Systems & Control Transactions

Research Article - Peer Reviewed Conference Proceeding
ESCAPE 35 - European Symposium on Computer Aided Process Engineering
Ghent, Belgium. 6-9 July 2025
Jan F.M. Van Impe, Grégoire Léonard, Satyajeet S. Bhonsale,
Monika E. Polańska, Filip Logist (Eds.)

PSE PRESS

A Propagated Uncertainty Active Learning Method for Bayesian Classification Problems

Arun Pankajakshan[a], Sayan Pal[a], Maximilian O. Besenhard[a], Asterios Gavriilidis[a], Luca Mazzei[a], Federico Galvanin[a,*]

[a] Department of Chemical Engineering, University College London, Torrington Place, London, WC1E 7JE, United Kingdom
* Corresponding Author: f.galvanin@ucl.ac.uk.

ABSTRACT

Bayesian classification (BC) is a powerful supervised machine learning method for modelling the relationship between a set of continuous variables and a set of discrete variables that represent classes. BC has been successful in engineering and medical applications, including feasibility analysis and clinical diagnosis. Gaussian process (GP) models are widely used in BC methods to model the probability of assigning a class to an input point, typically through an indirect approach: a GP predicts a continuous function value based on Bayesian inference, which is then transformed into class probabilities using a nonlinear function like a sigmoid. The final class labels are assigned based on these probabilities. In this commonly used workflow, the uncertainty associated with the class prediction is usually evaluated as the uncertainty in the GP function values. A disadvantage of this approach is that it does not consider the uncertainty directly associated with the decision-making. In this work, we propagate the uncertainty from the space of GP function values to the class probability space and use this to quantify the uncertainty directly associated with the decision-making process. Additionally, we employ the propagated uncertainty as the objective function in an active learning (AL) method to generate new informative data points for the GP classifier training. We compare the proposed AL method to existing state-of-the-art methods to evaluate its performance.

Keywords: Bayesian classification, Gaussian process, active learning, uncertainty propagation.

1. INTRODUCTION

Probabilistic learning is key to developing autonomous systems that adapt and make reliable decisions without human intervention [1]. One of the popular probabilistic learning approaches that has key applications in engineering and medical domains is Bayesian classification (BC) [2].

BC involves a probabilistic approach to the classification problem. A classification problem is a supervised machine learning method used to build a functional relationship between a set of input variables and a set of classes or categories related to the input variables. Usually, the classes are labelled using discrete values such as 0, 1, 2 etc. BC has proven successful in applications such as medical diagnosis [3] and feasibility analysis [4], where the uncertainty of model predictions is critical. A popular choice of BC implementation involves BC with Gaussian processes (GPs) [5].

In problems of BC with GPs, the posterior GP model, i.e. the model obtained after data fitting or learning through Bayesian inference [5], provides probability predictions corresponding to the class labels along with an associated uncertainty value. This uncertainty offers meaningful insights into the model and supports the design of learning strategies that utilize it. This field of informed learning process to facilitate efficient training of a GP model with limited informative data is known as active learning (AL) [6]. Therefore, the GP model's ability to quantify uncertainty is closely linked to the learning approach employed in AL.

Two approaches are commonly used for uncertainty quantification in BC with GPs: 1) aleatoric or intrinsic uncertainty associated with the class prediction and 2) uncertainty in the GP model predictions. In 1), the class conditional probability values predicted by the GP model are used for uncertainty quantification. For instance, extremely low or extremely high probabilities indicate low uncertainty in the outcome predicted by the model while probabilities near 0.5 indicate a presence of high uncertainty. In 2), uncertainty in BC is quantified as the uncertainty around the point estimates of the GP model predictions. The point estimates of the GP model predictions are further transformed into class probabilities for deciding the class label prediction. Both approaches do not consider the uncertainty around the predicted class probabilities which are used in decision-making regarding the class selection. This could lead to slow convergence of the classifier learning process or could result in a classifier model that is not accurately reliable. In this work, we propagate the uncertainty from the predictions of the posterior GP model to the class probability predictions using a linear error propagation rule. This propagated uncertainty is proposed to quantify the uncertainty associated with the BC problem. Further, we employ the propagated uncertainty in an AL framework to generate informative datasets to train the classifier model. This proposed novel AL strategy is compared with the existing methods (methods based on the approaches 1 and 2 discussed above) to demonstrate its usefulness in the context of AL.

2. METHODOLOGY

2.1 Bayesian binary classification with Gaussian processes

Binary classification involves the problem of assigning one of the two classes (0 or 1) to an input vector $\mathbf{x}$ by predicting the probability ϕ of class 1 given $\mathbf{x}$, $\phi = P(c = 1|\mathbf{x})$. This probability can be modelled as $\phi = g\left(h(\mathbf{x})\right)$, where $h(\mathbf{x})$ is a latent function, which is a mapping from the inputs $\mathbf{x}$ to continuous real values (latent function values) y, and g is the sigmoid transformation defined as $g(y) = 1/1 + e^{-y}$, which is a mapping from the latent function values y to the unit interval $[0,1]$. In Bayesian classification with Gaussian processes, a Gaussian process prior is placed over the latent function $h(\mathbf{x})$.

$$h(\mathbf{x}) \sim \mathcal{N}\left(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\right) \tag{1}$$

As discussed in [5], here the latent function h plays the role of a nuisance function, in the sense that we are not particularly interested in the values of h, but rather in $g\left(h(\mathbf{x})\right)$. We observe only the inputs $\mathbf{X}$ and the class labels $\mathbf{c}$. Let $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]^\intercal$ are the inputs corresponding to n data points. A GP prior is a collection of random variables $\{\mathbf{Y}(\mathbf{x})|\mathbf{x} \in \mathbf{X}\}$ indexed by the set $\mathbf{X}$, where any finite set of $\mathbf{Y}$ follows a joint multivariate Gaussian distribution. A GP prior is fully specified by its mean function $m(\mathbf{x}) = \mathrm{E}[\mathbf{Y}(\mathbf{x})]$ and its covariance function $k(\mathbf{x}, \mathbf{x}') = \mathrm{E}\left[\left(\mathbf{Y}(\mathbf{x}) - m(\mathbf{x})\right)\left(\mathbf{Y}(\mathbf{x}') - m(\mathbf{x}')\right)\right]$. For simplicity, we set the prior mean to zero, i.e., $m(\mathbf{x}) = 0$. For the covariance function, we employed the popular squared exponential covariance function with automatic relevance determination (ARD) distance measure. For the mathematical expression of the squared exponential kernel function, the readers are referred to [5]. We denote all the learned parameters of the covariance function that include the signal variance and the characteristic length scales by the hyperparameter vector $\boldsymbol{\theta}$. With this specification of prior mean and the covariance function, the prior over the latent function is jointly Gaussian. Let $y_{\mathrm{i}} = h(\mathbf{x}_{\mathrm{i}})$ and $\mathbf{y} = [y_1, \dots, y_{\mathrm{n}}]^\intercal$ denote the latent function values, then the prior takes the form:

$$p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{0}, \mathbf{K}) \tag{2}$$

where $\mathbf{K}$ is the $n \times n$ covariance matrix with elements defined by $\mathrm{K}_{ij} = k\left(\mathbf{x}_i, \mathbf{x}_j\right), i, j = 1, \dots, n$. Corresponding to each input data point $\mathbf{x}_i$, the probability of observing the class label 1 can be represented using a Bernoulli distribution. That is, given $\mathbf{x}_i$ and c_i, $\hat{c}_i = 1 \sim Bernoulli\left(g\left(h(\mathbf{x}_i)\right)\right)$, where $g\left(h(\mathbf{x}_i)\right) = \phi_i$ is the predicted probability of obtaining a class 1. For n independent data points, the likelihood $p(\mathbf{c}|\mathbf{y})$ can be derived from the probability mass function of the individual Bernoulli distributions as:

$$p(\mathbf{c}|\mathbf{y}) = \prod_{i=1}^{n} \left(g\left(h(\mathbf{x}_i)\right)\right)^{c_i} \left(1 - g\left(h(\mathbf{x}_i)\right)\right)^{1-c_i} \tag{3}$$

The posterior distribution over the latent function values $p(\mathbf{y}|\mathbf{X}, \mathbf{c}, \boldsymbol{\theta})$ can be computed from the prior and the likelihood using the Bayesian rule:

$$p(\mathbf{y}|\mathbf{X}, \mathbf{c}, \boldsymbol{\theta}) = \frac{p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta})}{p(\mathbf{c}|\mathbf{X}, \boldsymbol{\theta})} p(\mathbf{c}|\mathbf{y}) \tag{4}$$

where $p(\mathbf{c}|\mathbf{X}, \boldsymbol{\theta}) = \int p(\mathbf{c}|\mathbf{y})\, p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}) dy$ is the marginal likelihood. The evaluation of the posterior allows to make predictions $h(\mathbf{x}_*)$ for a new test point $\mathbf{x}_*$. However, the computation of the posterior distribution in Equation (4) is analytically intractable due to the non-Gaussian likelihood in Equation (3), which makes the evaluation of the integral on the right-hand side of the marginal likelihood impossible [5]. Owing to this, the training of the GP model (which involves maximizing the marginal likelihood) as well as the inference problem (making predictions based on the sample data, by using the learned hyperparameters) are evaluated using variational inference methods [7]. We employed evidence lower bound (ELBO) [7] as the variational inference loss function for GP model training.

2.2 Uncertainty in the inference

Common practices to quantify the uncertainty in the inference of BC with GP model involve either estimating the uncertainty as the variance over the latent function values or estimating it as the variance of the predicted classification outcomes. For any new input point $\mathbf{x}_*$, the variance over the latent function values can be computed as:

$$\sigma_y^2(\mathbf{x}_*) = k(\mathbf{x}_*, \mathbf{x}_*) - k_*^\intercal \mathbf{K}^{-1} k_* \tag{5}$$

where $k_* = [k(\mathbf{x}_1, \mathbf{x}_*), \dots, k(\mathbf{x}_n, \mathbf{x}_*)]^\intercal$. With the assumption that the predicted distribution of class labels is the same as the true distribution, the uncertainty of the classification outcome for any new input data point $\mathbf{x}_*$ can be computed as the random variance of the Bernoulli distribution corresponding to predicting a class 1 with a probability ϕ_*. This variance can be computed as:

$$\sigma_{\hat{c}}^2(\mathbf{x}_*) = \phi_*(1 - \phi_*) \tag{6}$$

2.3 The propagated uncertainty

In this work, we quantify the uncertainty around point estimates of the class probability predictions by propagating the uncertainty over the latent function values to the predicted probability values using the linear error propagation rule. The propagated uncertainty is computed as:

$$\sigma_\phi^2(\mathbf{x}_*) = \left(\frac{d\phi}{dy}\bigg|_{\mathbf{x}_*} \times \sigma_y(\mathbf{x}_*) \right)^2 = \left(\frac{e^y}{(1+e^y)^2} \right)^2 \times \sigma_y^2 \bigg|_{\mathbf{x}_*} \tag{7}$$

The uncertainty propagation is visually depicted in Figure 1.

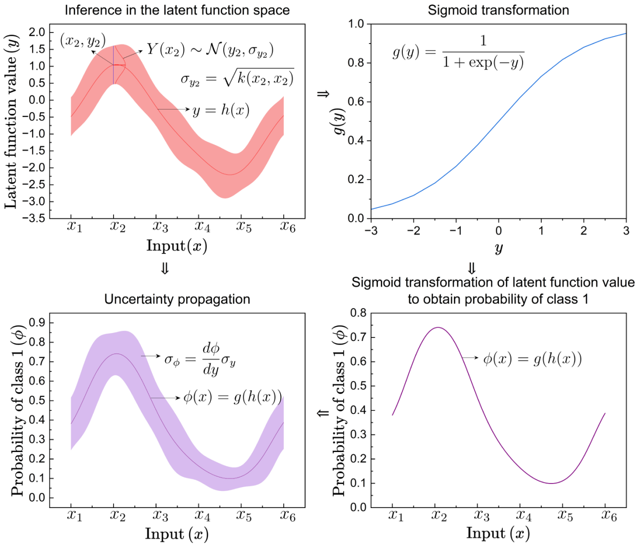

Figure 1. Uncertainty propagation to probability space.

2.4 Simulation study

A simulation study was designed to compare the performance of the three AL methods: **Method 1** using $\sigma_{\hat{c}}^2(\mathbf{x}_*)$ as the objective function (Equation (6)); **Method 2** using $\sigma_y^2(\mathbf{x}_*)$ as the objective function (Equation (5)) and **Method 3** using the proposed propagated uncertainty, $\sigma_\phi^2(\mathbf{x}_*)$ as the AL objective function (Equation 7). In all the three cases, the design of new experiments using the AL method involved seeking conditions where the corresponding objective function value is a maximum.

The steps involved in the simulation study are summarised in Figure 2. As explained in Figure 2, a classification dataset was first selected as the case study. The second step involved training of a selected classifier model on the full dataset selected in the first step. This trained model was assumed as the "True model" in the simulation study. Next, a limited number of preliminary experiments were designed using design of experiments (DoE) methods and the corresponding class labels were obtained by the simulation of the true model. The preliminary labelled dataset was used to train the GP model which is the test model to compare the AL methods. The GP model was subsequently trained using experiments designed through one of the AL methods.

It is often difficult to evaluate the performance of a classification model based on a single criterion. While accuracy is the simplest criterion, it could be misleading in case of imbalanced class problems. In addition, accuracy is a score based on the final class decision and not based on the statistical quality of model fitting. Therefore, in this work, the performance of the trained GP model was assessed using four different classification metrics: i) accuracy (%), ii) balanced accuracy (%), iii) ROC-AUC score, and iv) cross-entropy loss. The ROC-AUC score provides an aggregate performance measure of the classifier model across all possible threshold values on the predicted probabilities, which are ultimately used to decide class labels. The cross-entropy loss is a score that accounts for the statistical quality of the model fitting, evaluated with the expectation if the predicted class distribution is the same as the true class distribution (which is inferred from the sample data). A brief description of these metrics is provided in Table 1.

In the description of the accuracy and the balanced accuracy metrics, the variables TP, TN, FP and FN respectively indicate the number of true positive, true negative, false positive and false negative datapoints in the binary classification result. True positive are the datapoints that have been labelled as positive by the model (predicted label, $\hat{c} = 1$) and they are actually positive (observed label, $c = 1$), while false positive are the datapoints that have been labelled as positive by the model, but they are actually negative. The same logic applies to the definitions of true negative and false negative datapoints. The values of the metrics accuracy and balanced accuracy range between 0 and 100. The higher the values of these metrics, the better the performance of the classifier model. The ROC-AUC score is defined as the area under the receiver operating characteristic (ROC) curve [8]. The ROC-AUC score ranges from 0 to 1, with a value of 1 indicating perfect classification. The values

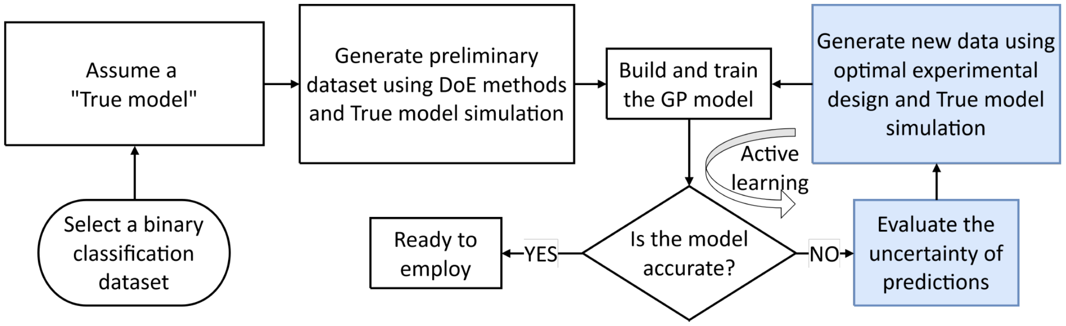
Assume a "True model"
Generate preliminary dataset using DoE methods and True model simulation
Build and train the GP model
Generate new data using optimal experimental design and True model simulation
Active learning
Select a binary classification dataset
Ready to employ
YES
Is the model accurate?
NO
Evaluate the uncertainty of predictions

Figure 2: Steps involved in the simulation study to compare the different active learning methods.

closer to 1 indicate better classification performance. The cross-entropy loss is defined as the negative log-likelihood of the classifier given the true class labels. A lower cross-entropy loss value indicates better model performance.

Table 1: Classification metrics used for performance evaluation.

Metrics	Description
Accuracy (%)	$\frac{TP + TN}{TP + TN + FP + FN} \times 100$
Balanced accuracy (%)	$\frac{1}{2}\left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP}\right) \times 100$
ROC-AUC score	Area under the ROC curve
Cross-entropy loss	$-\frac{1}{n}\sum_{i=1}^{n}[c_i \log \phi_i + (1 - c_i)\log(1 - \phi_i)]$

3. CASE STUDY

A classification dataset was selected from our recently published work on antisolvent precipitation of ibuprofen particles in a continuous flow precipitator [4]. The classification dataset published in this work consisted of three input variables, which are provided in Table 2. As explained in [4], the dataset was formed from the flow precipitation experiments carried out by varying the inputs provided in Table 2. The experiments resulted in three categories of outcome, which were labelled as: i) class 0 – infeasible experiments, which caused fouling in the flow channels and hindered the particle size measurements, ii) class 1 – partially feasible experiments that caused deposition of fine particles in the flow channels, but produced particle size measurements, and iii) class 2 – fully feasible experiments with no fouling issues. In this work, this multiclass classification dataset was converted to a binary classification problem by treating experiments with class label 1 also as infeasible experiments with label 0. Naturally, the class label 1 was then used to denote all the feasible experiments. The binary classification dataset formed in this way from the original multiclass classification dataset reported in [4] was trained on an SVM model with rbf kernel function. The trained SVM model was assumed as the true model.

Table 2: Input variables in classification dataset.

Input	Bound
Antisolvent flow rate (ml/min)	1 – 4
Antisolvent to solvent flow rate ratio (-)	1 – 9
Additive concentration (wt.%)	0 – 3

4. RESULTS AND DISCUSSION

In the simulation study, eight preliminary experiments were first designed using Sobol sampling method applied to the inputs and their bounds shown in Table 2. Class labels for the preliminary experiments were generated using the true model simulations. The GP model was then trained on the preliminary dataset. Then, twenty-two new experiments were designed one at a time using a specific AL method. To compare the three AL methods, the simulation study included three separate runs, each starting with the same DoE experiments but using different AL methods for the subsequent experiment designs. The results of the simulation study are summarised in Figure 3. In panel (a), the figure provides the scatter plots of marginal histograms of the input condition and the class labels. In each marginal histogram plot, the scatter plot represents the joint distribution of the specific input variable and the class labels. The horizontal histogram represents the class distribution, and the vertical histogram represents the distribution of the specific input variable. In Figure 3 (a), the histograms of Method 2 suggest that the method resulted in an imbalanced classification dataset, with most of the experiments concentrated in the extreme regions (extremely high and extremely low values of the input variables). Upon reviewing the dataset generated in Method 2, a possible explanation is that the

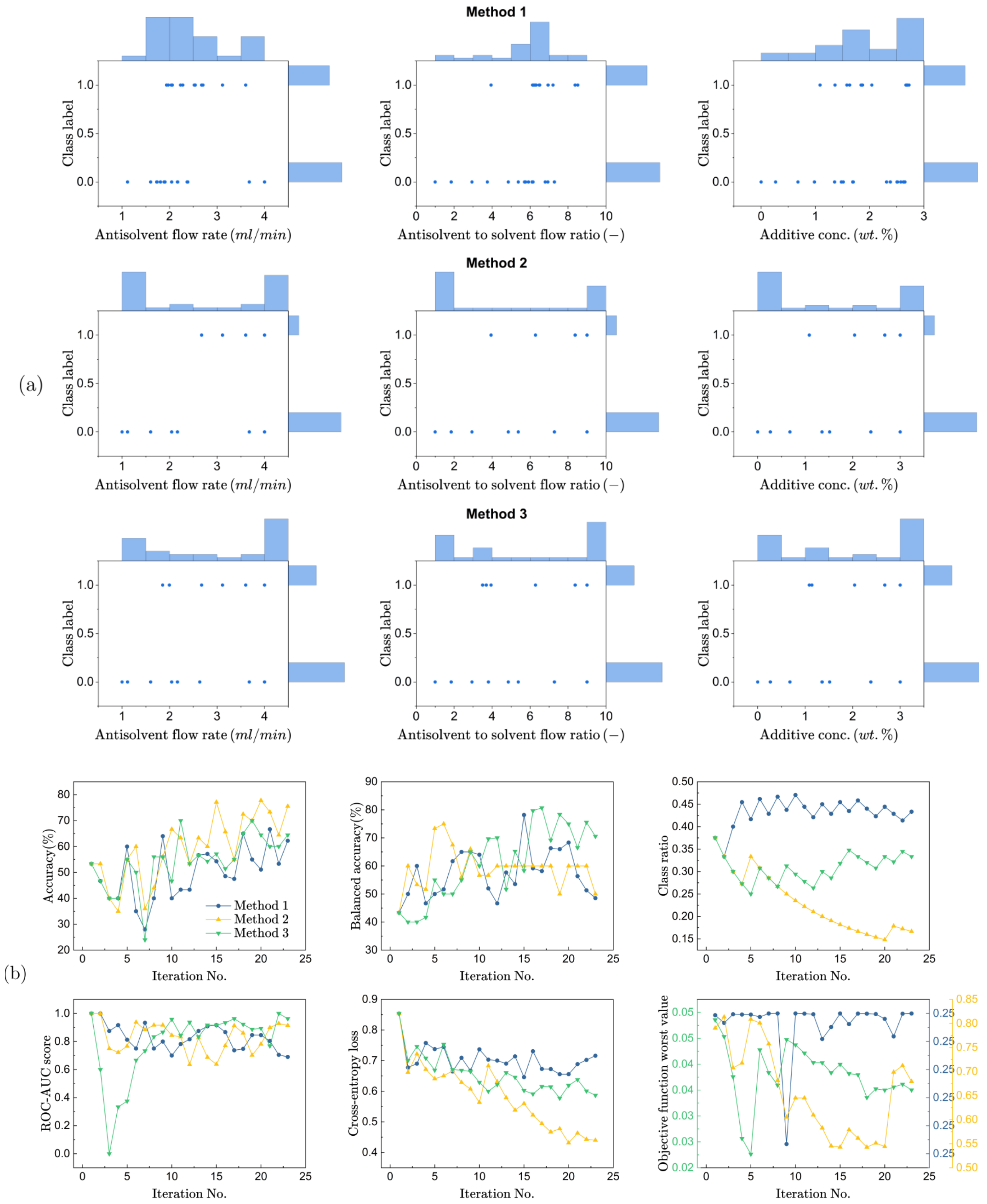

Figure 3: Comparison of the AL methods: (a) in terms of experimental design results and (b) in terms of the performance assessed using the classification metrics.

extreme regions were not explored in the initial experiments designed using Sobol sampling. This would have led to a higher variance at the extreme points, prompting Method 2 to focus on these regions. Figure 3 (a) also indicates that compared to Method 2, the other methods resulted in a more balanced classification, with Method 1 producing the best results. Moreover, we observe that in Methods 1 and 3, the experiments are more evenly distributed, and hence more explorative.

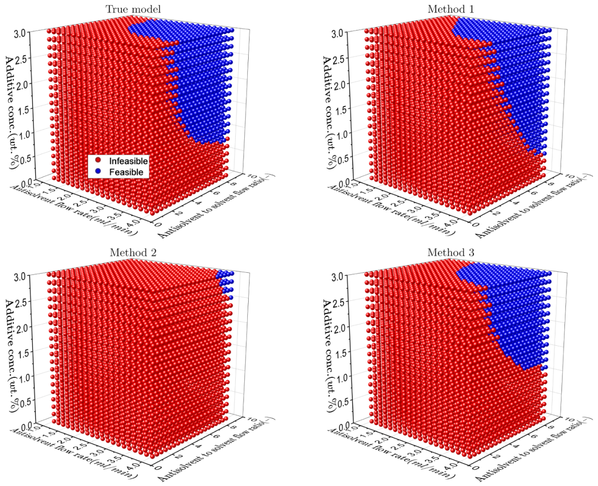

Figure 4. Simulations of the classifier models generated using the AL methods vs. the true model behaviour.

In Figure 3(b), the performance of classifier models developed using the three AL methods is evaluated using classification metrics averaged over five-fold cross-validation. While Method 2 shows higher classification accuracy, this is misleading due to the highly imbalanced dataset generated by this method. Balanced accuracy, which accounts for minority class contributions, shows how Method 3 performed best, scoring 70% compared to 48% for Method 1 and 50% for Method 2. Class ratios in Figure 3(b) and marginal distributions in Figure 3(a) reveal that Method 2 sampled many infeasible points, which were correctly classified but added limited learning value. This is evident from the poor decision boundary of Method 2 compared to the true model in Figure 4. Overall, Method 3 outperforms all the others by achieving the highest ROC-AUC score (0.96 compared to 0.91 for Method 2 and 0.69 for Method 1), good class balance (better than Method 2, but worse than Method 1), decreasing cross-entropy loss, and gradual reduction in the propagated uncertainty over the AL iterations. Its final classifier closely aligns with the true model, as shown in the simulation profiles of Figure 4.

5. CONCLUSION

We proposed a new method to quantify uncertainty in Bayesian classification. The proposed method focuses on propagating uncertainty from GP model predictions to the space of class probabilities. In a simulation study for binary classification, the method outperformed existing techniques in designing experiments that effectively train classifier models. Further studies will be needed to extend the applicability of the method to multiclass classification problems, and to validate its performance based on actual physical data.

ACKNOWLEDGEMENTS

This work was supported by the UK Engineering and Physical Sciences Research Council (EPSRC) [grant number EP/V050796/1].

REFERENCES

1. T. Gamer, M. Hoernicke, B. Kloepper, R. Bauer, and A. J. Isaksson, “The autonomous industrial plant – future of process engineering, operations and maintenance,” J Process Control, vol. 88, 2020.
2. C. K. I. Williams and D. Barber, “Bayesian classification with gaussian processes,” *IEEE Trans Pattern Anal Mach Intell*, vol. 20, no. 12, 1998.
3. B. Kompa, J. Snoek, and A. L. Beam, “Second opinion needed: communicating uncertainty in medical machine learning,” NPJ Digit Med, vol. 4, no. 1, 2021.
4. A. Pankajakshan et al., “MLAPI: A framework for developing machine learning-guided drug particle syntheses in automated continuous flow platforms,” Chem Eng Sci, vol. 302, p. 120780, Feb. 2025.
5. C. E. Rasmussen and C. K. I. Williams, Gaussian processes for machine learning. Cambridge, USA: The MIT Press, 2006.
6. D. D. Lewis and J. Catlett, “Heterogeneous Uncertainty Sampling for Supervised Learning,” in *Proceedings of the 11th International Conference on Machine Learning, ICML 1994*, 1994.
7. D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” in 2nd International Conference on Learning Representations, ICLR 2014.
8. Scikit-learn developers, “Scikit-learn: Machine Learning in Python.” Accessed: Dec. 30, 2024. [Online]. Available: https://scikit-learn.org/stable/modules/model_evaluation.html