^*, \mathcal{D}) \log p(\beta|x^*, \mathcal{D}) d\beta \tag{52}$$

Using the Bayesian linear regression formulation presented in Section 4 (see Equation (9)), the model parameter β follows a multivariate Student-t distribution such that:

$$p(\beta|\mathcal{D}, x^*, y^*) \sim t_{2a_{\sigma|\mathcal{D}\cup(x^*, y^*)}} \left(\mu_{\beta|\mathcal{D}\cup(x^*, y^*)}, \frac{b_{\sigma|\mathcal{D}\cup(x^*, y^*)}}{a_{\sigma|\mathcal{D}\cup(x^*, y^*)} - 1} \Sigma_{\beta|\mathcal{D}\cup(x^*, y^*)} \right) \tag{53}$$

$$H(\beta|x^*, y^* \mathcal{D}) = - \int_{\beta} t_{2a_{\sigma|\mathcal{D}\cup(x^*, y^*)}} \left(\mu_{\beta|\mathcal{D}\cup(x^*, y^*)}, \frac{b_{\sigma|\mathcal{D}\cup(x^*, y^*)}}{a_{\sigma|\mathcal{D}\cup(x^*, y^*)} - 1} \Sigma_{\beta|\mathcal{D}\cup(x^*, y^*)} \right) \times \log t_{2a_{\sigma|\mathcal{D}\cup(x^*, y^*)}} \left(\mu_{\beta|\mathcal{D}\cup(x^*, y^*)}, \frac{b_{\sigma|\mathcal{D}\cup(x^*, y^*)}}{a_{\sigma|\mathcal{D}\cup(x^*, y^*)} - 1} \Sigma_{\beta|\mathcal{D}\cup(x^*, y^*)} \right) d\beta \tag{54}$$

where $\mu_{\beta|\mathcal{D},(x^*, y^*)}$ and $\Sigma_{\beta|\mathcal{D},(x^*, y^*)}$ are the posterior mean and covariance matrix of model parameter β , respectively and they can be evaluated using Equations (12) and (13), respectively, according to the Bayesian linear regression formulation described in Section 4. Furthermore, the posterior values, $a_{\sigma|\mathcal{D}\cup(x^*, y^*)}$ and $b_{\sigma|\mathcal{D}\cup(x^*, y^*)}$ are evaluated using Equations (7) and (8), respectively.

To simplify notation, let $a^* = a_{\sigma|\mathcal{D}\cup(x^*, y^*)}$, $\mu_{\beta}^* = \mu_{\beta|\mathcal{D}\cup(x^*, y^*)}$ and $\Sigma_{\beta}^* = \frac{b_{\sigma|\mathcal{D}\cup(x^*, y^*)}}{a_{\sigma|\mathcal{D}\cup(x^*, y^*)} - 1} \Sigma_{\beta|\mathcal{D}\cup(x^*, y^*)}$. According to Reference [41], the final formulation of the entropy for the multivariate Student-t distribution is given by:

$$H(\beta|x^*, y^* \mathcal{D}) = \frac{1}{2} \log \det(R_{\beta|\mathcal{D}\cup(x^*, y^*)}) + \phi(2a^*, d) + (a^* + \frac{d}{2})M(2a^*, d, \Delta_{\beta}) \tag{55}$$

where $R_{\beta|\mathcal{D}\cup(x^*, y^*)}$ denotes the correlation matrix of β , d is the dimensionality of β , ϕ is a constant depending on d and a^* and $\Delta_{\beta} = \mu_{\beta}^{*T} \Sigma_{\beta}^{*-1} \mu_{\beta}^*$.

$$M(2a^*, d, \Delta_{\beta}) = e^{-\frac{\mu_{\beta}^{*T} \Sigma_{\beta}^{*-1} \mu_{\beta}^*}{2}} \sum_{j=0}^{\infty} \frac{1}{j!} \left\{ \Psi\left(\frac{d+a^*+2j}{2}\right) - \Psi\left(\frac{a^*}{2}\right) \right\} \tag{56}$$

Substituting from Equation (36) into Equation (56):

$$M(2a^*, d, \Delta_{\beta}) = \left((\Psi(a^* + \frac{d}{2} + 1)) - \Psi(a^*) \right) e^{-\frac{\mu_{\beta}^{*T} \Sigma_{\beta}^{*-1} \mu_{\beta}^*}{2} + 1} \tag{57}$$

Substituting from Equation (56) into Equation (55) results in:

$$H(\beta|x^*, y^* \mathcal{D}) = \frac{1}{2} \log \det(R_{\beta|\mathcal{D}\cup(x^*, y^*)}) + \phi(2a_{\mathcal{D}\cup(x^*, y^*)}, d) + (a_{\mathcal{D}\cup(x^*, y^*)} + \frac{d}{2}) \times \left((\Psi(a_{\mathcal{D}\cup(x^*, y^*)} + \frac{d}{2} + 1)) - \Psi(a_{\mathcal{D}\cup(x^*, y^*)}) \right) e^{-\frac{(a_{\sigma|\mathcal{D}\cup(x^*, y^*)} - 1) \mu_{\beta}^{*T} \Sigma_{\beta}^{*-1} \mu_{\beta}^*}{2b_{\sigma|\mathcal{D}\cup(x^*, y^*)}} + 1} \tag{58}$$

However, we could safely ignore the term $\phi(2a^*, d)$ since it does not depend on the query sample x^* . Finally, the query sample x^* minimizing the model entropy $H(\beta|x^*, y^* \mathcal{D})$ can be estimated as follows:

$$x_{ME} = \arg \min_{x^*} \frac{1}{2} \log \det(R_{\beta|\mathcal{D} \cup (x^*, y^*)}) + (a_{\mathcal{D} \cup (x^*, y^*)} + \frac{d}{2}) \times \left((\Psi(a_{\mathcal{D} \cup (x^*, y^*)} + \frac{d}{2} + 1)) - \Psi(a_{\mathcal{D} \cup (x^*, y^*)}) \right) e^{-\frac{(a_{\sigma|\mathcal{D} \cup (x^*, y^*)} - 1) \mu_{\beta}^T \mathcal{D} \cup (x^*, y^*) \Sigma_{\beta}^{-1} \mathcal{D} \cup (x^*, y^*) \mu_{\beta}}{2b_{\sigma|\mathcal{D} \cup (x^*, y^*)}} + 1} \tag{59}$$

For this strategy, we apply both of pool-based and query synthesis active learning approaches.

5.3. Exploitation-Based Strategies

In this section, we present the exploitation-based active learning strategies for regression that we apply in our proposed framework. Such strategies purely emphasize on maximizing a certain objective function, with no consideration given to the concept of exploration.

First, we describe the greedy strategy that is considered a pure exploitation method. Then, we propose using a novel active learning querying strategies that mainly focus on exploitation but in a less myopic way than the commonly adopted greedy strategy.

5.3.1. Greedy Strategy (G)

This query strategy addresses pure exploitation by querying the sample resulting in the maximum immediate value of the target objective function (reward). We apply this method as a baseline to compare with, where for every iteration, the query sample is chosen to maximize the expected utility function.

Although the greedy strategy is straightforward and simple, it is myopic in since that it purely considers exploitation, which could result in potential revenue loss, since it pays no attention to improving the model predictive power, which could severely affect the resulting decision, which is commonly known as exploration.

$$x_G = \arg \max_x \mathbb{E}[u(x)|\mathcal{D}] \tag{60}$$

where the expected utility function u can be expressed as a function of x and the regression model coefficients β .

5.3.2. Expected Value of Perfect Information (EVPI)

We propose a decision-theoretic querying approach which is based on the expected value of perfect information (EVPI). Evaluating the expected value of perfect information could be beneficial for active learning since one can evaluate how revealing a certain query sample is valuable. In other words, active learning could be guided to choose data points that do improve the gained expected utility using EVPI. According to Russell and Norvig [49], the expected value of perfect information for revealing a piece of information, named evidence E_j , given an initial evidence e is defined as:

$$\mathbb{E}[VPI_e(E_j)] = \sum_k P(E_j = e_{jk}|e) \mathbb{E}[u(\alpha_{e_{jk}}|e, E_j = e_{jk})] - \mathbb{E}[u(\alpha|e)] \tag{61}$$

where α is the action to be taken and the expected utility of taking action α given the evidence e and after revealing E_j , $\mathbb{E}[u(\alpha_{e_{jk}}|e, E_j = e_{jk})]$ is defined as:

$$\mathbb{E}[u(\alpha_{e_j}|e, e_j)] = \max_a \sum_{s'} P(Result(a) = s', e, e_j) u(s') \tag{62}$$

while the expected utility of taking action α given the evidence e and without revealing E_j is denoted by $\mathbb{E}[u(\alpha|e)]$ and it is defined as:

$$\mathbb{E}[u(\alpha|e)] = \arg \max_a \sum_{s'} P(\text{Result}(a) = s', e) u(s') \quad (63)$$

We apply the value of information formulation to the active learning with utility maximization. So, the action α is querying a data point x^* to obtain its label y^* . For the initial evidence e , it denotes the training labeled data points so far \mathcal{D} . Also, e_j represents the acquired label y^* of the query point x^* which represents the piece of information we seek to evaluate.

The expected value of perfect information after querying x^* and acquiring its label y^* is:

$$\mathbb{E}[VPI_e(y^*)] = \int_{y_j} P(y_j|x^*, \mathcal{D}) \mathbb{E}[u(x^*|\mathcal{D}, y_j)] - \mathbb{E}[u(x^*|\mathcal{D})] \quad (64)$$

Accordingly, the expected utility of acquiring the data sample x^* given the observed training data so far \mathcal{D} and after observing the evidence y_j , the true value of y^* , $\mathbb{E}[u(x^*|\mathcal{D}, y_j)]$, can be formulated as:

$$\mathbb{E}[u(x_{y_j}^*|\mathcal{D}, y_j)] = \max_x \mathbb{E}[u(x|\mathcal{D}, y_j)] = \mathbb{E}[u_{opt}|\mathcal{D}, y_j] \quad (65)$$

where the utility u is the target objective function to be maximized, which is conventionally a function of the data point x and the model parameters β . μ_{β}^* denotes the expectation of the updated model parameters β after revealing point x and its label value y_j .

The second term of Equation (64) could be safely ignored, since the objective is to decide whether to acquire the data label y^* or not, maximizing EVPI, this term is independent of y^* . This is implied by the following equation, Equation (66). Consequently, this term does not affect the process of maximizing EVPI.

$$\mathbb{E}[u(x^*|\mathcal{D})] = \max_x \mathbb{E}[u(x|\mathcal{D})] = \mathbb{E}[u_{opt}|\mathcal{D}] \quad (66)$$

Consequently, the term $\mathbb{E}[u_{opt}|\mathcal{D}]$ is constant over all query points x^* , so it could be safely ignored. Then, evaluating the EVPI by substituting from Equation (65) into Equation (64) results in the following formula:

$$\mathbb{E}[VPI_{\mathcal{D}}(y^*)] = \int_{y_j} P(y_j|\mathcal{D}, x^*) \mathbb{E}[u_{opt}|\mathcal{D}, y_j] dy_j - \mathbb{E}[u_{opt}|\mathcal{D}] \quad (67)$$

Finally, maximizing Equation (67) by differentiating it with respect to x^* , equating the obtained derivative to zero and solving the resulting equation or using any direct optimization method, we get the query point x^* of the highest value of the expected value of perfect information as indicated in the following equation.

$$x_{EVPI} = \arg \max_{x^*} \int_{y_j} P(y_j|\mathcal{D}, x^*) \mathbb{E}[u_{opt}|\mathcal{D}, y_j] dy_j - \mathbb{E}[u_{opt}|\mathcal{D}] \quad (68)$$

The expected value of perfect information method seems similar to the mean objective cost of uncertainty (MOCU) method proposed in Reference [31] and described in Section 2.5. Both methods can be viewed from decision theory perspective. The MOCU method seeks to minimize the expected regret which is the difference between the gained utility using the current model and the optimal utility. On the other hand, the EVPI method aims to maximize the difference between the optimal utility before and after acquiring a certain evidence. Accordingly, the MOCU method minimizes the deviation from the optimal decision. However, the EVPI method maximizes the expected utility improvement before and after acquiring a certain piece of information.

5.4. Balancing Exploration and Exploitation Strategies

This section describes several active learning strategies that seek to achieve the balance between exploration and exploitation.

5.4.1. Upper Confidence Bound (UCB)

The Upper Confidence Bound (UCB) strategy is proposed by Auer et al. in [36] in the context of multi-armed bandit problems [35]. We apply the UCB method as an active learning baseline strategy to compare with. The main advantage of this method is that it combines exploitation and exploration in a simple, yet an elegant way. The UCB strategy picks the unlabeled example maximizing the upper confidence bound of the random variable of interest, representing the utility function u .

$$x_{UCB} = \arg \max_{x^*} \mathbb{E}[u(x^*)|\mathcal{D}] + \eta\sigma_{u(x^*)|\mathcal{D}} \quad (69)$$

where u is the objective utility function to be maximized, $\mathbb{E}[u(x^*)|\mathcal{D}]$ and $\sigma_{u(x^*)|\mathcal{D}}$ denote the expected value and the standard deviation of the utility function for query point x^* given the training data acquired so far \mathcal{D} .

5.4.2. Probabilistic-Based Exploration-Exploitation (PEE)

This active learning strategy is originally inspired by simulated annealing [50]. More specifically, the probabilistic-based exploration-exploitation strategy is built on the ϵ -decreasing greedy algorithm [51]. In order to manage the trade-off between exploration and exploitation, this algorithm combines exploration and exploitation in a probabilistic way. With probability p_R , the exploration is performed via any exploratory strategy mentioned in Section 5.2 such as mutual information, Kullback–Leibler divergence and model entropy strategies. Furthermore, other exploration strategies in active learning literature can be incorporated into this method, such as uncertain sampling [4,15] and random sampling.

The exploration probability p_R is calculated as follows:

$$p_R = \alpha^{t-1} \quad (70)$$

where α is less than 1 and t is the current time step or iteration number. The exploration probability intuitively decays over time as seen in Equation (70) since the learning model gets to be more robust and capable of performing some exploitation to achieve the ultimate goal of utility maximization.

To implement this strategy, a uniform random variable Z is generated, if $Z \leq p_R$, any reasonable exploration strategy can be performed, otherwise pure exploitation is applied via maximizing the expected utility (the greedy strategy). However, any other exploitation strategy can be employed.

For the probabilistic-based exploration-exploitation strategy, we have implemented all of our proposed exploration based strategies in Section 5.2 in addition to uncertain sampling and random sampling. To perform exploitation, we use the greedy strategy since it is the simplest method. Although the greedy strategy is myopic since it does not account for enhancing the learning model estimate, in this PEE method the greedy strategy is integrated with an exploration strategy which already achieves an accurate model estimate.

5.4.3. Uncertainty of Strategy (UoS)

Similar to the probabilistic-based exploration-exploitation (PEE) strategy, this proposed active learning method seeks to balance the trade-off between exploration and exploitation in a probabilistic manner. Naturally, active learning querying strategies require a learning model estimate. Furthermore, many active learning strategies including: uncertain sampling [1] and greedy sampling, build their selection decisions entirely based on the learning model estimate. However, active querying methods that fully trust their estimate of the learning model and do not account for the learning model

uncertainty could probably yield inaccurate querying decisions. This argument motivates us to design a novel active learning method named Uncertainty of Strategy (UoS). The UoS method accounts for the inherent uncertainty of the querying criterion which is mainly caused by the model uncertainty or due to any other randomness in the active querying method.

The UoS strategy seeks to achieve the balance between exploitation and exploration. The exploitation can be easily performed using the current model estimate, for example, using greedy sampling. On the other hand, the exploration is done as follows: the UoS strategy sets a window of exploration around the active learning strategy's best estimate of a data point, which is returned by exploitation. The length of the exploration window can be estimated using the model uncertainty as described subsequently.

Let the query sample x_{UoS} follow a Gaussian distribution as follows:

$$x_{UoS} \sim \mathcal{N}(x_s, \sigma_s^2) \quad (71)$$

where the mean of this Gaussian distribution x_s represents the data point returned using pure exploitation. For the Gaussian distribution's variance σ_s^2 , it essentially depends on the model uncertainty. We estimate the strategy variance σ_s^2 using two different ways. The first method, named UoS-1 assumes that the strategy variance σ_s^2 is proportional to the model uncertainty, where the model uncertainty is estimated using the covariance matrix of the vector of model parameters β . Equation (72) defines the estimation of σ_s^2 in terms of the model uncertainty.

$$\sigma_s^2 = K \times \frac{b_{\sigma|\mathcal{D}}}{a_{\sigma|\mathcal{D}} - 1} \text{trace}[\Sigma_{\beta|\mathcal{D}}] \quad (72)$$

where K is a parameter set to adapt the units of the query point and the model parameters and to control the exploration/exploitation trade-off.

Like the PEE method, we set the K parameter to be time variant, in order to shrink the exploration window as iterations proceed since the model would become more reliable, so more emphasis should be devoted to exploitation.

$$K = Z^{t-1} \quad (73)$$

The second method, named UoS-2, estimates the strategy variance σ_s^2 empirically using a simple Monte Carlo simulation. This simulation runs for n iterations, where each iteration i proceeds as follows: first, an instance of model parameters vector β_i is generated according to the multivariate Student-T distribution using Equation (9). Then, this model parameters' instance β_i is used to evaluate the query point using a pure exploitation strategy x_{si} , this is, generally, a simple step as done in greedy strategy Equation (60) for example. Finally, after the n iterations finish, the strategy variance σ_s^2 is statistically evaluated as follows:

$$\sigma_s^2 = \frac{K}{n-1} \sum_{i=1}^n (x_{si} - \bar{x}_s)^2 \quad (74)$$

where K is a parameter for adapting units of the query point and model variance and for controlling the exploration-exploitation trade-off, akin to the UoS-1 method, K is defined in Equation (73). The expectation of strategy returned points \bar{x}_s is evaluated as the statistical mean over the n iterations as follows:

$$\bar{x}_s = \frac{1}{n} \sum_{i=1}^n x_{si} \quad (75)$$

The UoS-2 method is akin to the UoS-1 method for evaluating the strategy variance σ_s^2 defined in Equation (72) in since that it depends on the model uncertainty. However, this dependency is incorporated indirectly through the described Monte Carlo simulation. Algorithm 4 describes the UoS-2 method.

The proposed UoS active learning method, with its two variants, is general and can be combined with any exploitation-based strategy. Furthermore, the UoS method could be integrated with other popular active learning methods such as uncertain sampling [1] and expected model change [12], since most active learning strategies adhere to the greedy approach by querying a data point that maximizes or minimizes a certain selection criterion. In other words, the UoS querying approach could be used as a wrapper for any ordinary active learning method S that is greedy in its nature or does not consider the uncertainty of the learned model. This could be achieved by using S as the exploitation-based strategy used in the UoS method and adopting either of the two variants of the UoS method to estimate the strategy uncertainty.

Algorithm 4 The Uncertainty of Sampling Second Variant (UoS-2) Querying Method

Input: A dataset $\mathcal{D} = (x_i, y_i)$, an exploitation active learning strategy S , the number of simulation iterations n and a scaling parameter K .

Output: A query sample x^* .

Train the regression model using the training samples \mathcal{D} to obtain the mean $\mu_{\beta|\mathcal{D}}$ and the covariance $\Sigma_{\beta|\mathcal{D}}$ of the model parameters' vector β and the posterior estimates of $a_{\sigma|\mathcal{D}}$ and $b_{\sigma|\mathcal{D}}$.

for $i = 1$ to n **do**

Sample β_i from $t_{2a_{\sigma|\mathcal{D}}}(\mu_{\beta|\mathcal{D}}, \frac{b_{\sigma|\mathcal{D}}}{a_{\sigma|\mathcal{D}}} \Sigma_{\beta|\mathcal{D}})$.

$x_{si} \leftarrow$ the query sample returned after applying exploitation strategy S , using the sampled model parameters β_i .

end for

Evaluate the average query sample \bar{x}_s : $\bar{x}_s = \frac{1}{n} \sum_{i=1}^n x_{si}$

$\sigma_s^2 \leftarrow \frac{K}{n-1} \sum_{i=1}^n (x_{si} - \bar{x}_s)^2$.

$x^* \leftarrow$ Generate a query sample according to a Gaussian distribution as follows: $\mathcal{N}(\bar{x}_s, \sigma_s^2)$.

5.4.4. Utility minus Model Entropy (UME)

The Utility minus Model Entropy (UME) strategy controls the trade-off between exploration and exploitation in a novel way. The UME querying method adjusts the exploration and exploitation by explicitly modeling both of them in a formulated single objective function. Specifically, the UME method combines the ultimate goal of maximizing a certain utility function u , representing exploitation and the secondary but necessary target of minimizing model entropy, representing exploration, into one objective function. Then, the strategy queries the data sample x^* maximizing this hybrid objective as follows:

$$x_{UME} = \arg \max_{x^*} \mathbb{E}[u(x^*)|\mathcal{D}] - \eta H[\beta|x^*, \mathcal{D}] \tag{76}$$

where the model entropy $H[\beta|x^*, \mathcal{D}]$ is evaluated using Equation (58) and η is the exploration-exploitation trade-off control parameter. We conveniently let η be exponentially decreasing in time according to Equation (77). At early iterations, more emphasis is imposed on exploration to have better estimate for model parameters, however at later iterations since the model estimates get more robust over time, then more attention should be paid to the exploitation.

$$\eta = \eta_0 e^{-\alpha t} \tag{77}$$

where t is iteration number and $\alpha > 0$.

Substituting from Equation (58) into Equation (76) results in:

$$x_{UME} = \arg \max_{x^*} \mathbb{E}[u(x^*)|\mathcal{D}] - \frac{\eta}{2} \log \det(R_{\beta|\mathcal{D}\cup(x^*,y^*)}) + \phi(2a_{\mathcal{D}\cup(x^*,y^*)}, d) + (a_{\mathcal{D}\cup(x^*,y^*)} + \frac{d}{2}) \left((\Psi(a_{\mathcal{D}\cup(x^*,y^*)} + \frac{d}{2} + 1)) - \Psi(a_{\mathcal{D}\cup(x^*,y^*)}) \right) e^{-\frac{(a_{\sigma|\mathcal{D}\cup(x^*,y^*)} - 1)\mu_{\beta}^T \Sigma_{\mathcal{D}\cup(x^*,y^*)}^{-1} \mu_{\beta} + \beta_{\mathcal{D}\cup(x^*,y^*)}}{2^{b_{\sigma|\mathcal{D}\cup(x^*,y^*)}}} + 1}} \tag{78}$$

Since $\phi(2a^*, d)$ does not depend on the query sample x^* , then:

$$x_{UME} = \arg \max_{x^*} \mathbb{E}[u(x^*)|\mathcal{D}] - \frac{\eta}{2} \log \det(R_{\beta|\mathcal{D}\cup(x^*,y^*)}) (a_{\mathcal{D}\cup(x^*,y^*)} + \frac{d}{2}) \times \left((\Psi(a_{\mathcal{D}\cup(x^*,y^*)} + \frac{d}{2} + 1)) - \Psi(a_{\mathcal{D}\cup(x^*,y^*)}) \right) e^{-\frac{(a_{\sigma|\mathcal{D}\cup(x^*,y^*)} - 1)\mu_{\beta}^T \Sigma_{\mathcal{D}\cup(x^*,y^*)}^{-1} \mu_{\beta} + \beta_{\mathcal{D}\cup(x^*,y^*)}}{2^{b_{\sigma|\mathcal{D}\cup(x^*,y^*)}}} + 1}} \tag{79}$$

6. Case Study: Dynamic Pricing with Demand Learning

We apply the proposed active learning framework described in Section 5 to a real-world application which is dynamic pricing for revenue maximization in case of unknown behavior of the customers' demand.

The main challenge of dynamic pricing with unknown demand is that the chosen prices should achieve some balance between exploitation and exploration. Exploitation represents choosing prices aiming to maximize the achieved revenue. On the other hand, exploration selects prices that promote learning the demand model parameters. This motivates us to apply our proposed active learning framework in Figure 1 to this application.

We assume a linear demand elasticity for modeling the customers' demand behavior as typically used in the economics/finance literature (see Equation (80)). The price is the main controlling variable for demand. We assume a monopolist seller, who has a sufficient inventory to satisfy all potential demand and we, specifically, consider pricing a single product over a finite selling horizon T .

The linear demand model equation is defined as follows:

$$y = a + bp + \epsilon \tag{80}$$

such that $b < 0$ and $\epsilon \sim \mathcal{N}(0, \sigma^2)$.

The parameter b represents the price-demand sensitivity, so it is naturally negative since the price and demand have an inverse relationship. For example, if price rises by 10%, demand would diminish. On the other hand, when price decreases by 10%, demand would increase.

In order to estimate the demand model parameters a and b defined in Equation (80), we apply the Bayesian linear regression model described in Section 4. We employ the active learning framework with its different query generation schemes defined in Section 5.1 and described in detail in Algorithm 1, Algorithm 2 and Algorithm 3. Applying active learning formulation to the dynamic pricing problem, the training data \mathcal{D} consists of some pairs of prices and their corresponding demands (p_i, y_i) . In addition, the query point x^* denotes the vector $[1 \ p^*]$. For this application, the utility function after querying a certain price p represents the gained revenue R , which is defined as follows:

$$R = p(a + bp) \tag{81}$$

where a and b are the demand model parameters defined in Equation (80).

6.1. Active Learning Framework Application

In this section, we apply the active learning formulations represented in Section 5 to the dynamic pricing with demand learning problem. First, the exploration-based strategies hinge on minimizing

the regression model error, without considering the utility function u in their formulations. So, the formulations presented in Section 5.2 can be exactly used for the underlying dynamic pricing application. On the other hand, for the exploitation-based and the balancing strategies described in Section 5.3 and Section 5.4 respectively, specific formulations should be derived for the considered application, setting the utility function u to the gained revenue defined in Equation (81).

6.1.1. Exploration-Based Strategies

In our experiments, we apply the four presented strategies in Section 5.2 with pool-based and query synthesis schemes. Moreover, for mutual information, modified mutual information and Kullback–Leibler divergence, we implement the query synthesis approach without a predefined pool as described in Section 5.1 and Algorithm 3. When applying the query synthesis method without a predefined pool to the dynamic pricing problem, we construct U defined in Algorithm 3 as follows: since the dynamic pricing application has one controlling variable, the product price, we consider the range of all potential prices between p_{min} and p_{max} and along the active learning iterations, we exclude the prices that are previously queried, added to the training set \mathcal{D}_L . This set of prices P are used as unlabeled samples for evaluating the information-theoretic metrics as defined in (Equations (25), (41) and (47)).

6.1.2. Exploitation-Based Strategies

In this section, we apply the exploitation-based strategies introduced in Section 5.3 to the dynamic pricing with demand learning problem.

- Greedy Strategy

Given Equation (60) and setting the utility function u , to the gained revenue defined in Equation (81) results in:

$$p_G = \arg \max_{p^*} \mathbb{E}[R|p^*, \mathcal{D}] \quad (82)$$

Using the revenue definition in Equation (81), the expected revenue $E[R|p^*, \mathcal{D}]$ for any price p^* is evaluated using:

$$\mathbb{E}[R^*|p, \mathcal{D}] = p^*(x^{*T} \mu_{\beta|\mathcal{D}}) \quad (83)$$

where $x^* = [1 \ p^*]$.

We apply the greedy strategy in pool-based setting. In addition, we apply it in the query synthesis setting as well by maximizing the expected revenue as stated in Equation (82), using any optimization method or even using a simple grid search if the range of prices between p_{min} and p_{max} is limited.

By differentiating the expected revenue $E[R|p^*]$ w.r.t price p^* , the myopic price p_G maximizing the expected immediate revenue would be calculated as follows:

$$p_G = \frac{-\hat{a}|\mathcal{D}}{2\hat{b}|\mathcal{D}} \quad (84)$$

where $\hat{a}|\mathcal{D}$ and $\hat{b}|\mathcal{D}$ are the estimates of the demand model parameters a and b defined in Equation (80) using the labeled data gathered so far \mathcal{D} .

- Expected Value of Perfect Information (EVPI)

When applying the value of information strategy to the considered problem, the action α defined in Equation (61) is querying a price p^* . The initial evidence e represents the training labeled

data points so far \mathcal{D} . Similarly, e_j denotes the acquired demand y of the query price p^* , which represents the piece of information we seek to evaluate.

Accordingly, for the considered problem, the expected utility of taking action p^* given the evidence \mathcal{D} and after revealing evidence y_j , the expected utility term, $EU(\alpha_{e,k}|e, E_j = e_{jk})$, defined in Equation (65) can be formulated as:

$$\mathbb{E}[u(p_{y_j}^*|\mathcal{D}, y_j)] = \max_p R(p|\mathcal{D}, y_j) \tag{85}$$

where the utility u can be set to the immediate revenue R .

Using the linear demand model defined in Equation (80) and then applying the optimal price maximizing the immediate revenue defined in Equation (84) of the greedy strategy. Consequently, $\mathbb{E}[u(p_{y_j}^*|\mathcal{D}, y_j)]$ would be defined as follows:

$$\mathbb{E}[u(p_{y_j}^*|\mathcal{D}, y_j)] = \frac{-\hat{a}^2|\mathcal{D}, (x^*, y_j)}{4\hat{b}|\mathcal{D}, (x^*, y_j)} \tag{86}$$

The second term of Equation (61) could be safely ignored, since the objective is to experiment a price p^* , maximizing EVPI and this term is independent of p^* . This is implied by the following equation, Equation (87), it can be observed that this term does not affect the process of maximizing EVPI.

$$EU(p^*|\mathcal{D}) = \max_p R(p|\mathcal{D}, p^*) = \max_p R(p|\mathcal{D}) = \frac{-\hat{a}^2|\mathcal{D}}{4\hat{b}|\mathcal{D}} \tag{87}$$

Then, evaluating the EVPI method for revenue maximization problem, by substituting from Equation (86) into Equation (61) results in the following formula:

$$EVPI_{\mathcal{D}}(y^*) = \int_{y_j} P(y_j|\mathcal{D}, x^*) \frac{-\hat{a}^2|\mathcal{D}, (x^*, y_j)}{4\hat{b}|\mathcal{D}, (x^*, y_j)} dy_j \tag{88}$$

Finally, maximizing Equation (88) by differentiating it with respect to p^* , equating the derivative to zero and solving the resulting equation or using any direct optimization method, we get the price maximizing the expected value of perfect information as follows:

$$p_{EVPI} = \arg \max_x \int_{y_j} P(y_j|\mathcal{D}, x^*) \frac{-\hat{a}^2|\mathcal{D}, (x^*, y_j)}{4\hat{b}|\mathcal{D}, (x^*, y_j)} dy_j \tag{89}$$

6.1.3. Balancing Exploration and Exploitation Strategies

In this section, we consider applying the balancing strategies that combine both aspects of exploration and exploitation and attempt to achieve balance between both of them.

- Upper Confidence Bound (UCB)

Applying the UCB strategy to the dynamic pricing problem and setting the utility function u defined in Equation (69) to the immediate revenue R results in:

$$p_{UCB} = \arg \max_{p^*} \mathbb{E}[R|p^*, \mathcal{D}] + \eta \sigma_{R|p^*, \mathcal{D}} \tag{90}$$

where $\mathbb{E}[R|p^*, \mathcal{D}]$ and $\sigma_{R|p^*, \mathcal{D}}$ are the expectation and the standard deviation of the estimated immediate gained revenue R in response to price p^* and using training data labeled so far \mathcal{D} .

The expected revenue $\mathbb{E}[R^*|x^*, \mathcal{D}]$ is calculated as:

$$\mathbb{E}[R^*|p^*, \mathcal{D}] = p^* \mathbb{E}[y|x^*, \mathcal{D}] \tag{91}$$

where the expected demand $\mathbb{E}[y|x^*, \mathcal{D}]$ is computed using the Bayesian linear regression (see Equation (23)) presented in Section 4.

Accordingly:

$$\mathbb{E}[R^*|x^*, \mathcal{D}] = p^*(x^{*T} \mu_{\beta} | \mathcal{D}) \tag{92}$$

Using revenue definition in Equation (81) and the posterior variance for demand defined in Equation (24), the variance of revenue $\sigma_{R^*|x^*, \mathcal{D}}^2$ is calculated as follows:

$$\sigma_{R^*|x^*, \mathcal{D}}^2 = p^{*2} \frac{b_{\sigma|\mathcal{D}}}{a_{\sigma|\mathcal{D}} - 1} (1 + x^{*T} \Sigma_{\beta|\mathcal{D}} x^*) \tag{93}$$

Substituting from Equations (92) and (93) into Equation (90), then the price maximizing the UCB criterion can be evaluated as defined in Equation (94).

$$p_{UCB} = \arg \max_{p^*} p^*(x^{*T} \mu_{\beta|\mathcal{D}}) + \eta p^* \sqrt{\frac{b_{\sigma|\mathcal{D}}}{a_{\sigma|\mathcal{D}} - 1} (1 + x^{*T} \Sigma_{\beta|\mathcal{D}} x^*)} \tag{94}$$

where $\Sigma_{\beta|\mathcal{D}}$ is evaluated using Equation (6). For the Gamma distribution parameters $a_{\sigma|\mathcal{D}}$ and $b_{\sigma|\mathcal{D}}$, they are evaluated using Equations (7) and (8), respectively.

- Probabilistic-based Exploration-Exploitation (PEE)

In our experiments, we apply several instances of this hybrid strategy. We combine the pure exploitation, greedy, strategy as defined in Equation (84), with all the proposed exploration-based methods in addition to the popular active learning method, uncertain sampling [15] and we apply random sampling as a representative for random exploration.

- Uncertainty of Strategy (UoS)

For the uncertainty of strategy method, defined in Equation (71), the resulting price p_{UoS} follows a Gaussian distribution:

$$p_{UoS} \sim N(p_s, \sigma_s^2) \tag{95}$$

where the mean of this Gaussian distribution, p_s , represents the price returned using pure exploitation, which is the greedy strategy as defined in Equation (84). Regarding the variance of strategy σ_s^2 , it can be evaluated using two variants: in terms of model uncertainty and using Monte Carlo simulation as described in Section 5.4, specifically using Equations (72) and (74), respectively.

- Utility minus Model Entropy (UME)

The UME criterion as defined in Equation (76) is a function of the utility which is the immediate revenue and the model entropy. Consequently, substituting from Equations (31), (58) and (92) into Equation (76) results in the following equation:

$$p_{UME} = \arg \max_{p^*} p^*(x^{*T} \mu_{\beta} | \mathcal{D}) - \frac{\eta}{2} \log \det(R_{\beta|\mathcal{D}U(x^*, y^*)}) (a_{\mathcal{D}U(x^*, y^*)} + \frac{d}{2}) \times \left((\Psi(a_{\mathcal{D}U(x^*, y^*)} + \frac{d}{2} + 1)) - \Psi(a_{\mathcal{D}U(x^*, y^*)}) \right) e^{-\frac{(a_{\sigma|\mathcal{D}U(x^*, y^*)} - 1) \mu_{\beta}^T \mathcal{D}U(x^*, y^*) \Sigma_{\beta}^{-1} \mathcal{D}U(x^*, y^*) \mu_{\beta} \mathcal{D}U(x^*, y^*)}{2b_{\sigma|\mathcal{D}U(x^*, y^*)}} + 1} \tag{96}$$

where $x^* = [1 \ p^*]$ and d is the dimensionality which equals to 2 in the dynamic pricing application, with linear demand elasticity as defined in Equation (80). Therefore, by differentiating the objective function defined in Equation (96) and equating the resulting equation to zero, we can get the price maximizing the UME.

7. Experiments

7.1. Experimental Setup

In order to evaluate the performance of the proposed active learning framework summarized in Figure 1, as well as three baseline active learning methods including: random sampling (RS), the greedy or myopic strategy (check Section 5.3.1) and upper confidence bound method, which is intensively used in the multi armed bandit context [36] (see Section 5.4.1). In our experiments, we apply the proposed active learning framework to the dynamic pricing with demand learning problem described in Section 6. In the presented experiments, we mainly focus on analyzing the exploitation strategies introduced in Section 5.3 and the strategies balancing between exploitation and exploration presented in Section 5.4 since the main interest of the paper is applying active learning to utility optimization, which is exploitation.

In this work, we aim to perform a qualitative analysis to evaluate the performance of the pool-based approach versus the query synthesis approach since the query synthesis approach is computationally more efficient than the commonly adopted pool-based approach. Furthermore, the query synthesis approach could be more beneficial for objective optimization, such as maximizing revenue or even minimizing the learning model error, since it is not restricted to a certain given pool of samples. For mostly, all of the proposed active learning strategies including: our proposed strategies and the baseline methods, we implement two variants: one in pool-based setting and the other using query synthesis. In addition to these two active learning schemes, we further apply the third method, query synthesis without a predefined pool described in Section 5.1, to our proposed active learning methods that require the existence of a pool of unlabeled samples such as mutual information (MI), modified mutual Information (MMI) and Kullback–Leibler divergence (KL), in addition to the probabilistic-based exploration-exploitation (PEE) methods using either of MI, MMI or KL strategies for exploration.

In our experiments, we experiment different variants of the the probabilistic-based exploration-exploitation (PEE) approach combining greedy exploitation strategy with several exploration methods including our proposed approaches described in Section 5.2 in addition to random sampling and uncertain sampling [15]. The implemented PEE methods combining our proposed exploration methods presented in Section 5.2 with greedy sampling are denoted as KL-G, MI-G, MMMI-G and ME-G. In addition, we denote combining random and greedy sampling as (RS-G). Similarly, the method combining uncertain and greedy sampling is denoted as (US-G).

Some strategies such as the two variants of uncertainty of strategy (UoS) are basically designed so that the query point is generated or derived by optimizing an objective function. So, for these strategies we consider the query synthesis approach only since the pool-based approach does not apply for UoS.

In the adopted experiments, we apply the Bayesian linear regression model with conjugate prior of the model parameters β and σ , as described in Section 4, for estimating the demand at each iteration.

We conduct our experiments on synthetic and real datasets. The advantage of using artificial data is that the true model parameters $\beta = [a; b]$ are known. Accordingly, the ground truth value for the objective function, that is, the gained revenue, can be accurately computed with the knowledge of the true optimal model parameters as defined in Equation (81). Moreover, the estimation error of the model parameters β can be properly evaluated.

Evaluation Metrics

We assess the performance of our proposed active learning framework in terms of two aspects: the gained utility (revenue) and the accuracy of estimating the regression model parameters.

In order to evaluate the utility maximization, we measure the revenue gain or a normalized version of the total discounted utility u_T achieved in the considered time period as defined in Equation (97). We adopt discounted utilities to place more emphasis on getting rewards soon, as widely used in reinforcement learning [52].

$$u_{Gain} = \frac{\sum_{i=1}^T \gamma^{i-1} u_i}{\sum_{i=1}^T \gamma^{i-1} u_{opt}} \quad (97)$$

where u_i is the revenue obtained at iteration i and u_{opt} is the optimal revenue given the true model parameters a and b , which is calculated as:

$$u_{opt} = p_{opt}(a + bp_{opt}) \quad (98)$$

where p_{opt} is the optimal price, which equals to $\frac{-a}{2b}$ in case of linear demand model defined in Equation (80). Simplifying the term $\sum_{i=1}^T \gamma^{i-1}$ using the summation of geometric series formula, as follows:

$$u_{Gain} = \frac{\sum_{i=1}^T \gamma^{i-1} u_i}{(1 - \gamma^T)/(1 - \gamma) u_{opt}} \quad (99)$$

For the applications where the optimal utility is not known or cannot be evaluated, the total discounted utility can be used as an evaluation metric.

Concerning the demand model estimation error, we evaluate it in terms of the deviation of the final estimated demand model parameters Δ_β , from the true parameters β as indicated in Equation (100). The final model estimate is evaluated using the expectation of β given the training data $\mu_{\beta|\mathcal{D}}$, as defined in Section 4.

$$\Delta_\beta = \frac{\|\beta - \mu_{\beta|\mathcal{D}}\|_2}{\|\beta\|_2} \quad (100)$$

7.2. Experiments Using Synthetic Datasets

We perform a Monte Carlo simulation, generating 12 synthetic datasets of different parameters a , b and noise levels σ . We use two values for a , $a = 100$ and $a = 1000$. For each value for a , we adopt three different values for the sensitivity parameter b representing elastic demand ($b = -2$), neutral demand ($b = -1$) and inelastic demand ($b = -0.5$). Two different values are adopted for the noise parameter σ representing low (5%) and high (40%) noise levels. Investigating different noise levels enables us to analyze the impact of the noise on the different active learning strategies and evaluate their immunity towards noise. Moreover, for the dynamic pricing problem, we use different noise levels as a way for aggregating all other influencing factors that could affect the demand and may be hard to model, such as competition, seasonality or perishability of the products.

For each dataset, we run the experiment 10 runs and we present the average results over the runs. The synthetic datasets are created as follows: first, we generate $N = 1000$ price points from a Gaussian distribution with mean μ_p and variance σ_p^2 . Then, we assign values for demand elasticity parameters a and b . After that, assuming a linear demand model, we calculate the corresponding demands using Equation (80). We express the noise level parameter σ in terms of a percentage of the maximum possible demand a .

In our experiments, we set μ_p and σ_p of the Gaussian distribution used for generating the pricing data, using the pricing boundaries given by the seller p_{min} and p_{max} . For μ_p , it is the mean price of the prices in range of the $[p_{min}, p_{max}]$, which equals to $\frac{p_{min} + p_{max}}{2}$. Similarly, for the standard deviation σ_p , it is estimated using the standard deviation of the potential prices in the range of $[p_{min}, p_{max}]$. We set multivariate Gaussian prior for β as follows: $\mu = [10, -0.5]$, $\Sigma = 10^4 \mathbb{I}$. For the inverse Gamma prior distribution parameters of the noise parameter σ^2 , we set $a_\sigma = 2$ and $b_\sigma = 1$.

The simulation proceeds as follows: for each problem, we generate a pool of price-demand data points, starting with a very limited number of data points, $N_{init} = 3$ points, then we train a Bayesian regression model to obtain an initial estimate for the model parameters β . After that, we run the different exploitation and balancing active learning strategies described in Sections 5.3 and 5.4, respectively, with different schemes: pool-based, query synthesis and query synthesis without a predefined pool. For the query synthesis strategies (with and without pool variants), we assume that there is an oracle revealing the true demand value y^* for the chosen query point x^* . For each active learning strategy, we evaluate its performance by measuring the percentage revenue gain defined in Equation (99) and model estimation error defined in Equation (100).

Generally, most of the strategies balancing between exploration and exploitation have a hyper-parameter that controls the trade-off between exploration and exploitation. We set the controlling parameters of the balancing strategies introduced in Section 5.4 as follows: for the UCB method, the η parameter in Equation (69) is set to 0.01. For the PEE method, the α parameter in Equation (70) is set to 0.7. The K parameter of UoS strategy first variant, UoS-1, the parameter Z in Equation (73) is set to 0.5, while it is set to 0.7 for the second variant UoS-2. Finally, we set α parameter defined in Equation (77), such that at the last iteration T , where the exploration is nearly diminished, η equals to a small value: $\eta = 0.3$. Regarding the η_0 parameter of the same equation, Equation (77), we use values to let the impacts of the exploitation and exploration be comparable at the first iteration.

For the price-demand curve estimation problem, we enforce a constraint that the chosen price p^* at each iteration is within the pricing interval defined by the seller where the minimum allowable price is p_{min} and the maximum possible price p_{max} , accordingly $p_{min} \leq p^* \leq p_{max}$. The active learning loop continues till reaching a certain predefined number of iterations $T = 100$. For the pool-based strategies, the pool size $N = 1000$. We set the discount factor of revenue gained γ used in Equation (99) to 0.99.

The average results for the revenue gain and the regression model estimation error for different active learning strategies using different noise levels are represented in Tables 1 and 2, respectively.

Table 1. The average revenue gain of the active learning methods, over twelve synthetic datasets, using different noise levels σ . The strategies are sorted descendingly according to their average revenue gain over the two noise levels. The bold entries represent the maximum revenue gain per column (over all strategies).

Active Learning Strategy	$\sigma = 5\%$	$\sigma = 40\%$	Average over Noise levels
KL-G-Pool	99.17%	93.40%	96.28%
KL-G-Synth-Nopool	98.17%	90.65%	94.41%
UoS-1-Synth	99.64%	88.71%	94.18%
RS-G-Pool	98.02%	89.78%	93.90%
MMI-G-Synth-Nopool	99.31%	88.41%	93.86%
MMI-G-Synth	99.17%	88.50%	93.84%
MI-G-Synth-Nopool	99.19%	88.34%	93.77%
UME-Synth	99.60%	87.11%	93.36%
UoS-2-Synth	99.17%	87.43%	93.30%
RS-G-Synth	95.38%	90.49%	92.94%
G-Synth	99.49%	85.25%	92.37%
KL-G-Synth	98.85%	85.30%	92.08%
ME-G-Synth	99.22%	83.48%	91.35%
MI-G-Synth	99.41%	82.97%	91.19%
MI-G-Pool	99.11%	83.18%	91.14%
MMI-G-Pool	98.93%	80.17%	89.55%
ME-G-Pool	99.17%	79.74%	89.46%
EVPI-Synth	89.12%	86.33%	87.73%
EVPI-Pool	85.97%	87.86%	86.92%

Table 1. Cont.

Active Learning Strategy	$\sigma = 5\%$	$\sigma = 40\%$	Average over Noise levels
G-Pool	99.57%	68.28%	83.92%
UME-Pool	99.49%	68.13%	83.81%
US-G-Synth	85.15%	80.66%	82.91%
UCB-Pool	99.64%	64.19%	81.92%
UCB-Synth	99.48%	58.31%	78.90%
RS	77.61%	78.58%	78.09%
US-G-Pool	96.43%	57.67%	77.05%

Table 2. The average percentage model error of the active learning methods, over twelve synthetic datasets, using different noise levels σ . The strategies are sorted ascendingly according to their model estimation error over the two noise levels. The bold entries represent the minimum estimation error per column (over all strategies).

Active Learning Strategy	$\sigma = 5\%$	$\sigma = 40\%$	Average over Noise levels
RS	0.67%	8.74%	4.70%
RS-G-Synth	1.49%	10.35%	5.92%
EVPI-Pool	3.10%	9.46%	6.28%
RS-G-Pool	2.75%	10.10%	6.42%
EVPI-Synth	3.47%	10.49%	6.98%
MI-G-Pool	3.84%	11.11%	7.48%
KL-G-Synth-Nopool	2.61%	13.56%	8.09%
KL-G-Pool	2.88%	14.73%	8.81%
US-G-Synth	0.64%	17.64%	9.14%
UCB-Synth	4.38%	14.48%	9.43%
UoS-1-Synth	5.38%	15.20%	10.29%
KL-G-Synth	2.05%	18.90%	10.47%
UoS-2-Synth	6.02%	15.77%	10.90%
UME-Pool	2.63%	19.70%	11.17%
US-G-Pool	5.02%	17.53%	11.27%
G-Pool	3.64%	19.31%	11.48%
ME-G-Pool	5.14%	18.08%	11.61%
MI-G-Synth	3.42%	23.01%	13.21%
MMI-G-Pool	5.02%	22.76%	13.89%
MMI-G-Synth-Nopool	3.27%	25.01%	14.14%
MMI-G-Synth	4.11%	27.14%	15.62%
MI-G-Synth-Nopool	4.21%	27.54%	15.87%
UME-Synth	5.28%	26.66%	15.97%
G-Synth	5.24%	27.82%	16.53%
ME-G-Synth	6.02%	27.37%	16.69%
UCB-Pool	4.64%	30.13%	17.39%

One of the main contributions of this work is to perform a comparative analysis between the query synthesis and pool-based active learning approaches and demonstrate the benefits of applying active learning query synthesis based strategies for utility maximization. Accordingly, in the performed experiments, for each active learning, except random sampling since it is considered a passive learning method, we adopt a pool-based version and a query synthesis one. We provide an empirical analysis between both approaches. We include both aspects of the achieved revenue gain and model estimation error. Tables 3 and 4 represent the average revenue gain and the average model estimation error, respectively, for pool-based methods versus query synthesis ones.

Table 3. The average revenue gain for the pool-based versus query synthesis approaches, over twelve synthetic datasets, using different noise levels σ . The bold entries represent the maximum average revenue gain per row (over the two active learning approaches).

Dataset	Noise Level σ	Pool-Based	Query Synthesis
$a = 100, b = -0.5$	$\sigma = 5\%$	98.66%	98.36%
	$\sigma = 40\%$	80.44%	80.85%
$a = 100, b = -1$	$\sigma = 5\%$	96.41%	95.24%
	$\sigma = 40\%$	88.92%	90.86%
$a = 100, b = -2$	$\sigma = 5\%$	98.45%	98.47%
	$\sigma = 40\%$	83.80%	91.81%
$a = 1000, b = -0.5$	$\sigma = 5\%$	97.71%	97.99%
	$\sigma = 40\%$	56.24%	78.17%
$a = 1000, b = -1$	$\sigma = 5\%$	97.49%	96.25%
	$\sigma = 40\%$	89.12%	82.93%
$a = 1000, b = -2$	$\sigma = 5\%$	96.59%	97.84%
	$\sigma = 40\%$	64.92%	84.16%
Average	$\sigma = 5\%$	97.55%	97.36%
Average	$\sigma = 40\%$	77.24%	84.80%
Average		87.40%	91.08%

Table 4. The average percentage model error for the pool-based versus query synthesis approaches, over twelve synthetic datasets, using different noise levels σ . The bold entries represent the minimum average estimation error per row (over the two active learning approaches).

Dataset	Noise Level σ	Pool-Based	Query Synthesis
$a = 100, b = -0.5$	$\sigma = 5\%$	3.21%	4.60%
	$\sigma = 40\%$	12.59%	26.55%
$a = 100, b = -1$	$\sigma = 5\%$	2.79%	2.51%
	$\sigma = 40\%$	6.39%	9.01%
$a = 100, b = -2$	$\sigma = 5\%$	4.73%	2.63%
	$\sigma = 40\%$	16.58%	20.75%
$a = 1000, b = -0.5$	$\sigma = 5\%$	4.03%	2.68%
	$\sigma = 40\%$	14.33%	17.72%
$a = 1000, b = -1$	$\sigma = 5\%$	4.79%	3.97%
	$\sigma = 40\%$	31.24%	28.39%
$a = 1000, b = -2$	$\sigma = 5\%$	3.64%	6.66%
	$\sigma = 40\%$	22.61%	17.95%
Average	$\sigma = 5\%$	3.87%	3.84%
Average	$\sigma = 40\%$	17.29%	20.06%
Average		10.58%	11.95%

In addition, in order to investigate the superiority of either active learning approach, we evaluate the percentage of the pool-based strategies versus the query synthesis strategies existing in the top-10 performing methods in terms of achieving revenue gain, averaged over the different synthetic datasets, as presented in Table 5. Similarly, Table 6 shows the percentage of strategies from both approaches, pool-based and query synthesis, placed in the top-10 strategies achieving minimum regression model estimation error.

Table 5. The percentage of strategies in the top-10 strategies achieving revenue gain belonging to the pool-based approach versus the query synthesis approach, over twelve synthetic datasets, using different noise levels σ . The bold entries represent the maximum percentage per row (over the two active learning approaches).

Dataset	Noise Level σ	Pool-Based	Query Synthesis
$a = 100, b = -0.5$	$\sigma = 5\%$	50.00%	50.00%
	$\sigma = 40\%$	30.00%	70.00%
$a = 100, b = -1$	$\sigma = 5\%$	50.00%	50.00%
	$\sigma = 40\%$	40.00%	60.00%
$a = 100, b = -2$	$\sigma = 5\%$	50.00%	50.00%
	$\sigma = 40\%$	60.00%	40.00%
$a = 1000, b = -0.5$	$\sigma = 5\%$	50.00%	50.00%
	$\sigma = 40\%$	70.00%	30.00%
$a = 1000, b = -1$	$\sigma = 5\%$	60.00%	40.00%
	$\sigma = 40\%$	20.00%	80.00%
$a = 1000, b = -2$	$\sigma = 5\%$	50.00%	50.00%
	$\sigma = 40\%$	50.00%	50.00%
Average	$\sigma = 5\%$	51.67%	48.33%
Average	$\sigma = 40\%$	45.00%	55.00%
Average		48.33%	51.67%

Table 6. The percentage of strategies in the top-10 strategies achieving the least regression model error, belonging to the pool-based approach versus the query synthesis approach, over twelve synthetic datasets, using different noise levels σ . The bold entries represent the maximum percentage per row (over the two active learning approaches).

Dataset	Noise Level σ	Pool-Based	Query Synthesis
$a = 100, b = -0.5$	$\sigma = 5\%$	30.00%	70.00%
	$\sigma = 40\%$	20.00%	80.00%
$a = 100, b = -1$	$\sigma = 5\%$	50.00%	50.00%
	$\sigma = 40\%$	40.00%	60.00%
$a = 100, b = -2$	$\sigma = 5\%$	60.00%	40.00%
	$\sigma = 40\%$	40.00%	60.00%
$a = 1000, b = -0.5$	$\sigma = 5\%$	70.00%	30.00%
	$\sigma = 40\%$	20.00%	80.00%
$a = 1000, b = -1$	$\sigma = 5\%$	40.00%	60.00%
	$\sigma = 40\%$	40.00%	60.00%
$a = 1000, b = -2$	$\sigma = 5\%$	20.00%	80.00%
	$\sigma = 40\%$	60.00%	40.00%
Average	$\sigma = 5\%$	45.00%	55.00%
Average	$\sigma = 40\%$	36.67%	63.33%
Average		40.83%	59.17%

7.3. Experiments Using Real Datasets

To have the parameters more realistic, we have used several real datasets described in Table 7. We have gathered the first dataset in the table, transport, online through surveying. The dataset is a transportation ticket pricing data, where we ask users about the minimum and maximum fares they would pay for an economy class bus ticket of an air-conditioned bus between any general two cities, City A and City B, such that City A is away around 220 km from City B. We collected 41 responses from different users. In order to have data in the form of price and demand pairs, we perform the

following. For each price, we calculate the corresponding demand as the number of users who can afford this price according to the minimum and maximum prices of the data.

For beef dataset, it is obtained from the USDA Red Meats Yearbook [53]. The sugar dataset is adopted from Reference [54] and the spirits dataset is originated from Reference [55]. Finally, the coke dataset is adopted from Reference [56].

Table 7. A description for the real-world datasets.

Dataset	Size	\hat{a}	\hat{b}	$\hat{\sigma}$	Mean of Prices μ_p	Standard Deviation of Prices σ_p
Transport	41	41.3778	-0.1378	3.3902	145.00	90.8295
Beef	91	30.0515	-0.0465	0.5670	250.44	37.01
Sugar	18	1.3576	-0.3184	0.0292	1.005	0.0824
Spirits	69	4.4651	-1.2723	0.0573	2.1184	0.2089
Coke	20	50.5700	-0.3406	1.9319	22.96	3.2376

There is one hurdle in using such real datasets. In our proposed active learning framework, especially the query synthesis approach, the chosen data point or chosen price p^* could potentially be outside the available prices provided in the dataset. Thus, we utilize the dataset mainly for estimating linear demand model parameters vector β using ordinary least squares linear regression. Concerning the noise parameter σ^2 , we estimate it using the maximum likelihood estimator. The estimated model parameters \hat{a} , \hat{b} and $\hat{\sigma}$ for all the real datasets, are listed in Table 7. Using the obtained demand model parameters, we generate synthetic data using these parameters, with the same methodology described in Section 7.2. However, the mean and variance of the Gaussian distribution for generating pricing data μ_{u_p} and v_p , are estimated using the original prices of the real datasets.

Tables 8 and 9 represent the revenue gain and the estimation error of the regression model parameters, respectively, for the five considered real datasets described in Table 7.

Table 8. The revenue gain for the different active learning strategies using the five real datasets. The bold entries represent the maximum revenue gain per column (over all strategies).

Active Learning Strategy	Transport	Beef	Sugar	Spirits	Coke
G-Pool	98.53%	99.61%	79.17%	99.23%	63.49%
G-Synth	98.85%	99.81%	99.25%	99.08%	98.59%
EVPI-Pool	84.65%	94.28%	72.31%	94.32%	51.90%
EVPI-Synth	99.38%	87.95%	89.02%	86.11%	97.50%
UCB-Pool	98.94%	99.60%	79.17%	99.26%	63.49%
UCB-Synth	98.76%	99.77%	98.72%	99.44%	98.41%
MI-G-Pool	98.52%	99.55%	79.01%	99.43%	63.08%
MI-G-Synth	98.62%	99.04%	98.58%	99.26%	97.14%
MI-G-Synth-Nopool	98.75%	99.50%	98.73%	99.23%	97.27%
MMI-G-Pool	98.19%	99.57%	79.06%	99.40%	63.01%
MMI-G-Synth	98.51%	99.43%	98.51%	99.29%	96.82%
MMI-G-Synth-Nopool	98.16%	99.40%	98.37%	99.39%	97.19%
KL-G-Pool	98.70%	99.22%	78.79%	99.32%	63.05%
KL-G-Synth	99.10%	98.42%	99.00%	98.77%	99.13%
KL-G-Synth-Nopool	98.19%	98.66%	99.41%	98.62%	99.21%
ME-G-Pool	98.83%	99.58%	79.03%	98.86%	63.09%
ME-G-Synth	98.88%	99.47%	98.59%	99.35%	97.33%
US-G-Pool	95.53%	99.74%	79.15%	98.18%	63.46%
US-G-Synth	84.86%	86.34%	89.95%	93.42%	86.53%
RS-G-Pool	97.84%	98.94%	78.57%	99.28%	62.81%
RS-G-Synth	95.40%	97.00%	97.17%	96.22%	96.15%
UoS-1-Synth	98.61%	99.69%	99.31%	99.27%	98.70%
UoS-2-Synth	98.01%	99.70%	98.72%	99.50%	99.15%
UME-Synth	98.63%	99.68%	93.60%	99.15%	99.85%
UME-Pool	98.53%	99.53%	78.90%	99.14%	62.38%
RS	78.54%	93.45%	71.89%	94.12%	52.10%

Table 9. The percentage error of regression model parameters for the different active learning strategies for the real datasets. The bold entries represent the minimum estimation error per column (over all strategies).

Active Learning Strategy	Transport	Beef	Sugar	Spirits	Coke
G-Pool	7.19%	5.36%	4.56%	3.36%	1.71%
G-Synth	9.60%	3.36%	1.02%	6.50%	1.31%
EVPI-Pool	2.24%	2.26%	3.62%	0.43%	2.34%
EVPI-Synth	4.62%	0.52%	1.17%	5.78%	1.53%
UCB-Pool	6.14%	4.89%	4.56%	3.28%	1.95%
UCB-Synth	9.15%	3.47%	1.45%	4.36%	1.44%
MI-G-Pool	7.18%	4.60%	3.66%	3.22%	1.79%
MI-G-Synth	9.76%	1.87%	1.16%	3.98%	1.89%
MI-G-Synth-Nopool	9.34%	3.47%	1.41%	5.17%	1.07%
MMI-G-Pool	10.78%	4.69%	4.26%	2.77%	1.65%
MMI-G-Synth	11.45%	2.35%	1.09%	4.90%	1.01%
MMI-G-Synth-Nopool	11.81%	4.07%	1.11%	3.79%	1.71%
KL-G-Pool	3.95%	4.87%	3.37%	3.42%	1.94%
KL-G-Synth	4.16%	1.59%	1.91%	0.70%	1.57%
KL-G-Synth-Nopool	8.50%	2.21%	1.60%	2.23%	1.54%
ME-G-Pool	5.64%	4.44%	3.81%	4.14%	1.70%
ME-G-Synth	8.84%	3.42%	0.94%	3.41%	1.81%
US-G-Pool	10.23%	2.87%	4.12%	1.48%	1.86%
US-G-Synth	3.17%	0.82%	0.50%	0.77%	0.85%
RS-G-Pool	6.23%	2.50%	5.38%	1.70%	3.41%
RS-G-Synth	2.22%	1.02%	1.00%	0.92%	1.10%
UoS-1-Synth	10.38%	4.10%	1.35%	3.04%	2.24%
UoS-2-Synth	10.86%	3.39%	1.50%	3.57%	1.05%
UME-Synth	9.93%	3.30%	0.59%	4.64%	1.05%
UME-Pool	8.16%	4.53%	7.35%	3.14%	1.94%
RS	1.49%	1.46%	2.59%	1.43%	2.72%

Table 10 summarizes the average utility (revenue) gain and average model percentage error, averaged over the five real datasets described in Table 7, for all the considered active learning strategies.

Table 10. The average results of active learning methods in terms of the average revenue gain and average percentage error, over the five considered real datasets. The strategies are sorted descendingly according to their average obtained revenue gain. The bold entries for the first column represent the maximum average revenue gain over all strategies and the bold entries for the second column represent the minimum estimation error over all strategies.

Active Learning Strategy	Average Revenue Gain	Average Model Percentage Error
UoS-1-Synth	99.12%	4.22%
G-Synth	99.12%	4.36%
UCB-Synth	99.02%	3.97%
UoS-2-Synth	99.02%	4.07%
KL-G-Synth	98.88%	1.99%
KL-G-Synth-Nopool	98.82%	3.22%
ME-G-Synth	98.72%	3.68%
MI-G-Synth-Nopool	98.70%	4.09%
MI-G-Synth	98.53%	3.73%
MMI-G-Synth	98.51%	4.16%
MMI-G-Synth-Nopool	98.50%	4.50%
UME-Synth	98.18%	3.90%
RS-G-Synth	96.39%	1.25%
EVPI-Synth	91.99%	2.72%
US-G-Synth	88.22%	1.22%
UCB-Pool	88.09%	4.16%
G-Pool	88.01%	4.44%

Table 10. Cont.

Active Learning Strategy	Average Revenue Gain	Average Model Percentage Error
MI-G-Pool	87.92%	4.09%
ME-G-Pool	87.88%	3.95%
MMI-G-Pool	87.85%	4.83%
KL-G-Pool	87.82%	3.51%
UME-Pool	87.70%	5.02%
RS-G-Pool	87.49%	3.84%
US-G-Pool	87.21%	4.11%
EVPI-Pool	79.49%	2.18%
RS	78.02%	1.94%

We have experimented different values for the number of initial training points N_{init} in order to evaluate the impact of varying the number of initial training points on the performance of the different active learning methods. Tables 11 and 12 show the average revenue gain and model percentage error, respectively, averaged over the five considered real datasets described in Table 7. For space considerations, we include the results of this experiment for the real datasets only. The synthetic datasets exhibit a very similar behavior.

Table 11. The average revenue gain of active learning methods versus different number of initial training points N_{init} , averaged over the five considered real datasets. The bold entries represent the maximum average revenue gain per row (over the different number of initial training points N_{init}).

Active Learning Strategy	$N_{init} = 3$	$N_{init} = 5$	$N_{init} = 10$
G-Pool	88.01%	88.14%	88.14%
G-Synth	99.12%	99.46%	99.82%
EVPI-Pool	79.49%	80.87%	80.06%
EVPI-Synth	91.99%	94.81%	97.33%
UCB-Pool	88.09%	88.21%	88.19%
UCB-Synth	99.02%	99.47%	99.82%
MI-G-Pool	87.92%	87.68%	87.92%
MI-G-Synth	98.53%	98.85%	99.21%
MI-G-Synth-Nopool	98.70%	98.92%	98.96%
MMI-G-Pool	87.85%	87.93%	87.88%
MMI-G-Synth	98.51%	98.87%	99.14%
MMI-G-Synth-Nopool	98.50%	98.65%	99.14%
KL-G-Pool	87.82%	87.72%	87.65%
KL-G-Synth	98.88%	99.08%	98.47%
KL-G-Synth-Nopool	98.82%	98.40%	99.00%
ME-G-Pool	87.88%	88.11%	87.98%
ME-G-Synth	98.72%	98.95%	99.31%
US-G-Pool	87.21%	86.74%	87.52%
US-G-Synth	88.22%	86.88%	88.33%
RS-G-Pool	87.49%	87.64%	87.77%
RS-G-Synth	96.39%	95.76%	97.22%
UoS-1-Synth	99.12%	99.39%	99.76%
UoS-2-Synth	99.02%	99.12%	99.30%
UME-Synth	98.18%	98.13%	99.20%
UME-Pool	87.70%	87.79%	87.74%
RS	78.02%	79.15%	79.30%
Average	92.66%	92.87%	93.24%

Table 12. The average model percentage error of active learning methods versus different number of initial training points N_{init} , averaged over the five considered real datasets. The bold entries represent the minimum average estimation error per row (over the different number of initial training points N_{init}).

Active Learning Strategy	$N_{init} = 3$	$N_{init} = 5$	$N_{init} = 10$
G-Pool	4.44%	3.02%	1.72%
G-Synth	4.36%	2.99%	1.90%
EVPI-Pool	2.18%	1.35%	1.07%
EVPI-Synth	2.72%	1.94%	2.07%
UCB-Pool	4.16%	2.93%	1.83%
UCB-Synth	3.97%	2.56%	1.77%
MI-G-Pool	4.09%	2.37%	1.79%
MI-G-Synth	3.73%	2.11%	1.12%
MI-G-Synth-Nopool	4.09%	1.98%	1.00%
MMI-G-Pool	4.83%	3.16%	1.65%
MMI-G-Synth	4.16%	2.77%	0.67%
MMI-G-Synth-Nopool	4.50%	2.67%	2.12%
KL-G-Pool	3.51%	2.43%	2.76%
KL-G-Synth	1.99%	3.17%	2.37%
KL-G-Synth-Nopool	3.22%	2.13%	2.18%
ME-G-Pool	3.95%	2.56%	1.51%
ME-G-Synth	3.68%	2.78%	1.38%
US-G-Pool	4.11%	3.16%	1.70%
US-G-Synth	1.22%	1.47%	1.04%
RS-G-Pool	3.84%	1.83%	1.67%
RS-G-Synth	1.25%	1.53%	1.08%
UoS-1-Synth	4.22%	3.10%	1.80%
UoS-2-Synth	4.07%	2.61%	1.75%
UME-Synth	3.90%	3.48%	1.97%
UME-Pool	5.02%	3.91%	1.59%
RS	1.94%	1.46%	0.98%
Average	3.58%	2.52%	1.63%

Similar to the synthetic datasets, we compare both active learning approaches, pool-based and query synthesis over the five real datasets in terms of the revenue gain and the model estimation error. Thus, Tables 13 and 14 demonstrate the average revenue gain and average model estimation error for both approaches over the five real datasets presented in Table 7.

The percentage of pool-based strategies versus query synthesis strategies ranked within the top-10 strategies in terms of the revenue gain is presented in Table 15. Similarly, for model estimation error, Table 16 shows the percentage of strategies of both active learning approaches placed within the top-10 strategies achieving the least model estimation error.

Table 13. The average revenue gain for the pool-based approach versus query synthesis approach, using the five real datasets. The bold entries represent the maximum average revenue gain per row (over the two active learning approaches).

Dataset	Pool-Based Approach	Query Synthesis Approach
Transport	96.83%	97.51%
Beef	98.96%	97.59%
Sugar	78.32%	97.13%
Spirits	98.64%	97.74%
Coke	61.98%	97.26%
Average	86.94%	97.45%

Table 14. The percentage model error for the pool-based versus query synthesis strategies, using five the real datasets. The bold entries represent the minimum average estimation error per row (over the two active learning approaches).

Dataset	Pool-Based Approach	Query Synthesis Approach
Transport	6.77%	8.25%
Beef	4.10%	2.60%
Sugar	4.47%	1.19%
Spirits	2.69%	3.58%
Coke	2.03%	1.41%
Average	4.01%	3.41%

Table 15. The percentage of pool-based versus query synthesis strategies, ranked within the top-10 strategies achieving the highest revenue gain, using the five real datasets. The bold entries represent the maximum percentage per row (over the two active learning approaches).

Dataset	Pool-Based Approach	Query Synthesis Approach
Transport	30.00%	70.00%
Beef	50.00%	50.00%
Sugar	0.00%	100.00%
Spirits	40.00%	60.00%
Coke	0.00%	100.00%
Average	24.00%	76.00%

Table 16. The percentage of pool-based versus query synthesis strategies ranked within the top-10 strategies with respect to achieving the least percentage model error, using the five real datasets. The bold entries represent the maximum percentage per row (over the two active learning approaches).

Dataset	Pool-Based Approach	Query Synthesis Approach
Transport	60.00%	40.00%
Beef	30.00%	70.00%
Sugar	0.00%	100.00%
Spirits	50.00%	50.00%
Coke	0.00%	100.00%
Average	28.00%	72.00%

8. Discussion

In this section, we investigate the empirical results presented in Section 7. The main findings inferred from the experimental results are summarized as follows:

- It is evident from the presented results presented in Tables 1 and 10, that our proposed active learning strategies, especially the balancing methods, outperform the standard baselines: the upper confidence bound method, greedy sampling and random sampling, in terms of the achieved utility function (the revenue gain). There are several reasons for this compelling performance.

First, our proposed balancing methods attain the balance between exploration and exploitation using several novel approaches as described in Section 5.4. For example, for the proposed uncertainty of sampling method (UoS), it combines both aspects of utility maximization and regression model estimation in a probabilistic way, where the exploration is controlled using the model uncertainty.

In addition, the utility minus entropy (UME) method incorporates the model uncertainty, in addition to the utility function into one hybrid objective function to be optimized, as indicated in Equation (76). The explicit formulation of exploration in the active learning selection criterion

imposes an emphasis over the exploration in order to obtain accurate model estimation and hence achieve high future utility returns along the active learning iterations.

Finally, in the probabilistic-based exploration-exploitation method, we employ several powerful exploration methods, with the pure exploration method, the simple greedy sampling. The proposed exploration methods presented in Section 5.2, which are Kullback–Leibler divergence, mutual information and model entropy, have a great impact on estimating the regression model parameters, which indirectly helps boosting the gained utility (revenue).

- Table 1 shows the revenue gain for different artificial datasets, using different noise levels. It could be observed from this table that our proposed balancing strategies between exploration and exploitation such as the two variants of UoS, the four variants of PEE method combining the information theoretic exploration and pure exploitation: KL-G, MI-G and MMMI-G, ME-G and UME, show a significant revenue gain compared to the pure exploitation strategies such as greedy sampling and EVPI, especially for noisy datasets where $\sigma = 40\%$.
- Moreover, Table 1 indicates that the proposed balancing strategies outperform the baselines including: random sampling, greedy method and the UCB method. It can be observed that our proposed balancing strategies yield a substantial utility (revenue) gain in case of large noise 40%. For example, the KL-G strategy in both pool-based and synthetic settings, achieves around 2%–4% improvement, on average, over greedy sampling (GS). Furthermore, the KL-G method achieves 13%–15% improvement over the upper confidence bound (UCB) method [36] and around 16%–18% over random sampling RS.
- Table 2 demonstrates the estimation error of the regression model parameters, averaged over different artificial datasets for all the considered active learning strategies. Our proposed balancing strategies achieving high utility (revenue) gain such as KL-G in both pool and synthesis settings, UoS-1 and UoS-2, are not the best performing methods in terms of the model estimation error. However, these methods eventually yield a better model estimation than the baselines including: greedy sampling and UCB as indicated in Table 2. Furthermore, the main target is utility (revenue) optimization and the model estimation is a necessary but secondary objective. Moreover, the other proposed balancing strategies such as KL-G, MI-G and MMI-G have comparable performance to the baselines.
- For the real datasets, it can be observed from Table 10 that our proposed first variant of the uncertainty of strategy (UoS-1) is the best performing method in terms of the revenue gain. Although G-Synth has the same average revenue gain as UoS-1 method's gain, the UoS-1 method has lower estimation error rates than G-Synth. As mentioned in Section 5.4, the UoS-1 method accounts for the model uncertainty to control the exploration window (see Equation (72)). In addition to its promising performance, the UoS method is practically simple to implement.
- The two variants of our introduced balancing method, uncertainty of strategy (UoS), achieve significant performance in terms of the achieved revenue gain as indicated in Tables 1 and 10 for synthetic and real datasets, respectively. The major reason for the significant performance of the UoS method is that it accounts for the uncertainty of the selection criterion itself. Furthermore, this method combines the exploration and exploitation probabilistically like the UCB and the PEE methods.
- Regarding the proposed PEE methods: KL-G, MI-G, MMMI-G and ME-G, they produce substantial performance in terms of the achieved revenue gain for synthetic and real datasets, as shown in Tables 1 and 10, respectively. Specifically, the KL-G method is the best performing method in terms of the achieved revenue gain for the synthetic datasets (see Table 1).
- Furthermore, for the results of the real datasets presented in Table 10, the KL-G, MI-G, MMI-G and ME-G methods are of the top-10 strategies with respect to the achieved revenue gain. Moreover, for the model estimation error, they are comparable to the considered baselines. However, the KL-G variants provide competitive model estimation for both synthetic and real datasets as shown in Tables 1 and 10, respectively.

- There are three major reasons for the promising results of the PEE strategies: KL-G, MI-G, MMI-G and ME-G. First, these strategies are based on information theoretic concepts: Kullback–Leibler divergence [44] and entropy [43], as described in Section 5.2. Second, these methods adopt a probabilistic approach for balancing the exploration and exploitation as presented in Section 5.4, unlike the UCB method. The third reason is that the employed exploration strategies perform an effective exploration since they take into account the information of the unlabeled samples and the model uncertainty.
- In addition to our proposed strategies of the probabilistic-based exploration-exploitation (PEE) method, we have extended two more PEE methods combining uncertainty sampling [1] and random sampling, to perform exploration, with greedy sampling for exploitation. We experiment these two baselines for comparison purposes. For synthetic datasets, it could be noticed from Table 2 that the RS-G method with both versions, pool-based and query synthesis, achieves comparative model estimation. However, the RS-G method compromises the achieved revenue gain as indicated from Table 1, since it obtains a revenue gain that is around 3%–4% below the top method, KL-G-Pool. Similarly, for the real datasets' results presented in Table 10, the two methods US-G and RS-G obtain accurate model estimate, however both of these methods compromise the achieved revenue.
- These results essentially elucidate the significance of our proposed information-theoretic exploration strategies presented in Section 5.2. Although the same exploitation method is used, the greedy sampling and the same probabilistic approach, the PEE method, is followed for combining exploration and exploitation (see Section 5.4), the proposed methods, specifically, KL-G and MIMI-G, exhibit better performance than the US-G and RS-G methods in terms of the achieved revenue, which is the main target.

Furthermore, our proposed methods obtain model estimation performance close to the US-G and RS-G methods, for both synthetic and real datasets as presented in Tables 1 and 10, respectively. The other two proposed methods MI-G and ME-G, also outperform the RS-G and US-G methods, for real datasets and produce comparable performance for the synthetic datasets.

The reason for the performance preeminence of our proposed information theory-based strategies over the US-G and RS-G methods in terms of revenue gain is that the proposed methods essentially exploit the potential information of the unlabeled data and the model uncertainty. Moreover, these strategies not only improve the model estimation error but also query representative data samples that minimize the model uncertainty, which promotes the exploitation performance.

- From Table 10, it can be noticed that the greedy sampling performs well in the real datasets, since the considered real datasets have very low noise, expressed in terms of $\hat{\sigma}$ in Table 7. Also, the UCB baseline [36] performs comparably well on the real datasets due to the datasets' robustness. Similarly for artificial datasets, Table 1 shows that for the low noise datasets, having $\sigma = 5\%$, both of the greedy sampling and the UCB methods perform quite well, comparable to the best performing method, our proposed balancing method UoS-1. On the other hand, for the noisy datasets where $\sigma = 40\%$, both of the greedy sampling and the UCB methods result in poor performance, in terms of the gained revenue. This could be apparently observed from Table 1.
- Regarding the other developed pure exploitation method, namely the expected value of information (EVPI), it could be noticed from Tables 2 and 10, the EVPI strategy results in better model estimation than the greedy sampling as EVPI chosen samples incur some diversity unlike the points chosen by the greedy method which essentially queries points maximizing the utility function. However, the greedy sampling, adequately, outperforms EVPI in case of low noise and for real datasets (see Tables 1 and 10, respectively). On the other hand, for the noisy datasets, the EVPI approach, in both settings pool-based and query synthesis, attains larger revenue gain than the corresponding methods for greedy sampling as shown in Table 1 since the EVPI method is less myopic than the greedy sampling, so it is more immune to the noisy datasets.
- Concerning the random sampling (RS) baseline method, since it is a pure exploration strategy, convincingly, it does not achieve high revenue gains for synthetic and real datasets as indicated

in Tables 1 and 10, respectively. However, since random sampling could be regarded as a pure exploration methods, it, intuitively, performs well with respect to the model estimation error as shown in Tables 1 and 10, for synthetic and real datasets, respectively. However, as previously mentioned, random sampling considerably jeopardizes the gained revenue.

- In this work, we perform a comparative empirical analysis between the pool-based and query synthesis active learning approaches. The empirical analysis considers both evaluation metrics, the achieved gained utility (revenue) and the percentage regression model error. As we mentioned in Section 7, we exclude the random sampling from this analysis since it is a passive learning method.
- Concerning the revenue gain, Tables 3 and 13 demonstrate the average revenue for the strategies of each approach, for synthetic and real datasets, respectively. It could be observed that the query synthesis approach clearly outperforms the pool-based approach for both the artificial (with improvement around 3.5%) and real datasets (with improvement around 10%). The improvement is more significant in case of real datasets as will be discussed subsequently. Moreover, as indicated in Table 15, the query synthesis approach is more dominant within the top-10 strategies in terms of achieving revenue gain, for real datasets. Specifically, 76% of the top-10 strategies achieving revenue gain, belong to the query synthesis approach, whereby only 24% strategies are pool-based methods.
- Regarding the low noise synthetic data, both of pool-based and query synthesis approaches result in similar performance in terms of average revenue gain as shown in Table 3.
- On the other hand, in case of large noise level, it could be inferred from Table 3 that the query synthesis approach surpasses the pool-based one. In other words, the revenue gain improvement of the query synthesis over the pool-based approach is around 7.5%. In addition, the query synthesis methods have higher ranks than the pool-based ones in terms of the average revenue gain. Specifically, the ratio between the former and the latter is 55% to 45%, respectively. These results are persuasive since for noisy data, the pool of samples could be misleading, so querying a synthetic data sample in the global input space, that is not necessarily belonging to a specific set of data samples, would be more effective for optimizing a certain utility function.
- Regarding the model percentage error, the pool-based approach produces a slightly better model estimate for artificial datasets (see Table 4). However, the query synthesis methods have more advanced ranks than the pool-based methods as presented in Table 6, for both of the low and high noise levels.
- By examining the real datasets results, we can find that the query synthesis approach accomplishes less error rates than the pool-based approach as shown in Table 14. In addition, when investigating the top-10 strategies in terms of minimizing the model estimation errors in Table 16, one could observe that the query synthesis methods occupy 72% of the top-10 strategies, compared to 24% for the pool-based methods.
- The two real datasets, namely the sugar and coke datasets, described in Table 7 represent typical cases where the pool-based approach suffers from a major performance hurdle in terms of the obtained revenue gain as presented in Table 13. The reason for the poor performance of the pool based approach for these two datasets is that the available pool of data samples is limited and not representative enough. Furthermore, the available data samples do not contain the optimal price maximizing the target utility function, which is the gained revenue.
- For example, for the coke dataset, according to the linear demand model parameters presented in Table 7 and using the revenue equation (Equation (81)), the optimal price maximizing revenue is $\frac{-\hat{a}}{2\hat{b}} = 74.24$. However, the mean and the standard deviation, of the available prices of this dataset are $\mu_p = 22.96$ and $\sigma_p = 3.2376$, respectively. Accordingly, the available prices of the pool are too far from optimal, that is why the pool-based strategies do not perform well on this dataset. The sugar dataset exhibits a very similar behavior as well. The lack of diversity in the pool is considered a serious drawback for the pool-based approach. In contrast, the query synthesis approach is not affected by such problem since the query synthesis approach chooses the data

sample to be labeled from the entire input space and it is not restricted by the available pool of data (see Tables 13 and 15).

- One could infer from Tables 1 and 10 that the two variants of query synthesis approach with a predefined pool (see Algorithm 2) and without a predefined pool (see Algorithm 3), yield comparable performance for the different strategies. This is reasonable since both methods are logically equivalent, they only differ in implementation details. The query synthesis without a predefined pool approach, defined in Section 5.1 is essentially designed for the applications where it could be complicated to have a pool of representative data samples to be used for the information theory-based active learning strategies, namely KL, MI and MMI. In our experiments, we utilize the domain knowledge of the pricing data and construct a set of data samples belonging to the price range defined by the seller $[p_{min}, p_{max}]$ to be used by the information theoretic methods: KL, MI and MMI.
- Finally, Tables 11 and 12 show that increasing the initial training points enhances the performance of most of the active learning methods, in terms of both the revenue gain and the model estimation accuracy. These results are reasonable since having more initial data samples promotes the regression model's accuracy, so the gained revenue is improved as well. In addition, Table 12 indicates that as the number of initial training samples increases, the performance of the different active learning methods gets closer to each other since the initial model estimate gets more robust. For the revenue gain, the query synthesis methods achieve similar performance as indicated in Table 11 for $N_{init} = 10$. However, most of the pool-based methods do not achieve a significant performance improvement due to the limitation of the pool-based approach, previously discussed, for some real datasets, the sugar and coke datasets.

9. Conclusions

In this paper, we propose a novel active learning framework for optimizing a general utility function. Specifically, this work targets the class of problems incurring some trade-off between exploration and exploitation. We introduce several novel active learning methods for exploration, exploitation and for balancing both. The presented exploration strategies are essentially based on information theory concepts such as mutual information (MI), Kullback–Leibler divergence (KL) and model entropy (ME). Consequently, when combined with exploitation, such information theoretic exploration methods achieve promising performance in terms of the achieved utility and the learning model error as well. Furthermore, we develop new approaches for balancing exploration and exploitation such as the uncertainty of strategy (UoS) method that controls the exploration window according to the model uncertainty. In addition, we present another balancing method, utility minus entropy (UME) where the model entropy is explicitly modeled and augmented with the target utility function into one hybrid objective function to be optimized.

In this work, we investigate two main approaches of active learning, the pool-based approach which is widely used in active learning literature and the membership query synthesis approach. Moreover, we present an empirical analysis for comparing both approaches. The experiments show the exceptional performance of the query synthesis approach compared to the pool-based approach for the synthetic and real datasets. The compelling results for query synthesis approach could help boosting the active learning research towards employing the query synthesis approach.

We have applied the proposed framework to an operation research related application, namely, dynamic pricing with demand learning. However, our proposed framework can easily be adapted to other applications. We perform several experiments using synthetic and real datasets. In our experiments, we compare our proposed active learning strategies to several baselines and our presented strategies yield a significant performance improvement in terms of both aspects: the achieved gained revenue and the regression model error.

Author Contributions: Conceptualization, D.E., A.F.A. and S.I.S.; Formal analysis, D.E.; Methodology, D.E. and A.F.A.; Project administration, A.F.A. and S.I.S.; Resources, A.F.A. and S.I.S.; Software, D.E.; Supervision, A.F.A. and S.I.S.; Validation, D.E. and A.F.A.; Writing—original draft, D.E.; Writing—review & editing, A.F.A. and S.I.S.

Funding: This research received no external funding.

Conflicts of Interest: The authors of this work declare no conflict of interest.

References

- Cohn, D.; Atlas, L.; Ladner, R. Improving generalization with active learning. *Mach. Learn.* **1994**, *15*, 201–221. [[CrossRef](#)]
- Olsson, F. *A Literature Survey of Active Machine Learning in the Context of Natural Language Processing*; Technical Report; Swedish Institute of Computer Science: Kista, Sweden, 2009.
- Tong, S.; Chang, E. Support vector machine active learning for image retrieval. In Proceedings of the Ninth ACM International Conference on Multimedia, Ottawa, ON, Canada, 30 September–5 October 2001; pp. 107–118.
- Settles, B. *Active Learning Literature Survey*; Technical Report; University of Wisconsin-Madison Department of Computer Sciences: Madison, WI, USA, 2009.
- Kaelbling, L.P.; Littman, M.L.; Moore, A.W. Reinforcement learning: A survey. *J. Artif. Intell. Res.* **1996**, *4*, 237–285. [[CrossRef](#)]
- Črepinšek, M.; Liu, S.H.; Mernik, M. Exploration and exploitation in evolutionary algorithms: A survey. *ACM Comput. Surv. (CSUR)* **2013**, *45*, 35. [[CrossRef](#)]
- Ten Hagen, S.; Van Someren, M.; Hollink, V. Exploration/exploitation in adaptive recommender systems. In Proceedings of the European Symposium on Intelligent Technologies, Hybrid Systems and their implementation on Smart Adaptive Systems, Oulu, Finland, 10–12 July 2003.
- den Boer, A.V. Dynamic pricing and learning: historical origins, current research, and new directions. *Surv. Oper. Res. Manag. Sci.* **2015**, *20*, 1–18. [[CrossRef](#)]
- Lobo, M.S.; Boyd, S. Pricing and learning with uncertain demand. In Proceedings of the INFORMS Revenue Management, New York, NY, USA, 5–6 June 2003.
- Ibrahim, M.N.; Atiya, A.F. Analytical solutions to the dynamic pricing problem for time-normalized revenue. *Eur. J. Oper. Res.* **2016**, *254*, 632–643. [[CrossRef](#)]
- Elreedy, D.; Atiya, A.F.; Fayed, H.; Saleh, M. A framework for an agent-based dynamic pricing for broadband wireless price rate plans. *J. Simul.* **2019**, *13*, 1–15. [[CrossRef](#)]
- Cai, W.; Zhang, Y.; Zhou, J. Maximizing expected model change for active learning in regression. In Proceedings of the IEEE 13th International Conference on Data Mining, Dallas, TX, USA, 7–10 December 2013, pp. 51–60.
- Burbidge, R.; Rowland, J.J.; King, R.D. Active learning for regression based on query by committee. In *Proceedings of the International Conference on Intelligent Data Engineering and Automated Learning*; Springer: Berlin, Germany, 2007, pp. 209–218.
- Wu, D. Pool-Based Sequential Active Learning for Regression. *arXiv* **2018**, arXiv:1805.04735.
- Lewis, D.; Gale, W. A Sequential Algorithm for Training Text Classifiers. In Proceedings of the 17th International ACM SIGIR Conference on Research and Development in Information Retrieval, Dublin, Ireland, 3–6 July 1994.
- Cohn, D.A.; Ghahramani, Z.; Jordan, M.I. Active learning with statistical models. *J. Artif. Intell. Res.* **1996**, *4*, 129–145. [[CrossRef](#)]
- Guo, Y.; Greiner, R. Optimistic Active-Learning Using Mutual Information. In Proceedings of the IJCAI, Hyderabad, India, 6–12 January 2007; Volume 7, pp. 823–829.
- Freund, Y.; Seung, H.S.; Shamir, E.; Tishby, N. Selective sampling using the query by committee algorithm. *Mach. Learn.* **1997**, *28*, 133–168. [[CrossRef](#)]
- Roy, N.; McCallum, A. Toward optimal active learning through monte carlo estimation of error reduction. In Proceedings of the ICML, Williamstown, MA, USA, 28 June–1 July 2001; pp. 441–448.
- Wu, D.; Lin, C.T.; Huang, J. Active learning for regression using greedy sampling. *Inf. Sci.* **2019**, *474*, 90–105. [[CrossRef](#)]

21. Sugiyama, M. Active learning in approximately linear regression based on conditional expectation of generalization error. *J. Mach. Learn. Res.* **2006**, *7*, 141–166.
22. Holub, A.; Perona, P.; Burl, M.C. Entropy-based active learning for object recognition. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, Anchorage, AK, USA, 23–28 June 2008; pp. 1–8.
23. Sourati, J.; Akcakaya, M.; Dy, J.G.; Leen, T.K.; Erdogmus, D. Classification active learning based on mutual information. *Entropy* **2016**, *18*, 51. [[CrossRef](#)]
24. Sourati, J.; Akcakaya, M.; Erdogmus, D.; Leen, T.K.; Dy, J.G. A probabilistic active learning algorithm based on fisher information ratio. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 2023–2029. [[CrossRef](#)] [[PubMed](#)]
25. Alabdulmohsin, I.M.; Gao, X.; Zhang, X. Efficient Active Learning of Halfspaces via Query Synthesis. In Proceedings of the AAAI Twenty-Ninth AAAI Conference on Artificial Intelligence, Hyatt Regency, Austin, 25–29 January 2015; pp. 2483–2489.
26. Angluin, D. Queries and concept learning. *Mach. Learn.* **1988**, *2*, 319–342. [[CrossRef](#)]
27. Saar-Tsechansky, M.; Provost, F. Decision-centric active learning of binary-outcome models. *Inf. Syst. Res.* **2007**, *18*, 4–22. [[CrossRef](#)]
28. Garnett, R.; Krishnamurthy, Y.; Xiong, X.; Schneider, J.; Mann, R. Bayesian optimal active search and surveying. *arXiv* **2012**, arXiv:1206.6406.
29. Zuluaga, M.; Krause, A.; Püschel, M. ϵ -pal: An active learning approach to the multi-objective optimization problem. *J. Mach. Learn. Res.* **2016**, *17*, 3619–3650.
30. Frazier, P.I.; Powell, W.B.; Dayanik, S. A knowledge-gradient policy for sequential information collection. *SIAM J. Control Optim.* **2008**, *47*, 2410–2439. [[CrossRef](#)]
31. Yoon, B.J.; Qian, X.; Dougherty, E.R. Quantifying the objective cost of uncertainty in complex dynamical systems. *IEEE Trans. Signal Process.* **2013**, *61*, 2256–2266. [[CrossRef](#)]
32. Dehghannasiri, R.; Yoon, B.J.; Dougherty, E.R. Optimal experimental design for gene regulatory networks in the presence of uncertainty. *IEEE/ACM Trans. Comput. Biol. Bioinform. (TCBB)* **2015**, *12*, 938–950. [[CrossRef](#)]
33. Vermorel, J.; Mohri, M. Multi-armed bandit algorithms and empirical evaluation. In *Proceedings of the European Conference on Machine Learning*; Springer: Berlin, Germany, 2005; pp. 437–448.
34. Krause, A.; Guestrin, C. Nonmyopic active learning of gaussian processes: An exploration-exploitation approach. In Proceedings of the 24th International Conference on Machine Learning, Corvallis, OR, USA, 20–24 June 2007; pp. 449–456.
35. Robbins, H. Some aspects of the sequential design of experiments. In *Herbert Robbins Selected Papers*; Springer: Berlin, Germany, 1985; pp. 169–177.
36. Auer, P. Using confidence bounds for exploitation-exploration trade-offs. *J. Mach. Learn. Res.* **2002**, *3*, 397–422.
37. Ben Taieb, S.; Atiya, A.F. A bias and variance analysis for multistep-ahead time series forecasting. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *27*, 62–76. [[CrossRef](#)] [[PubMed](#)]
38. Fumo, N.; Biswas, M.R. Regression analysis for prediction of residential energy consumption. *Renew. Sustain. Energy Rev.* **2015**, *47*, 332–343. [[CrossRef](#)]
39. Bayoumi, A.E.M.; Saleh, M.; Atiya, A.F.; Aziz, H.A. Dynamic pricing for hotel revenue management using price multipliers. *J. Revenue Pric. Manag.* **2013**, *12*, 271–285. [[CrossRef](#)]
40. Klauenberg, K.; Wübbeler, G.; Mickan, B.; Harris, P.; Elster, C. A tutorial on bayesian normal linear regression. *Metrologia* **2015**, *52*, 878. [[CrossRef](#)]
41. Kotz, S.; Nadarajah, S. *Multivariate T-Distributions and Their Applications*; Cambridge University Press: Cambridge, UK, 2004.
42. O’Hagan, A.; Forster, J.J. Kendall’s Advanced Theory of Statistics, Volume 2B: Bayesian Inference. Available online: <https://eprints.soton.ac.uk/46376/> (accessed on 28 June 2019).
43. Shannon, C.E. A mathematical theory of communication. *Bell Syst. Tech. J.* **1948**, *27*, 379–423. [[CrossRef](#)]
44. Kullback, S. Kullback–Leibler distance. *Am. Stat.* **1987**, *41*, 340–341.
45. Lopez Quintero, F.O.; Contreras-Reyes, J.E.; Wiff, R.; Arellano-Valle, R.B. Flexible Bayesian analysis of the von Bertalanffy growth function with the use of a log-skew-t distribution. *Fish. Bull.* **2017**, *115*, 13–26. [[CrossRef](#)]
46. Baez, J.; Pollard, B. Relative entropy in biological systems. *Entropy* **2016**, *18*, 46. [[CrossRef](#)]
47. Mata, M.; Machado, J. Entropy Analysis of Monetary Unions. *Entropy* **2017**, *19*, 245. [[CrossRef](#)]

48. Kulick, J.; Lieck, R.; Toussaint, M. Active Learning of Hyperparameters: An Expected Cross Entropy Criterion for Active Model Selection. Available online: <https://ipvs.informatik.uni-stuttgart.de/mlr/papers/14-kulick-maxce.pdf> (accessed on 26 September 2014).
49. Russell, S.J.; Norvig, P. *Artificial Intelligence: A Modern Approach*; Pearson Education Limited: Kuala Lumpur, Malaysia, 2016.
50. Van Laarhoven, P.J.; Aarts, E.H. Simulated annealing. In *Simulated Annealing: Theory and Applications*; Springer: Berlin, Germany, 1987; pp. 7–15.
51. Kuleshov, V.; Precup, D. Algorithms for multi-armed bandit problems. *arXiv preprint* **2014**, arXiv:1402.6028.
52. Sutton, R.S.; Barto, A.G. *Introduction to Reinforcement Learning*; MIT Press: Cambridge, UK, 1998; Volume 135.
53. MUSDA-ERS Electronic Data Archive, Red Meats Yearbook, housed at Cornell University's Mann Library. Available online: <http://usda.mannlib.cornell.edu/> (accessed on 25 September 2001).
54. Schultz, H. A comparison of elasticities of demand obtained by different methods. *Econ. J. Econ. Soc.* **1933**, *1*, 274–308. [[CrossRef](#)]
55. Durbin, J.; Watson, G.S. Testing for serial correlation in least squares regression: I. *Biometrika* **1950**, *37*, 409–428. [[PubMed](#)]
56. Sun, Y. Coke Demand Estimation Dataset. Available online: leeds-faculty.colorado.edu/ysun/doc/Demand_estimation_worksheet.doc (accessed on 23 February 2010).



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).