Towards Assured Machine Learning for CPS: Identify "Unconfident" Regions

Xiaozhe Gu, Arvind Easwaran
Nanyang Technological University, Singapore
Email: guxi0002@e.ntu.edu.sg, arvinde@ntu.edu.sg

***Abstract*—Machine learning (*ML*) techniques are increasingly applied to decision-making and control problems in Cyber-Physical systems (CPS) among which many are safety-critical, e.g., chemical plants, robotics, autonomous vehicle, e.t.c. Despite the benefit brought by ML techniques, they also raise safety issues because 1) most models produced by ML algorithms are not transparent and behave as a block box and 2) the training data which plays a role as safety requirement is usually incomplete. An important technique for ML to achieve safety is "Safe Fail", i.e., a model rejects to predict and applies the backup solution when it has very low confidence in this prediction.**

Data-driven models produced by ML algorithms learn from training data and hence they are only as good as the examples that have learned. As observed in many previous studies, feature space that lack of data generally have a much higher error rate than regions that contain sufficient training samples [1]. Therefore, it is important for CPS containing ML components to classify such error-prone "unconfident" regions from "confident" regions such that predictions made in "unconfident" regions could be rejected. In this paper, we propose an efficient tree based classifier to address this problem. From the experiment results, we also show that ML models has much worse performance in the "unconfident" regions identified by our proposed technique.

I. Introduction

In this section, we can talk about

- The trend of ML in CPS and the need of safety AI.
- The safe fail technique, and the need to determine whether to reject predict.

A. Background: Prediction with Uncertainty

For classification problems, the output of ML models is usually interpreted as model confidence in that prediction. For example, output obtained at the end of the softmax layers of standard deep learning are often interpreted the predictive probabilities. If the obtained "confidence" level is very low, the models could select a reject option. A support vector machine (SVM) like classifier for classification with a reject option is proposed in [2] for binary classification problems. For ensemble classifiers, the votes of base classifiers [3] will be used to determine whether to reject a prediction.

The implicit assumption shared by classifiers is that ML models are most uncertain near the boundary between the different classes and the distance from the decision boundary is inversely related to the confidence that the instance belong to that class. This is reasonable to some extents because the decision boundaries learned by these models are usually located where a lot training samples belong to different classes overlap. However, for feature space $\mathcal{X}$ that contain few or no training samples at all, then the learned the decision boundary may be completely based on an inductive bias, thereby containing much epistemic uncertainty [4]. It is possible that an input instance that far from any training samples would be classified as some class with very high probability [5].

Prediction uncertainty can also be obtained by Bayesian inference. Bayesian probability theory provides a mathematically grounded tool to model the prediction uncertainty. Gaussian process (GP) [6] is a well known probabilistic model and assumes that $p(f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N))$ is jointly Gaussian, with some mean $\mu(\mathbf{x})$ and covariance $\Sigma(\mathbf{x})$. Given training samples $(\mathbf{X}, \mathbf{y})$ and unobserved instance $\mathbf{x}_*$, the output $\mathbf{f}_*$ is also conditional Gaussian $p(\mathbf{f}_*|x_*, \mathbf{X}, \mathbf{y}) = \mathcal{N}(\mu_*, \sigma_*)$ where the standard deviation can be interpreted as the prediction confidence. GP is computational intensive and has complexity $O(N^3)$, where N is the number of data. Bayesian methods can also be applied to neural networks (NNs). Infinitely wide single hidden layer NNs with distributions placed over their weights converge to Gaussian processes [7]. Variational inference [8], [9] can be used to obtain approximations for finite Bayesian neural networks. The dropout techniques in NNs can also be interpreted as a Bayesian approximation of Gaussian process [5]. With all the good properties of Bayesian learning, there are some controversial aspects: 1) is its reliance on priors and 2) it is computationally intensive.

The conformal prediction framework [10], [11] uses past experience to determine precise levels of confidence in new predictions. Given a certain error probability ϵ requirement, it forms a prediction interval $[\underline{f(\mathbf{x})}, \overline{f(\mathbf{x})}]$ for regression or a prediction label set $\{\text{Label } 1, \text{Label } 2, \ldots\}$ for classification so that the interval/set contains the actual prediction with probability $1-\epsilon$. However, its theoretically accuracy guarantee depends on the assumption that all the data are independent and identically distributed (in fact, i.i.d. assumption replaced by the weaker assumption of "exchangeability"). Besides, for regression problems, it tends to produce prediction bands whose width is roughly constant over the whole feature space [12].

B. Motivation

Data-driven models are trained by ML learning algorithms using a subset of possible scenarios that could be encountered operationally. Thus, the models produced by ML algorithms can only be as good as the examples that have learned. However, the training set is usually incomplete and there is no guarantee that it is even representative of the space of possible inputs [13]. Previous study [14] has demonstrated that feature space that lack of data generally have a much higher error rate.

Example 1: For example, Figure 1 shows the decision boundary learned by a SVM classifier to predict the whether a wall-following robot is turning right sharply. The value in the contour map represents the probability learned by the

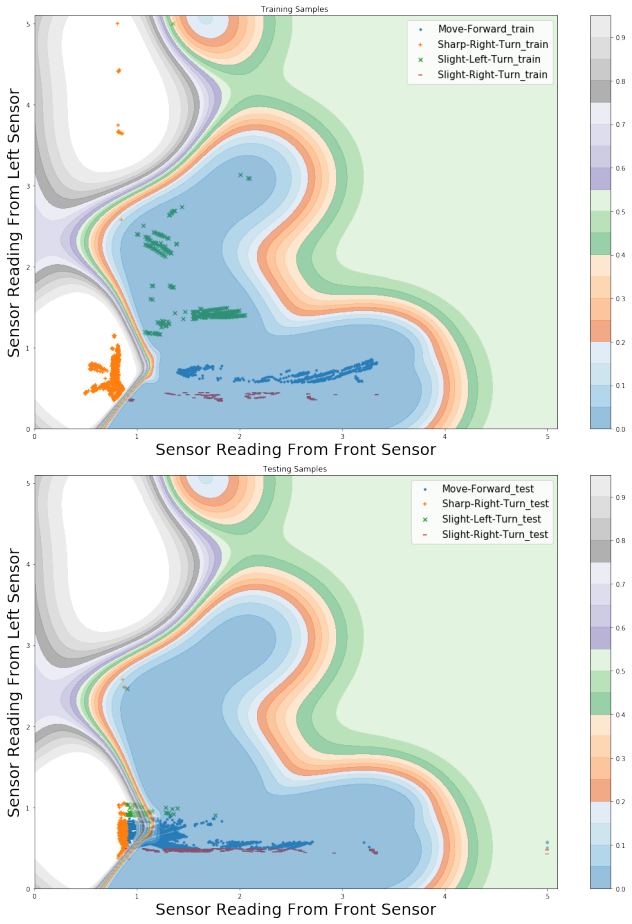

Fig. 1. Wall-following navigation task with mobile robot SCITOS-G5 based on sensor readings from the front and left sensor [15].

classifier that the instance belong to class "Sharp-Right-Turn". As we can observe, the training samples is not representative of testing samples at all. However, the classifier still has relative high confidence in some regions where it has not well learned but lots of many testing samples locates. As a result, a lot of testing samples that belong to other classes are misclassified as "Sharp-Right-Turn", and accuracy for testing samples decreases to 66% while the accuracy for training samples is almost 100%.

In practice, safety-critical scenarios like traffic accidents are very rare and ML algorithms usually do not receive enough such training samples.

C. Contribution

Therefore, it is very important to classify the feature space that the ML model is well trained from that it does not receive enough training samples. Systems with ML components should avoid making predictions in feature space where the model is not well trained if a false prediction could cause huge human and economic losses. Thus, in this paper, we propose a "monitor" used as a complement to check whether for ML models have encountered a testing sample from the "unconfident" feature space. We now outline a number of desired characteristics for such a "monitor".

Preferred Characteristics

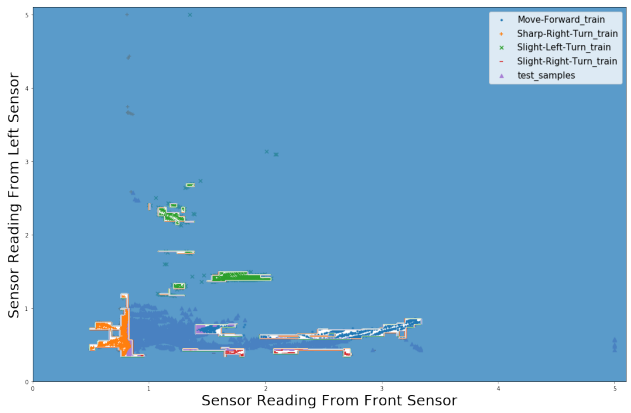

Fig. 2. Example: The "unconfident" regions for the Wall-following navigation task of robot SCITOS-G5 in Example 1

- The identified boundary of the"unconfident" regions in the feature space are preferred to be interpretable and understandable. The direct advantage of interpretable boundary is that, we could try to collect more samples from these regions (if possible) so that the model's performance in these regions could be improved.
- The decision whether a new input instance belongs to "unconfident" regions must the determined efficiently. Note that, the model will be used as a complement to monitor whether ML models have encountered any input instances from their "unconfident" regions. Thus, especially for control problems, the additional overhead from such a "monitor" should be as small as possible.
- After the feature space is partitioned into multiple regions, there exist some metrics to compare between these regions and select the "unconfident" ones.

In this paper, we propose a efficient technique to partition the feature space into multiple hyperrectangles based on classification and regression tree [16]. The data density in these hyperrectangles is then used to determine the threshold that whether hyperrectangle should be considered as "unconfident". In Figure 2, we show the resulting identified "unconfident" regions for the wall-following navigation task in Example 1. Since most testing samples are in the "unconfident" regions, the classifier should avoid making predictions for these samples. Finally, from the experiment results in Section II, we can observe that, ML models have much higher losses/error rates in the "unconfident" regions identified by the proposed technique.

II. Evaluation

In this section, we evaluate the effectiveness of our proposed technique by showing that different ML models (e.g., SVM, NNs, GP) have much higher loss (or error rates) in the identified "unconfident" regions. However, one thing we should point out is that, for classification problems, "unconfident" regions do not always mean lower error rates. For example, Figure 3 shows a binary classification problem where the decision boundary is a circle. The SVM classifier has very high accuracy in the feature space far from the circle decision boundaries, even though it has learned very few or no training samples from these regions.

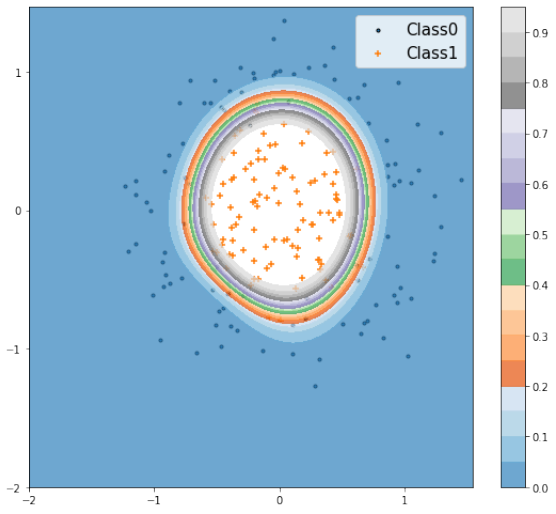

Fig. 3. A binary classification toy example

A. Classification Problems

1) The Navigation Task for Mobile Robot SCITOS-G5 [15]:

B. Regression Problems

1) The Inverse Dynamics of a SARCOS Robot Arm [17]: The task is to map from a 21-dimensional input space (7 joint positions, 7 joint velocities, 7 joint accelerations) to the corresponding torque.

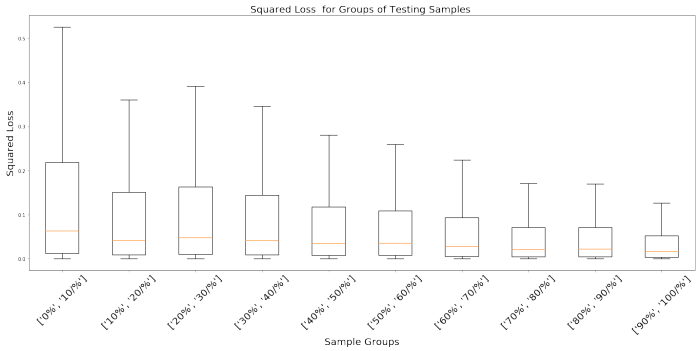

Fig. 4. SARCO: box plot of mean squared loss of NNs

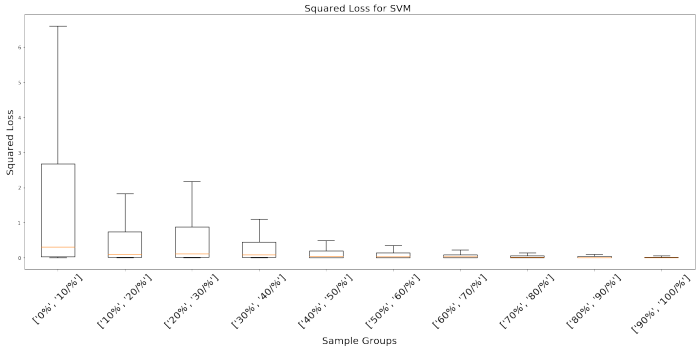

Fig. 5. SARCO: box plot of mean squared loss of SVM

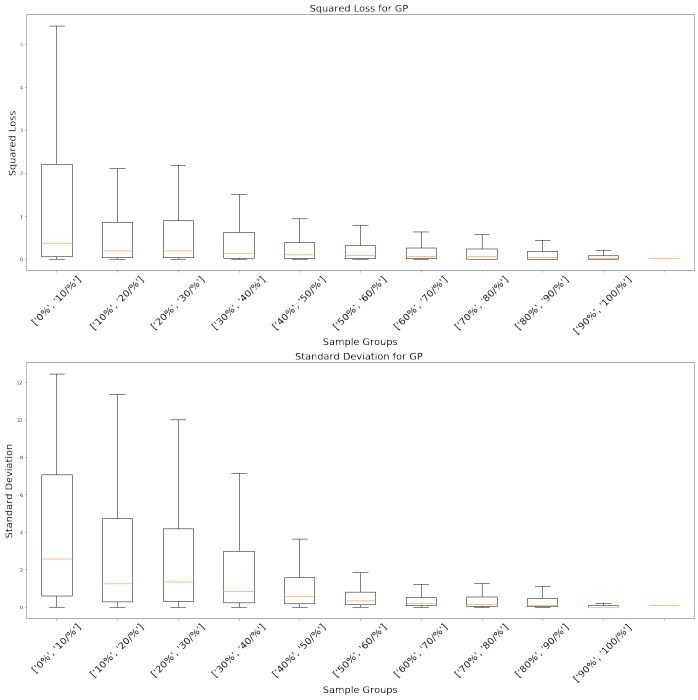

Fig. 6. SARCO: box plot of mean squared loss and standard deviation of GP

ACKNOWLEDGMENT

III. CONCLUSION

REFERENCES

[1] G. M. Weiss, "Mining with rarity: a unifying framework," *ACM Sigkdd Explorations Newsletter*, vol. 6, no. 1, pp. 7–19, 2004.

[2] P. L. Bartlett and M. H. Wegkamp, "Classification with a reject option using a hinge loss," *Journal of Machine Learning Research*, vol. 9, no. Aug, pp. 1823–1840, 2008.

[3] K. R. Varshney, R. J. Prenger, T. L. Marlatt, B. Y. Chen, and W. G. Hanley, "Practical ensemble classification error bounds for different operating points," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 11, pp. 2590–2601, 2013.

[4] J. Attenberg, P. Ipeirotis, and F. Provost, "Beat the machine: Challenging humans to find a predictive model's “unknown unknowns”," *J. Data and Information Quality*, vol. 6, no. 1, pp. 1:1–1:17, 2015.

[5] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*, 2016, pp. 1050–1059.

[6] M. Seeger, "Gaussian processes for machine learning," *International journal of neural systems*, vol. 14, no. 02, pp. 69–106, 2004.

[7] R. M. Neal, *Bayesian learning for neural networks*. Springer Science & Business Media, 2012, vol. 118.

[8] J. Paisley, D. Blei, and M. Jordan, "Variational bayesian inference with stochastic search," *arXiv preprint arXiv:1206.6430*, 2012.

[9] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[10] V. Vovk, A. Gammerman, and G. Shafer, *Algorithmic Learning in a Random World*. Berlin, Heidelberg: Springer-Verlag, 2005.

[11] V. Vovk, I. Nouretdinov, A. Gammerman *et al.*, "On-line predictive linear regression," *The Annals of Statistics*, vol. 37, no. 3, pp. 1566–1590, 2009.

[12] J. Lei, M. G'Sell, A. Rinaldo, R. J. Tibshirani, and L. Wasserman, "Distribution-free predictive inference for regression," *Journal of the American Statistical Association*, pp. 1–18, 2018.

[13] R. Salay, R. Queiroz, and K. Czarnecki, "An analysis of ISO 26262: Using machine learning safely in automotive software," *CoRR*, vol. abs/1709.02435, 2017. [Online]. Available: http://arxiv.org/abs/1709.02435

[14] G. M. Weiss, "Learning with rare cases and small disjuncts," in *Machine Learning Proceedings 1995*. Elsevier, 1995, pp. 558–565.

[15] D. Dheeru and E. Karra Taniskidou, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[16] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and regression trees*, ser. The Wadsworth statistics/probability series. Wadsworth & Brooks/Cole Advanced Books & Software, 1984.

[17] "Sarcos." [Online]. Available: http://www.gaussianprocess.org/gpml/data/