Bayesian Active Learning by Soft Mean Objective Cost of Uncertainty

Guang Zhao[1], **Edward R. Dougherty**[1], **Byung-Jun Yoon**[1,3], **Francis J. Alexander**[3], **Xiaoning Qian**[1,2]
[1]Electrical & Computer Engineering, [2]Computer Science & Engineering, Texas A&M University
[3]Brookhaven National Laboratory

Abstract

To achieve label efficiency for training supervised learning models, pool-based active learning sequentially selects samples from a set of candidates as queries to label by optimizing an acquisition function. One category of existing methods adopts one-step-look-ahead strategies based on acquisition functions tailored with the learning objectives, for example based on the expected loss reduction (ELR) or the mean objective cost of uncertainty (MOCU) proposed recently. These active learning methods are optimal with the maximum classification error reduction when one considers a single query. However, it is well-known that there is no performance guarantee in the long run for these myopic methods. In this paper, we show that these methods are not guaranteed to converge to the optimal classifier of the true model because MOCU is not strictly concave. Moreover, we suggest a strictly concave approximation of MOCU—*Soft MOCU*—that can be used to define an acquisition function to guide Bayesian active learning with theoretical convergence guarantee. For training Bayesian classifiers with both synthetic and real-world data, our experiments demonstrate the superior performance of active learning by Soft MOCU compared to other existing methods.

Proceedings of the 24[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130.

1 INTRODUCTION

Active learning has been one of effective learning strategies for training supervised learning models when collecting or labeling data is difficult or expensive (Gal et al., 2017; Tran et al., 2019; Sinha et al., 2019). Active learning methods sequentially collect data in the input feature space and acquire their corresponding labels to improve model predictions based on different objective functions. The goal is to derive generalizable supervised models with less labeled data compared to the traditional blind training data collection approach that does not explicitly consider the cost incurred by collecting or labeling data.

In this paper, we focus on learning optimal Bayesian classifiers with limited training data. To achieve sample and label efficiency, we study pool-based Bayesian active learning. It starts with a prior of an uncertain model and collects training data in a sequential manner by optimizing an acquisition function measuring the benefit to our learning objective from querying labels for corresponding candidates. By reducing model uncertainty through the active learning procedure, we aim to approach the optimal classifier of the unknown true model, which has the minimum prediction error.

Several notable Bayesian active learning methods have been proposed using different acquisition functions. Maximum Entropy Search (MES) or Uncertainty Sampling selects the candidate with the maximum predictive probability entropy (Sebastiani and Wynn, 2000; Mussmann and Liang, 2018). However, observing the most uncertain candidate may not provide the most useful model information if the observation itself is noisy. Another Shannon entropy based method, Bayesian Active Learning by Disagreement (BALD) (Houlsby et al., 2011; Kirsch et al., 2019), selects the candidate to minimize the entropy of the uncertain model parameters. The Equivalence Class Edge Cutting algorithm (EC^2) targeting at active learning with finite possible models, chooses the candidate that maximally reduces the version space

probability mass (Golovin et al., 2010). Based on the policy Gibbs error, a generalization of the Shannon entropy, Cuong et al. (2013) proposed the maximum Gibbs error criterion (maxGEC) to query the candidate that has the maximum Gibbs error so that the remaining posterior entropy is minimized. While various different acquisition functions are used, most of the existing active learning methods focus on reducing the model uncertainty instead of directly reducing the classification error, despite this being the ultimate learning objective. To rectify this shortcoming, in this paper, we focus on active learning that directly focuses on reducing the model uncertainty that impacts the classification accuracy of the resulting classifier. While the reduction of model uncertainty often results in the decrease in classification error, it is important to note that not all model uncertainty affects classification error. Rather than reducing uncertainty in general, an active learning scheme that aims at reducing the model uncertainty that critically impacts the objective (*i.e.*, classification accuracy) can significantly improve its label efficiency.

There is one category of methods based on Expected Loss Reduction (ELR) that aims to maximize the reduction in classification error directly in a one-step-look-ahead manner (Roy and McCallum, 2001; Zhu et al., 2003; Kapoor et al., 2007). They directly target at reducing the classification error and can achieve the expected optimal performance that is achievable with one single query (Roy and McCallum, 2001). However, these methods do not have any theoretical convergence guarantee, and empirically, they suffer from myopic behavior with degraded efficiency in the long run. Yoon et al. (2013) proposed a metric, Mean Objective Cost of Uncertainty (MOCU), which enables model uncertainty quantification by estimating the expected classification performance loss compared with the optimal classifier due to the uncertainty. MOCU is equivalent to ELR when applied to active learning and provides a tool to analyze the convergence of active learning methods to the true optimal classifier.

In this paper, we first analyze why ELR- or MOCU-based active learning methods may get stuck before collecting enough data to identify the true optimal classifier—despite their efficacy in identifying optimal one-step queries. We further propose a novel acquisition function based on a strictly concave approximation of MOCU, referred to as *Soft MOCU*, to address this problem. Thanks to the strict concavity of Soft MOCU, the resulting acquisition function can capture the continuous change in model uncertainty. As a result, one-step-look-ahead active learning guided by this acquisition function alleviates the limitations due to its myopic nature and is guaranteed to converge to the optimal classifier. We provide theoretical proof of the convergence of the Soft-MOCU-based method. Last but not least, we demonstrate the expected sample efficiency of Soft-MOCU-based active learning with both synthetic and real-world datasets.

2 BACKGROUND

We first review the basic concepts in Bayesian active learning for classification, focusing on the acquisition function targeting directly at the learning objective.

2.1 Mean Objective Cost of Uncertainty

Mean Objective Cost of Uncertainty (MOCU) is a metric measuring the direct influence on the performance with respect to the learning objective due to model uncertainty (Yoon et al., 2013, 2020). We here provide a review in the context of learning Bayesian classifiers.

Consider the classification problem in the input feature space $\boldsymbol{x} \in \mathcal{X}$ and output label space $y \in \mathcal{Y} = \{0, 1, \ldots, M-1\}$ with a probabilistic model characterized by θ as $p(y|\boldsymbol{x}, \theta)$. The aim is to find a classifier $\psi : \mathcal{X} \to \mathcal{Y}$ to estimate the label given a testing feature vector $\boldsymbol{x}^* \in \mathcal{X}$ as $\psi(\boldsymbol{x}^*)$. In this paper we focus on the 0-1 loss to measure the performance of a classifier, which directly reflects the classification error. Denote the expected classification error of ψ on $\boldsymbol{x}$ as $C_\theta(\psi, \boldsymbol{x}) = \mathbb{E}_{p(y|\boldsymbol{x},\theta)}[\mathbb{1}(\psi(\boldsymbol{x}) \neq y)] = 1 - p(y = \psi(\boldsymbol{x})|\boldsymbol{x}, \theta)$. The optimal classifier of θ, denoted as ψ_θ, is defined as the classifier that minimizes the error $\psi_\theta := \arg\min_\psi C_\theta(\psi, \boldsymbol{x}) = \arg\max_y p(y|\boldsymbol{x}, \theta)$.

In the practical situations with model uncertainty where the true model parameter θ_r is unknown, we often assume that based on prior knowledge or observed data, we can derive a distribution $\pi(\theta)$ over the uncertain model parameter set $\theta \in \Theta$. As we do not know the true model, the learning objective is to train an *Optimal Bayesian Classifier* (OBC) $\psi_{\pi(\theta)}$ that minimizes the expected classification error over $\pi(\theta)$ (Dalton and Dougherty, 2013):

$$\psi_{\pi(\theta)} = \arg\min_\psi \mathbb{E}_{\pi(\theta)}[C_\theta(\psi, \boldsymbol{x})] = \arg\max_y p(y|\boldsymbol{x}), \tag{1}$$

where $p(y|\boldsymbol{x}) = \mathbb{E}_{\pi(\theta)}[p(y|\boldsymbol{x}, \theta)]$ is the *predictive distribution*. OBC is the optimal classifier based on the current knowledge. If we can observe enough data to update our model knowledge $\pi(\theta)$ with reduced model uncertainty, OBC will converge to the true optimal classifier based on the true model θ_r .

In Bayesian classification, MOCU can be defined as the expected difference between the expected error of OBC

and the optimal classifier due to model uncertainty:

$$\mathcal{M}(\pi(\theta)) = \mathbb{E}_{\boldsymbol{x}}[\mathbb{E}_{\pi(\theta)}[C_\theta(\psi_{\pi(\theta)}, \boldsymbol{x}) - C_\theta(\psi_\theta, \boldsymbol{x})]], \quad (2)$$

where $\mathbb{E}_{\boldsymbol{x}}$ stands for averaging over the feature space $\mathcal{X}$. The first term is the OBC error. Since ψ_θ is the optimal classifier with a specific θ, for the terms inside the expectation, $C_\theta(\psi_{\pi(\theta)}, \boldsymbol{x}) - C_\theta(\psi_\theta, \boldsymbol{x}) \geq 0$. So the second term is a lower bound of the OBC error. Denote $\text{supp}(\pi)$ as the support of $\pi(\theta)$. If $\mathcal{M}(\pi(\theta)) = 0$, then $\forall \boldsymbol{x} \in \mathcal{X}$, $\forall \theta \in \text{supp}(\pi)$, $\psi_{\pi(\theta)} = \psi_\theta$, i.e. $\arg\max_y p(y|\boldsymbol{x}) = \arg\max_y p(y|\boldsymbol{x}, \theta)$, indicating that OBC is the true optimal classifier. Note that MOCU does not capture all the model uncertainty as we only require $\arg\max_y p(y|\boldsymbol{x}, \theta) = \arg\max_y p(y|\boldsymbol{x})$ instead of $p(y|\boldsymbol{x}, \theta) = p(y|\boldsymbol{x})$ to make MOCU$= 0$. But with MOCU$= 0$, we have found the true optimal classifier and there is no need to further reduce the model uncertainty considering our learning objective.

2.2 Pool-based Bayesian Active Learning

Bayesian active learning sequentially searches for candidates in $\mathcal{X}$ as queries to acquire their labels by optimizing an acquisition function. Then by including the new observed data into the training dataset D, the learning algorithm updates the posterior distribution $\pi(\theta|D)$, with which the acquisition function will be computed to guide active learning in each iteration. In the following discussion, to simplify the notations, we use $\pi(\theta)$ and $p(y|\boldsymbol{x})$ for the posterior and predictive distribution conditioned on D by omitting D in the notations. When a new observation pair $(\boldsymbol{x}^*, y^*)$ is collected, the posterior and the predictive distribution are updated by $\pi(\theta|\boldsymbol{x}^*, y^*) = \frac{\pi(\theta)p(y^*|\boldsymbol{x}^*, \theta)}{p(y^*|\boldsymbol{x}^*)}$ and $p(y|\boldsymbol{x}; \boldsymbol{x}^*, y^*) = \mathbb{E}_{\pi(\theta|\boldsymbol{x}^*, y^*)}[p(y|\boldsymbol{x}, \theta)]$.

We can define the acquisition function based on MOCU in a one-step-look-ahead manner:

$$U^{\text{M}}(\boldsymbol{x}; \pi(\theta)) = \mathcal{M}(\pi(\theta)) - \mathbb{E}_{p(y|\boldsymbol{x})}\mathcal{M}(\pi(\theta|\boldsymbol{x}, y)), \quad (3)$$

which is the expected reduction of MOCU if observing the new pair $(\boldsymbol{x}, y)$. As y is not known at the current iteration to acquire the label, it is averaged over all possible values of y.

We can show that $C_\theta(\psi_\theta, \boldsymbol{x}')$ in the MOCU definition (2) can be cancelled in two MOCUs in (3). Since $\pi(\theta) = \mathbb{E}_{p(y|\boldsymbol{x})}[\pi(\theta|\boldsymbol{x}, y)]$ ($\boldsymbol{x}$ is often assumed to be independent of θ so $\pi(\theta|\boldsymbol{x}) = \pi(\theta)$), we can rewrite the first term in (3) as:

$$\mathcal{M}(\pi(\theta)) = \mathbb{E}_{\boldsymbol{x}'}\{\mathbb{E}_{p(y|\boldsymbol{x})}[\mathbb{E}_{\pi(\theta|\boldsymbol{x}, y)}[C_\theta(\psi_{\pi(\theta)}, \boldsymbol{x}') - C_\theta(\psi_\theta, \boldsymbol{x}')]]\} \quad (4)$$

while the second term in (3) can be expanded as:

$$\mathbb{E}_{p(y|\boldsymbol{x})}\mathcal{M}(\pi(\theta|\boldsymbol{x}, y)) = \mathbb{E}_{\boldsymbol{x}'}\{\mathbb{E}_{p(y|\boldsymbol{x})}[\mathbb{E}_{\pi(\theta|\boldsymbol{x}, y)}[C_\theta(\psi_{\pi(\theta|\boldsymbol{x}, y)}, \boldsymbol{x}') - C_\theta(\psi_\theta, \boldsymbol{x}')]]\}. \quad (5)$$

So the term $C_\theta(\psi_\theta, \boldsymbol{x}')$ can be cancelled out. The acquisition function is just the OBC prediction error reduction after observing the new pair $(\boldsymbol{x}, y)$:

$$U^{\text{M}}(\boldsymbol{x}; \pi(\theta)) = \mathbb{E}_{\boldsymbol{x}'}\{\mathbb{E}_{\pi(\theta)}[C_\theta(\psi_{\pi(\theta)}, \boldsymbol{x}')]\} - \mathbb{E}_{\boldsymbol{x}'}\{\mathbb{E}_{p(y|\boldsymbol{x})}[\mathbb{E}_{\pi(\theta|\boldsymbol{x}, y)}[C_\theta(\psi_{\pi(\theta|\boldsymbol{x}, y)}, \boldsymbol{x}')]]\}, \quad (6)$$

which is the same acquisition function as Error Loss Reduction (ELR) (Roy and McCallum, 2001).

In this paper, we focus on MOCU-based active learning with the OBC as the classifier. As shown in (6), MOCU-based active learning queries the candidate to achieve the maximum expected reduction in OBC classification error in each iteration. Hence, the MOCU-based method is the optimal strategy for active learning of the OBC with a single query.

3 METHODS

In this section, we first show that MOCU-based active learning based on a one-step-look-ahead strategy may get stuck before MOCU converges to 0 with the corresponding OBC converging to the true optimal classifier. We then propose a new acquisition function that has the guarantee that the OBC converges to the true optimal classifier.

3.1 Analysis of MOCU-based Active Learning

We discuss the myopic behavior of MOCU-based Bayesian active learning due to the inherent limitations of the one-step-look-ahead setup. As we have shown, MOCU-based active learning is an example of one-step-look-ahead strategies, which are only optimal for the current single iteration. Practically, MOCU-based active learning usually performs well in the first several iterations of active learning but there is no guarantee of good performance in the long run.

We now analyze why MOCU-based active learning may get stuck before the OBC converges to the true optimal classifier. In other words, when the acquisition function for all the candidates in the pool is 0, i.e. $\forall \boldsymbol{x} \in \mathcal{X}, U^{\text{M}}(\boldsymbol{x}; \pi(\theta)) = 0$, the active learning will degenerate to random sampling and keep selecting the candidate based on the adopted tie-breaking strategy. When that happens, we say that active learning gets stuck without converging to the true optimal classifier if MOCU is still larger than 0.

We first show that the MOCU (2) is a concave function of $\pi(\theta)$, but it is not strictly concave everywhere with nonzero curvature to guide active learning. From the definition of ψ_θ and $\psi_{\pi(\theta)}$ in Section 2.1, we have $C_\theta(\psi_\theta, \boldsymbol{x}) = 1 - \max_y p(y|\boldsymbol{x}, \theta)$

and $\mathbb{E}_{\pi(\theta)}[C_\theta(\psi_{\pi(\theta)}, \boldsymbol{x})] = \mathbb{E}_{\pi(\theta)}[1 - p(\psi_{\pi(\theta)}(\boldsymbol{x})|\boldsymbol{x}, \theta)] = 1 - \max_y \mathbb{E}_{\pi(\theta)}[p(y|\boldsymbol{x}, \theta)]$. Substituting them in (2),

$$\mathcal{M}(\pi(\theta)) = \mathbb{E}_{\boldsymbol{x}}\{\mathbb{E}_{\pi(\theta)}[\max_y p(y|\boldsymbol{x}, \theta)] - \max_y \mathbb{E}_{\pi(\theta)}[p(y|\boldsymbol{x}, \theta)]\}, \quad (7)$$

$$= \mathbb{E}_{\boldsymbol{x}}\{\mathbb{E}_{\pi(\theta)}[\max_y p(y|\boldsymbol{x}, \theta)] - p(\psi_{\pi(\theta)}(\boldsymbol{x})|\boldsymbol{x}, \theta)]\} \quad (8)$$

In (7), the first term $\mathbb{E}_{\pi(\theta)}[\max_y p(y|\boldsymbol{x}, \theta)]$ is a linear function of $\pi(\theta)$. While the second term $\max_y \mathbb{E}_{\pi(\theta)}[p(y|\boldsymbol{x}, \theta)]$ is the maximum over M linear functions and thus is a convex piecewise linear function. As a result, (7) equals to a linear function subtracting a convex function and therefore it is concave and also piecewise linear. It is thus not strictly concave everywhere. Within each piece of the linear functions in (8), the OBC classifier $\psi_{\pi(\theta)}(\boldsymbol{x}), \boldsymbol{x} \in \mathcal{X}$ takes the same label for different $\pi(\theta)$. To gain the intuition, assume a three-class classification problem with the uncertainty class of two models $\Theta = \{\theta_1, \theta_2\}$ and the pool with only one candidate $\mathcal{X} = \{\boldsymbol{x}\}$. Let the probabilistic model $p(y|\boldsymbol{x}, \theta_1) = (0.5, 0.4, 0.1)$ and $p(y|\boldsymbol{x}, \theta_2) = (0.15, 0.4, 0.45)$. Since $\pi(\theta_2) = 1 - \pi(\theta_1)$ in this setup, the MOCU can be expressed as a function of $\pi(\theta_1)$ as shown in Fig. 1. As we explained, it is a piecewise linear function. Within the three intervals corresponding to the linear function pieces, $\arg\max_y p(y|\boldsymbol{x})$ is 2, 1, and 0 from left to right, respectively.

In (3), we observe that $\pi(\theta) = \mathbb{E}_{p(y|\boldsymbol{x})}[\pi(\theta|\boldsymbol{x}, y)]$. Since MOCU is a concave function, based on Jensen's Inequality, we have $U^{\mathrm{M}}(\boldsymbol{x}; \pi(\theta)) \geq 0$. While MOCU is not a strictly concave function as explained, we can find two conditions that make the equality to hold: first, $\forall y \in \mathcal{Y}$, $\pi(\theta|\boldsymbol{x}, y) = \pi(\theta)$, which means observing $\boldsymbol{x}$ does not help change the knowledge about θ; second, $\pi(\theta|\boldsymbol{x}, y)$ changes but the change of $\pi(\theta|\boldsymbol{x}, y)$, $\forall y \in \mathcal{Y}$, is within the same linear piece of MOCU. In the second condition, observing $\boldsymbol{x}$ one time only provides little information of θ, and that information will not change the OBC classifier. If MOCU is larger than 0 but all the $\boldsymbol{x}$ cannot provide enough information to update the OBC classifier, the acquisition function can then be 0 for all the candidates, with which MOCU-based active learning will get stuck before converging to the true optimal classifier. Appendix B provides such a synthetic example and Appendix E explicitly shows that MOCU-based active learning can get stuck. From the discussion above, we can see that MOCU-based active learning may get stuck because MOCU is not strictly concave.

In the next section, we propose a strictly concave function to approximate MOCU and an one-step-look-ahead acquisition function based on it. The approximation makes the corresponding active learning to have similar short-term optimality for single iterations as in MOCU-based active learning. More importantly, the imposed strictly concavity leads to the theoretical guarantee that the OBC will converge to the true optimal classifier without getting stuck.

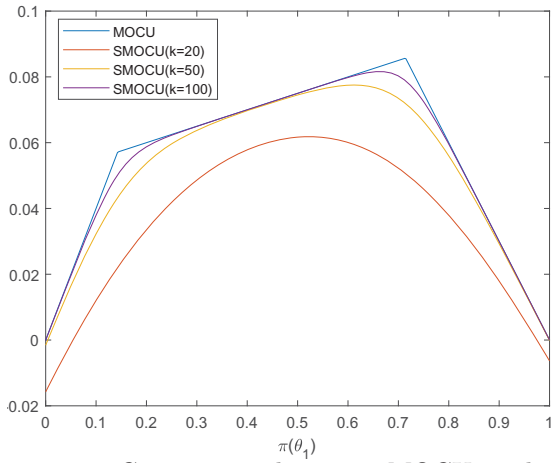


Figure 1: Comparison between MOCU and Soft MOCU with different k values

3.2 Soft-MOCU-based Active Learning

As we discussed, the myopic behavior that MOCU-based active learning has is due to the linear function pieces causing the acquisition function for active learning to lose the guiding capability when the update of $\pi(\theta|\boldsymbol{x}, y)$ is not significant enough. To address this problem, we propose a new acquisition function based on modified MOCU, which has the theoretical convergence guarantee to the true optimal classifier.

In this paper, We approximate the maximum operator in (2) by the `log-sum-exp` function:

$$\max_y p(y|\boldsymbol{x}) \approx \frac{1}{k} \log[\sum_y \exp(k \cdot p(y|\boldsymbol{x}))], \quad (9)$$

where k is a parameter controlling the approximation. Using a larger k in `log-sum-exp` gives a better approximation to the maximum operator. Note that other functions can also be used. With this approximation, we can define the following *Soft MOCU* (SMOCU) as:

$$\mathcal{M}^s(\pi(\theta)) = \mathbb{E}_{\boldsymbol{x}}\{\mathbb{E}_{\pi(\theta)}[\max_y p(y|\boldsymbol{x}, \theta)] - \frac{1}{k} \log[\sum_y \exp(k \cdot p(y|\boldsymbol{x}))]\}, \quad (10)$$

which is now a strictly concave function instead of being piecewise linear. Similarly, as with larger k, Soft MOCU gets closer to MOCU. We illustrate the modified MOCU with different k values in Fig. 1 for the example described in Section 3.1.

We now define an acquisition function by the reduction

of Soft MOCU to guide active learning in the one-step-look-ahead manner:

$$U^s(\boldsymbol{x};\pi(\theta)) = \mathcal{M}^s(\pi(\theta)) - \mathbb{E}_{p(y|\boldsymbol{x})}[\mathcal{M}^s(\pi(\theta|\boldsymbol{x},y))] \quad (11)$$

As shown in the example, Soft MOCU can provide a good approximation to MOCU. More critically, it also has large curvature on the changing points of MOCU so that the above acquisition function has large values when the update of $\pi(\theta|\boldsymbol{x},y)$ is significant causing the change of OBC. While when the update of $\pi(\theta|\boldsymbol{x},y)$ is not significant (falling within intervals of linear pieces in the original MOCU), Soft MOCU still has small curvature so that the acquisition function has small positive values instead of being zero as in (3). With these properties of the Soft-MOCU-based acquisition function, when the model has high uncertainty with large MOCU values, the approximation by Soft MOCU will not affect the choice of candidates and the corresponding active learning performs similarly as the original MOCU-based method to achieve short-term optimality. On the other hand, when the model has low uncertainty and a single query will not be able to change $\pi(\theta|\boldsymbol{x},y)$ significantly, for example when $\pi(\theta_1)$ is close to 0 or 1 in Fig. 1, the MOCU-based method will get stuck. However, our Soft-MOCU-based acquisition function can still guide active learning out of the myopic behavior. Please refer to Appendix A for the pseudo-code of our Soft-MOCU-based active learning and the complexity analysis.

3.3 Theoretical Convergence Guarantee

We now first prove that Soft MOCU (10) is a strictly concave function. If active learning is guided by the acquisition function (11) based on Soft MOCU, MOCU will converge to 0. This means that we can learn the optimal classifier of the true model without getting stuck with the theoretical convergence guarantee.

We assume that both $\mathcal{X}$ and Θ are discrete with finite elements, and the true model parameter $\theta_r \in \Theta$ with $\pi^0(\theta_r) > 0$ for the prior $\pi^0(\theta)$.

Lemma 1 *$\mathcal{M}^s(\pi(\theta))$ is a strictly concave function of $\pi(\theta)$.*

Proof. It is known that `log-sum-exp` is a convex function (page 74, Sec. 3.1 in Boyd and Vandenberghe (2004)). We now prove that $f(p(y|\boldsymbol{x})) = \frac{1}{k}\log[\sum_y \exp(k\cdot p(y|\boldsymbol{x}))]$ is a strictly convex function of $p(y|\boldsymbol{x})$ conditioning on $\sum_y p(y|\boldsymbol{x}) = 1$. In the following proof, we denote $p(y|\boldsymbol{x})$ for $y \in \mathcal{Y}$ as the vector $\boldsymbol{z}$ for simplicity. From Boyd and Vandenberghe (2004),

$$\nabla^2 f(\boldsymbol{z}) = k(\mathrm{diag}(g) - gg^T),\ g := \frac{\exp(\boldsymbol{z})}{\mathbf{1}^T\exp(\boldsymbol{z})}, \quad (12)$$

where $\exp(\boldsymbol{z}) = (e^{z_1},\ldots,e^{z_M})$. Note that in the expression of $\nabla^2 f(\boldsymbol{z})$, $\mathrm{diag}(g)$ is a full-rank matrix and $\mathrm{rank}(gg^T)$ is 1. Therefore, $rank(\nabla^2 f) = n-1$ and $f(\boldsymbol{z})$ is affine (being a linear function) along only one direction. Apparently that direction is along the all-ones vector $\mathbf{1}$, as can be verified by: $\mathbf{1}^T\nabla^2 f(\boldsymbol{z})\mathbf{1} = 0$, and f is strictly convex along any other directions. In addition, since $\boldsymbol{z}$ denotes a probability mass function, it is constrained on the hyperplane $\mathbf{1}^T\boldsymbol{z} = 1$. On the hyperplane, no vector is parallel to $\mathbf{1}$, as $\mathbf{1}^T(\boldsymbol{z}+\alpha\mathbf{1}) \neq 1 \,\text{for}\, \alpha \neq 0$. Hence, within the hyperplane f is a strictly convex function.

Since $f(p(y|\boldsymbol{x}))$ is a strictly concave function and $p(y|\boldsymbol{x})$ is a linear function of $\pi(\theta)$, $\log[\sum_y \exp(k\cdot p(y|\boldsymbol{x}))]$ is therefore a strictly convex function of $\pi(\theta)$. $\mathcal{M}^s(\pi(\theta))$ is equal to a linear function subtracting a strictly convex function and hence is a strictly concave function of $\pi(\theta)$. □

Lemma 2 *$\forall \boldsymbol{x} \in \mathcal{X}, U^s(\boldsymbol{x};\pi(\theta)) \geq 0$; the equality only holds for the case $\pi(\theta) = \pi(\theta|\boldsymbol{x},y),\ \forall y \in \mathcal{Y}$.*

Proof. Since Soft MOCU is a strictly concave function and $\mathbb{E}_{p(y|\boldsymbol{x})}[\pi(\theta|\boldsymbol{x},y))] = \pi(\theta)$, by Jensen's inequality, we have

$$U^s(\boldsymbol{x};\pi(\theta)) = \mathcal{M}^s(\pi(\theta)) - \mathbb{E}_{p(y|\boldsymbol{x})}[\mathcal{M}^s(\pi(\theta|\boldsymbol{x},y))] \geq 0. \quad (13)$$

and the equality only holds if $\pi(\theta) = \pi(\theta|\boldsymbol{x},y),\ \forall y \in \mathcal{Y}$. □

Lemma 3 *If $U^s(\boldsymbol{x};\pi(\theta)) = 0, \forall \boldsymbol{x} \in \mathcal{X}$, then $\mathcal{M}(\pi(\theta)) = 0$.*

This lemma states that if the acquisition function values of all the candidates are 0, then we can conclude that MOCU is 0. This means that the OBC of $\pi(\theta)$ has converged to the true optimal classifier ψ_{θ_r}. MOCU-based active learning does not have such a property. Because of that, it may get stuck before converging to the true optimal classifier.

Proof. We will show that the lemma holds by proving the contraposition: if $\mathcal{M}(\pi(\theta)) > 0,\ \exists \boldsymbol{x} \in \mathcal{X}\ s.t.\ U^s(\boldsymbol{x};\pi(\theta)) > 0$.

Based on (2), $M(\pi(\theta)) > 0$ indicating $\exists \boldsymbol{x}^* \in \mathcal{X}\ \exists \theta^* \in \mathrm{supp}(\pi)$ s.t. $\psi_{\pi(\theta)}(\boldsymbol{x}^*) \neq \psi_{\theta^*}(\boldsymbol{x}^*)$, i.e. $\max_y p(y|\boldsymbol{x}^*) \neq \max_y p(y|\boldsymbol{x}^*,\theta^*)$, where $\mathrm{supp}(\pi)$ is the support of $\pi(\theta)$. So $\exists y^* \in \mathcal{Y}\ s.t.\ p(y^*|\boldsymbol{x}^*,\theta^*) \neq p(y^*|\boldsymbol{x}^*)$. Now we assume that we observe $(\boldsymbol{x}^*,y^*)$, then the update of $\pi(\theta^*)$ can be written as:

$$\pi(\theta^*|\boldsymbol{x}^*,y^*) = \frac{\pi(\theta^*)p(y|\boldsymbol{x}^*,\theta^*)}{p(y^*|\boldsymbol{x}^*)}. \quad (14)$$

Since $p(y^*|\boldsymbol{x}^*,\theta^*) \neq p(y^*|\boldsymbol{x}^*)$, we have $\pi(\theta^*|\boldsymbol{x}^*,y^*) \neq$

$\pi(\theta^*)$. With Lemma 2, we can have $U^s(\boldsymbol{x}^*;\pi(\theta)) > 0$. That concludes our proof. □

Lemma 4 *If a candidate $\boldsymbol{x}$ is measured infinitely often almost surely (a.s.), $U^s(\boldsymbol{x};\pi^n(\theta)) \xrightarrow{a.s.} 0$ as $n \to \infty$.*

Intuitively, if a candidate has been measured many times, there is no benefit to measure it again.

Proof. For a candidate $\boldsymbol{x}$, define a set of θ as $\Theta_{\boldsymbol{x}} = \{\theta \in \Theta : p(y|\boldsymbol{x},\theta) = p(y|\boldsymbol{x},\theta_r)\}$. At the n-th iteration, assume that the candidate $\boldsymbol{x}$ has been observed $N_{\boldsymbol{x}}(n)$ times and $\lim_{n\to\infty} N_{\boldsymbol{x}}(n) = \infty$. Based on the posterior consistency theory we have $\sum_{\theta\in\Theta_x} \pi^n(\theta) \xrightarrow{a.s.} 1$ as $n \to \infty$ (Gelman et al., 2013). Since $p^n(y|\boldsymbol{x}) = \sum_{\theta\in\Theta} \pi^n(\theta)p(y|\boldsymbol{x},\theta)$, we have $p^n(y|\boldsymbol{x}) \xrightarrow{a.s.} p(y|\boldsymbol{x},\theta_r)$. By Bayes' rule, $\pi^n(\theta|\boldsymbol{x},y) = \frac{\pi^n(\theta)p(y|\boldsymbol{x},\theta)}{p^n(y|\boldsymbol{x})}$, and hence we have $\pi^n(\theta|\boldsymbol{x},y) - \pi^n(\theta) \xrightarrow{a.s.} 0$. With Lemma 2, we can conclude that $U^s(\boldsymbol{x}_n;\pi^n(\theta)) \xrightarrow{a.s.} 0$ as $n \to \infty$. □

Theorem 1 *Assume that both $\mathcal{X}$ and Θ are discrete with finite elements, the true model parameter $\theta_r \in \Theta$ and $\pi^0(\theta_r) > 0$; then for the active learning algorithm defined by the acquisition function (11), we have $\mathcal{M}(\pi^n(\theta)) \xrightarrow{a.s.} 0$ as $n \to \infty$.*

Proof. As the number of active learning iterations $n \to \infty$, some of the candidates will be measured infinitely often. Following the Soft-MOCU-based method by the acquisition function (11), denote the set of candidates being measured infinitely often as $\mathcal{X}_I = \{\boldsymbol{x} \in \mathcal{X} : \lim_{n\to\infty} N_{\boldsymbol{x}}(n) = \infty\}$. With the query sequence of the candidates as $\{\boldsymbol{x}_n\}$, we have $\exists N,\ s.t.\ \forall n > N, \boldsymbol{x}_n \in \mathcal{X}_I$, which means that after N iterations, we can only observe candidates from the set $\mathcal{X}_I$. Based on Lemma 4, this indicates $U^s(\boldsymbol{x}_n;\pi^n(\theta)) \xrightarrow{a.s.} 0$.

On the other hand, as Soft-MOCU-based active learning maximizes the acquisition function in each iteration, we have $U^s(\boldsymbol{x}_n;\pi^n(\theta)) = \max_{\boldsymbol{x}\in\mathcal{X}} U^s(\boldsymbol{x};\pi^n(\theta))$. Then the maximum value $U^s(\boldsymbol{x}_n;\pi^n(\theta))$ converging to 0 means that $U^s(\boldsymbol{x};\pi^n(\theta)),\ \boldsymbol{x} \in \mathcal{X}$ converges to 0 uniformly. Based on Lemma 3, we have $\mathcal{M}(\pi^n(\theta)) \xrightarrow{a.s.} 0$ and we can conclude the proof. □

We should emphasize that the inverse of Lemma 3 is not true. When MOCU is 0, the acquisition function of some candidate $\boldsymbol{x}$'s can still be positive. To understand this, as we have shown in Section 2.1, MOCU does not capture all the model uncertainty. On the other hand, based on Lemma 2, the acquisition function based on Soft MOCU can only be 0 when there is no model uncertainty.

4 EMPIRICAL RESULTS

We first investigate the influence of the parameter k on the performance of our Soft-MOCU-based active learning (SMOCU). We then benchmark SMOCU with other active learning methods, including random sampling, MES (Sebastiani and Wynn, 2000), BALD (Houlsby et al., 2011) and MOCU, on both simulated and real-world classification datasets. The code is made available at `https://github.com/QianLab/Soft_MOCU`.

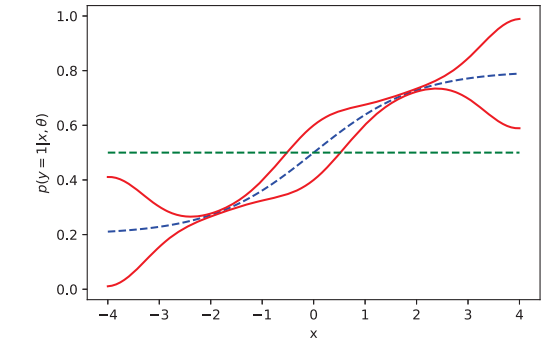
4.1 Performance of Soft-MOCU with Different k Values

Here we compare the performance of SMOCU with different k values together with MOCU and BALD on a binary classification problem with one feature $x \in [-4, 4]$. The underlying probabilistic model is:

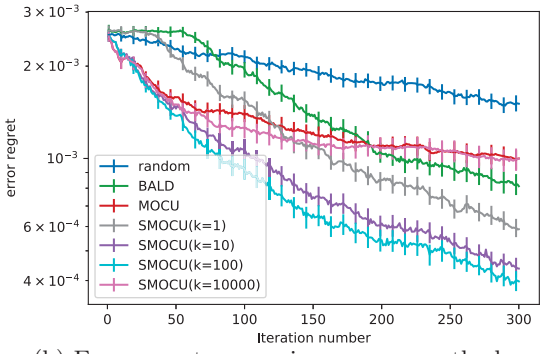
$$\begin{aligned} p(y=1|x,\alpha,\beta) &= S(x) + \epsilon(x,\alpha,\beta) \\ S(x) &= 0.6\frac{\exp(x)}{1+\exp(x)} + 0.2 \\ \epsilon(x,\alpha,\beta) &= \alpha\exp(-x^2) + \\ &\quad \beta[\exp(-(x-4)^2) + \exp(-(x+4)^2)], \end{aligned} \tag{15}$$

where $\boldsymbol{\theta} = (\alpha,\beta)^{\mathrm{T}}$ is the uncertain parameter vector with α and β independently uniformly distributed in the intervals $[-0.1, 0.1]$ and $[-0.2, 0.2]$ respectively. Fig. 2a illustrates the uncertain probabilistic model with red lines indicating the upper and lower bounds of the predictive probability. The probabilistic model has higher uncertainty near $x = \pm 4$ depending on β than the uncertainty near $x = 0$ depending on α. Observing data near $x = \pm 4$ can reduce model uncertainty significantly and is preferred by BALD, but it cannot help on the label prediction since the optimal classifier will always label $x = \pm 4$ as 1 or 0. On the other hand, as the optimal labels of the points in the middle are uncertain given the prior knowledge, MOCU-based active learning will query these points first to better reduce the classification error at the beginning.

We randomly sample the true parameters from the prior and perform different active learning methods for 300 iterations. We compare different methods by the error regret, which is defined as the error difference between the OBC and the true optimal classifier. We repeat the simulations for 500 runs and plot the average performance with standard deviation bars in Fig. 2b. From the figure, not surprisingly, BALD performs inefficiently at the beginning since it queries the candidates on both sides. MOCU performs well at the beginning but becomes inefficient after about 100 iterations, indicating some of the 500 simulations get stuck as we analyzed in Section 3.1.



(a) Predictive probability of class 1



(b) Error regret comparison among methods

Figure 2: (a) Predictive probability of class 1 under uncertainty: the red lines indicate the upper and lower bounds of the predictive probability; the blue dashed line is the mean of the predictive probability; the green dashed line indicates that the probability is equal to 0.5. (b) Active learning performance.

For Soft MOCU with different k values, as we shown in Fig. 1, Soft MOCU gets closer to MOCU with increasing k. As a result, Soft-MOCU-based active learning should perform more similar to the MOCU-based method as k increases. We can see from the figure that, when k is small ($k = 1$), the performance is close to BALD that aims to reduce the total model uncertainty. With increasing k ($= 10$ or 100), the performance of Soft-MOCU-based active learning at the beginning gets closer to MOCU and more importantly, in the long run it performs better than both BALD and MOCU, demonstrating Soft-MOCU-based active learning can converge to the optimal classifier with fewer iterations. As expected, when k is really large ($k = 10000$), Soft MOCU can get really close to MOCU with very small curvature with respect to $\pi(\theta)$ as illustrated in Fig. 1, which leads to similar performance degradation as shown in Fig. 2b.

We next benchmark Soft-MOCU-based active learning for more simulated experiments and real-world experiments, for which we compare active learning methods based on random sampling, MES, BALD, MOCU, and our Soft MOCU with $k = 10$ and 100.

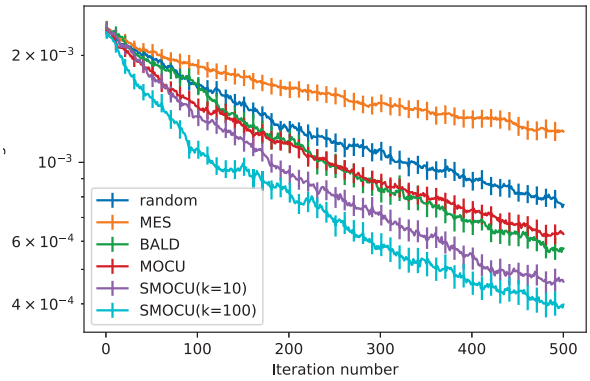


Figure 3: Comparison of different active learning methods based on the expected OBC error regret for binary classification.

4.2 Simulated Experiments

We test these active learning methods on a simulated experiment similar as the *block in the center* dataset in Houlsby et al. (2011). The experiment includes a binary classification problem with candidates from 2-d feature space $[-4, 4]^2$. The simulated data are described by a Bayesian logistic regression model:

$$p(y = 1|\boldsymbol{x}, \boldsymbol{w}, b) = \frac{1}{1 + \exp(-\boldsymbol{w}^T\boldsymbol{x} - b)}, \qquad (16)$$

with a uniform parameter prior $w_1 \sim \mathcal{U}(0.3, 0.8)$, $w_2 \sim \mathcal{U}(-0.1, 0.1)$ and $b \sim \mathcal{U}(-0.25, 0.25)$; w_1, w_2 and b are independent. With this prior setting, the uncertainty of $p(y|\boldsymbol{x}, \boldsymbol{w}, b)$ is low in the region near the x_2 axis where the decision boundary lies and the uncertainty is high in the region far away from the x_2 axis. Within the block region of $[-1, 1]^2$, the observed labels are flipped with the probability 0.3.

We randomly sample 100 particles from the parameter prior as the uncertain parameter set, and randomly choose one of them as the true parameter. We also uniformly sample 100 candidates from the feature space as the candidate pool. Then we perform these different methods for 500 iterations and calculate the error regret. We repeat the simulation for 500 times and plot the performance comparison with standard deviations in Fig. 3. From the figure, MES has quite poor performance as it simply queries the candidates with the predictive probability close to 0.5. It may sample many noisy observations from the noisy block region. BALD performs poorly at the beginning since it cannot identify which uncertainty is related to the learning objective. MOCU performs well in the first several iterations, but poorly in the long run. As expected, our Soft-MOCU-based methods perform better than other competing methods with $k = 100$ performing the best.

We also compare these different methods on a multi-class classification setup. We assume the fea-

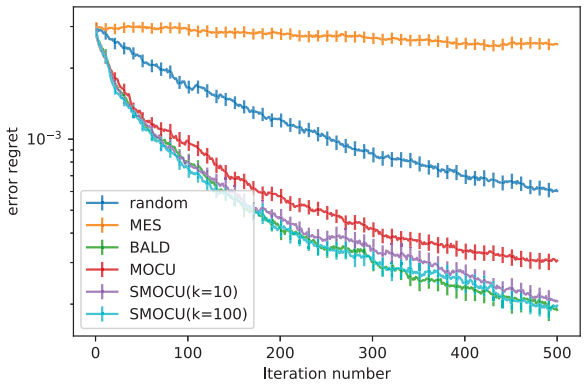


Figure 4: Comparison of different active learning methods based on the expected OBC error regret for three-class classification

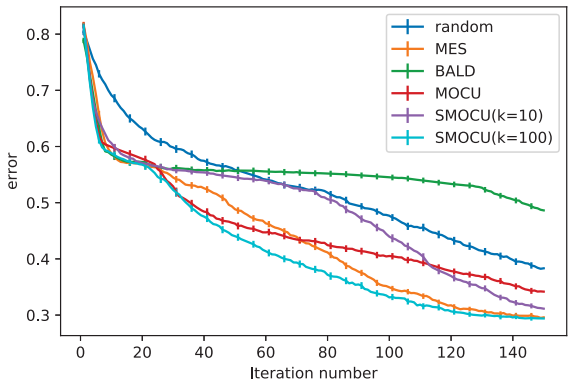


Figure 5: Classification error comparison on UCI User Knowledge dataset

ture space $\mathcal{X} = [-2, 2]^2$ and label space $\mathcal{Y} = \{0, 1, 2\}$ with the probabilistic model $p(y|\boldsymbol{x}, \sigma_y^2) = f_y(\boldsymbol{x}, \sigma_y^2)/\sum_{y'} f(\boldsymbol{x}, \sigma_{y'}^2)$, where $f_y(\boldsymbol{x}, \sigma_y^2) = \exp(-(x - m_y)^2/2\sigma_y^2)$, $y \in \mathcal{Y}$. We set m_y to be $(0, 0)$, $(1, 0)$, $(0, 1)$ for $y = 0, 1, 2$ respectively; and independent uncertain parameters $\sigma_y^2 \sim \mathcal{U}(1, 5)$, $y \in \mathcal{Y}$. Similar as the previous binary classification experiment, we perform the five methods for 500 times and plot the average error regret with standard deviations in Fig. 4. From the figure, MES performs poorly as it samples the candidates with maximal predictive entropy, while querying these candidates provides little information to improve classification. We again observe that MOCU performs poorly in the long run while both Soft-MOCU-based methods have better empirical performance on a par with BALD.

4.3 Real-world Benchmark Experiments

We compare different active learning methods on the UCI User Knowledge dataset (Kahraman et al., 2013). The dataset assigns the knowledge status of 403 students into four levels (High, Medium, Low, Very Low) based on five input features in $[0, 1]^5$, which reflect the degree of study or exam performance. Here we use the 1st and 5th features as inputs for classification and equally separate the 2-d feature space into 4×4 bins. Within the ith bin we assume a categorical distribution for the knowledge levels $p(y|\boldsymbol{x} \in i\text{-th bin}) = \boldsymbol{p}^{(i)}$, $1 \leq i \leq 16$ with parameters $\boldsymbol{p}^{(i)} = (p_0^{(i)}, p_1^{(i)}, p_2^{(i)}, p_3^{(i)})$. Assume that each parameter independently follows a Dirichlet distribution $\boldsymbol{p}^{(i)} \sim \text{Dir}(\boldsymbol{\alpha}^{(i)})$ with $\boldsymbol{\alpha}^{(i)}$ as the hyperparameters. We randomly choose 8 bins and set uniform priors on them with $\boldsymbol{\alpha}^{(i)} = \mathbf{1}$. For the other 8 bins, we set the prior by setting $\alpha_j^{(i)} = 1$ if j is the true label, and $\alpha_j^{(i)} = 10$ for other labels. To obtain a balanced classification problem, we randomly sample 50 samples from each class to test the five different methods. We repeat the active learning procedures for 150 times and compare the average classification error in Fig. 5. From the figure, we can clearly observe two stages in the active learning procedures: the first stage has about 20 iterations, in which all the methods learn the optimal classification rules in the 8 bins with the uniform prior; while in the following iterations as the second stage, different methods perform differently based on their acquisition functions. BALD keeps choosing candidates from the bins with the uniform prior in the second stage as those bins still have larger model uncertainty. However, they cannot help improve the classification. MOCU performs well at the beginning, and then converges slowly. Our Soft-MOCU-based method with $k = 100$ is again demonstrated to converge faster than other methods.

More experiments with similar performance trends and detailed discussions can be found in Appendix C & D.

5 CONCLUSIONS

Although the existing ELR- or MOCU-based methods are optimal for active learning when considering single queries, we investigated why they may perform poorly in the long run—both theoretically and empirically. Based on the analysis, we proposed a Bayesian active learning method with a new acquisition function for learning optimal Bayesian classifiers. This acquisition function is defined based on a strictly concave approximation of MOCU, which we refer to as *Soft MOCU*. Our new Soft-MOCU-based active learning is efficient for the initial iterations as it approximates the original MOCU-based active learning scheme. A critical feature of Soft MOCU is that its strict concavity enables the resulting acquisition function to capture small model uncertainty reduction and thus guarantees the OBC to converge to the true optimal classifier even when the myopic one-step-look-ahead queries may not provide significant changes to the model pos-

terior $\pi(\theta|\boldsymbol{x}, y)$. Consequently, our proposed active learning method can be efficient both at the beginning as well as in the long run. In addition to the theoretical guarantee, our empirical results also demonstrated the superior performance of our Soft-MOCU-based method. Finally, as analyzed and observed in our experiments, Soft MOCU with larger k performs better at the beginning as it closely approximates MOCU with local optimality whereas Soft MOCU with smaller k performs better in the long run. Adaptively updating the value of k during the active learning procedure is an interesting research direction.

Acknowledgments

X. Qian was supported in part by the National Science Foundation (NSF) Awards 1553281, 1812641, 1835690, and 1934904. B.-J. Yoon was supported in part by the NSF Award 1835690. The work of E. R. Dougherty and F. J. Alexander was supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Mathematical Multifaceted Integrated Capability Centers program under Award DE-SC0019303.

References

Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Nguyen Viet Cuong, Wee Sun Lee, Nan Ye, Kian Ming A Chai, and Hai Leong Chieu. Active learning for probabilistic hypotheses using the maximum Gibbs error criterion. In *Advances in Neural Information Processing Systems*, pages 1457–1465, 2013.

Lori A Dalton and Edward R Dougherty. Optimal classifiers with minimum expected error within a Bayesian framework—Part I: Discrete and Gaussian models. *Pattern Recognition*, 46(5):1301–1314, 2013.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep Bayesian active learning with image data. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1183–1192. JMLR. org, 2017.

Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian data analysis*. CRC press, 2013.

Daniel Golovin, Andreas Krause, and Debajyoti Ray. Near-optimal Bayesian active learning with noisy observations. In *Advances in Neural Information Processing Systems*, pages 766–774, 2010.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

H Tolga Kahraman, Seref Sagiroglu, and Ilhami Colak. The development of intuitive knowledge classifier and the modeling of domain dependent data. *Knowledge-Based Systems*, 37:283–295, 2013.

Ashish Kapoor, Eric Horvitz, and Sumit Basu. Selective supervision: Guiding supervised learning with decision-theoretic active learning. In *IJCAI*, volume 7, pages 877–882, 2007.

Andreas Kirsch, Joost van Amersfoort, and Yarin Gal. Batchbald: Efficient and diverse batch acquisition for deep Bayesian active learning. In *Advances in Neural Information Processing Systems*, pages 7024–7035, 2019.

Stephen Mussmann and Percy Liang. On the relationship between data efficiency and error for uncertainty sampling. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3674–3682, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

Nicholas Roy and Andrew McCallum. Toward optimal active learning through sampling estimation of error reduction. In *Proceedings of the 18th International Conference on Machine Learning*, ICML '01, pages 441—-448, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1558607781.

Paola Sebastiani and Henry P Wynn. Maximum entropy sampling and optimal Bayesian experimental design. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(1):145–157, 2000.

Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

Toan Tran, Thanh-Toan Do, Ian Reid, and Gustavo Carneiro. Bayesian generative active deep learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6295–6304, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

Byung-Jun Yoon, Xiaoning Qian, and Edward R Dougherty. Quantifying the objective cost of uncertainty in complex dynamical systems. *IEEE Transactions on Signal Processing*, 61(9):2256–2266, 2013.

Byung-Jun Yoon, Xiaoning Qian, and Edward R. Dougherty. Quantifying the multi-objective cost of uncertainty. *arXiv preprint arXiv:2010.04653*, 2020.

Xiaojin Zhu, John Lafferty, and Zoubin Ghahramani. Combining active learning and semi-supervised learning using Gaussian fields and harmonic functions. In *ICML 2003 workshop on the continuum from labeled to unlabeled data in machine learning and data mining*, volume 3, 2003.