ALT-MAS: A Data-Efficient Framework for Active Testing of Machine Learning Algorithms

Huong Ha[*]
RMIT University
Australia

Sunil Gupta
Deakin University
Australia

Santu Rana
Deakin University
Australia

Svetha Venkatesh
Deakin University
Australia

ABSTRACT

Machine learning models are being used extensively in many important areas, but there is no guarantee a model will always perform well or as its developers intended. Understanding the correctness of a model is crucial to prevent potential failures that may have significant detrimental impact in critical application areas. In this paper, we propose a novel framework to efficiently test a machine learning model using only a small amount of labeled test data. The idea is to estimate the metrics of interest for a model-under-test using Bayesian neural network (BNN). We develop a novel data augmentation method helping to train the BNN to achieve high accuracy. We also devise a theoretic information based sampling strategy to sample data points so as to achieve accurate estimations for the metrics of interest. Finally, we conduct an extensive set of experiments to test various machine learning models for different types of metrics. Our experiments show that the metrics estimations by our method are significantly better than existing baselines.

1 INTRODUCTION

Today, supervised machine learning models are employed across sectors to assist humans in making important decisions. Understanding the correctness of a model is thus crucial to avoid potential (and severe) failures. In practice, however, it is not always possible to accurately evaluate the model's correctness using the held-out training data in the development process (Sawade et al., 2010). Consider a hospital that buys an automated medical image classification system. The supplier will provide a performance assessment, but this evaluation may not hold in this new setting as the supplier and the hospital data distributions may differ. Similarly, an enterprise that develops a business prediction system might find that the performance changes significantly over time as the input distribution shifts from the original training data. In these cases, the model performance needs to be re-evaluated as the assessments provided from the supplier or from the development process can be inaccurate. To accurately evaluate the model performance, new labeled data points from the deployment area are needed. But the process of labelling is expensive as one would usually need a large number of test instances. *Thus the open question is how to test the performance of a machine learning model (model-under-test) with parsimonious use of labeled data from the deployment area.*

This work focuses on addressing this challenge treating the model-under-test as a black-box as in common practice one only has access to the model outputs. We propose a *data-efficient framework* that can assess the performance of a black-box model-under-test on any metric (that is applicable for black-box models) without prior knowledge from users. Furthermore, our framework can estimate multiple metrics simultaneously. The motivation for evaluating one or multiple metrics is inspired by the current practice of users who need to assess the model-under-test on one or varied aspects that are important to them. For instance, for a classification system, the user might want to solely check the overall accuracy or simultaneously check the overall accuracy, macro-precision (recall) and/or the accuracies of some classes of interest.

To achieve sample efficiency, we formulate our testing framework as an active learning (AL) problem (Cohn et al., 1994). First, a small subset of the test dataset is labeled, and a surrogate model

---

[*]Correspondence to: `huong.ha@rmit.edu.au`

is learned from this subset to predict the ground truth of unlabeled data points in the test dataset. Second, an acquisition function is constructed to decide which data point in the test dataset should be chosen for labelling. The data point selected by the acquisition function is sent to an external oracle for labelling, and is added to the labeled set. The process is conducted iteratively until the labelling budget is depleted. The metrics of interest are estimated using the learned surrogate model.

With this framework, one choice is to use a standard AL method to learn a surrogate model that accurately predicts the labels of all data points in the test dataset, however, this choice is not optimal. To efficiently estimate the metrics of interest, the surrogate model should not need to accurately predict the labels of all data points; it only needs to accurately predict the labels of those data points that contribute significantly to the accuracy of the metric estimations. For our active testing framework, we first propose a data augmentation method that helps to train the surrogate model to achieve high accuracy. Second, we derive a theoretic information based acquisition function that can select data points for whom labels should be acquired so as to enable maximal estimation accuracy of the metric of interest. We then generalize our framework to be able to work with multiple metrics. Finally, we demonstrate the efficacy of our proposed testing framework using various models-under-test and a wide range of metric sets. In summary, our main contributions are:

1. ALT-MAS, a data-efficient testing framework that can accurately estimate the performance of a machine learning model;

2. A novel data augmentation strategy to train BNNs to achieve high accuracy, and a novel sampling methodology to estimate the metrics of interest accurately and efficiently; and,

3. Demonstration of the empirical effectiveness of our proposed machine learning testing framework on various models-under-test for a wide range of metrics.

Related work   There have been various research works tackling the problem of evaluating machine learning model performance using limited labeled data. However, to the best of our knowledge, the methods proposed in all these works, except Druck & McCallum (2011) and Welinder et al. (2013), can only work with some specific metrics such as accuracy and precision-recall curve. The approaches in Druck & McCallum (2011) and Welinder et al. (2013) can estimate any metrics like ours, however, they require information from the classifier internal structure, particularly the softmax outputs. Further details regarding related work are in App. A.

2   PROBLEM FORMULATION

Let us assume we are given a black-box model-under-test $\mathcal{A}$ that gives the prediction $\mathcal{A}(x)$ for an input x, with $\mathcal{A}(x) \in \mathcal{C} = \{1, \ldots, C\}$. Let us also assume we have access to (i) an unlabeled test dataset $\mathcal{X} = \{x_i\}_{i=1}^N$, and, (ii) an oracle that can provide the label y_x for each input x in $\mathcal{X}$. Given a set of performance metrics $\{Q_k\}_{k=1}^K, Q_k : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}$, the goal is to *efficiently* estimate the values of these metrics, $\{Q_k^*\}_{k=1}^K$, when evaluating the model-under-test $\mathcal{A}$ on the test dataset $\mathcal{X}$. That is, we aim to estimate,

$$Q_k^* = Q_k(\mathcal{A}_\mathcal{X}, \mathcal{Y}_\mathcal{X}), \quad k = 1, \ldots, K, \tag{1}$$

with $\mathcal{A}_\mathcal{X} = \{\mathcal{A}(x)\}_{x \in \mathcal{X}}$ and $\mathcal{Y}_\mathcal{X} = \{y_x\}_{x \in \mathcal{X}}$, using the minimal number of oracle queries. Note we focus on classifiers because they are common supervised learning models and also the target models of most machine learning testing papers (Druck & McCallum, 2011; Welinder et al., 2013; Sabharwal & Sedghi, 2017; Jiang et al., 2018; Zhang et al., 2019).

3   ALT-MAS: ACTIVE TESTING WITH METRIC-AWARE SAMPLING

For our active testing framework, BNN is chosen as the surrogate model due to its effectiveness and scalability. In this section, we propose methods to (i) augment the training data so that the BNN can achieve high accuracy from a limited number of labeled data (**Section 3.1**), (ii) estimate the metrics of interest given the BNN (**Section 3.2**), and (iii) sample the most informative data point to maximize the estimation accuracy of a specific metric (**Section 3.3**) or a set of metrics (**Section 3.4**).

3.1 DATA AUGMENTATION STRATEGY

At each iteration t, we train a binary classifier $\mathcal{C}_\eta$ aiming to predict the data points (in the test dataset $\mathcal{X}$) whose model-under-test predictions agree with the ground-truth. Using the predictions by $\mathcal{C}_\eta$, we then construct an *augmented labeled set* $\mathcal{S}_l^t = \{\mathcal{X}_S^t, \mathcal{Y}_{\mathcal{X}_S^t}\}$ where $\mathcal{X}_S^t$ are all data points in $\mathcal{X}$ which $\mathcal{C}_\eta$ identifies that the model-under-test predictions are accurate, and $\mathcal{Y}_{\mathcal{X}_S^t}$ are the corresponding model-under-test outputs of $\mathcal{X}_S^t$. The BNN is then trained using both the labeled set $\mathcal{D}_l^t$ and the augmented labeled set $\mathcal{S}_l^t$. Further details on how to train the binary classifier is in App. B.2.

3.2 METRIC ESTIMATION WITH BAYESIAN NEURAL NETWORK

At iteration t, using the predicted posterior distribution $q_\theta(\omega|\mathcal{D}_l^t, \mathcal{S}_l^t)$ provided by the trained BNN with weights ω, we can approximate the value of a metric Q_k as,

$$\hat{Q}_k^t = \int Q_k(\mathcal{A}_\mathcal{X}, [\mathcal{Y}_{\mathcal{X}_l^t}, \hat{\mathcal{Y}}_{\mathcal{X}_{ul}^t,\omega}]) q_\theta(\omega|\mathcal{D}_l^t, \mathcal{S}_l^t)\,\mathrm{d}\omega \approx \frac{1}{M} \sum_{j=1}^{M} Q_k(\mathcal{A}_\mathcal{X}, [\mathcal{Y}_{\mathcal{X}_l^t}, \hat{\mathcal{Y}}_{\mathcal{X}_{ul}^t,\hat{\omega}_j}]), \quad (2)$$

where $\mathcal{X}_l^t, \mathcal{X}_{ul}^t$ denote the sets of labeled and unlabeled data points in $\mathcal{X}$ respectively, $\hat{\mathcal{Y}}_{\mathcal{X}_{ul}^t,\omega}$ denotes the predicted labels of $\mathcal{X}_{ul}^t$ using the BNN weights ω, and $\{\hat{\omega}_j\}_{j=1}^{M}$ are M stochastic forward passes from the BNN predicted posterior distribution $q_\theta(\omega|\mathcal{D}_l^t, \mathcal{S}_l^t)$.

3.3 SAMPLING METHODOLOGY FOR A SINGLE METRIC

At iteration t, the sampling process aims to select a data point x_t^* to label so as to maximally increase the metric estimation accuracy. This can be considered equivalent to sampling the data point x_t^* such that knowing its label results in maximal uncertainty reduction for the metric estimation. To identify x_t^*, our idea is to (i) evaluate how much the uncertainty of each data point's label contributes to the uncertainty of the metric estimation, and, (ii) sample the data point causing the highest uncertainty in the metric estimation. **To solve (i)**, we define a new random variable (r.v.) $\tilde{Q}_k(x)$ as follows,

$$\tilde{Q}_k(x) = \mathbb{E}_{\hat{\mathcal{Y}}_{\mathcal{X}_{ul}^t \setminus x} \sim p(\hat{\mathcal{Y}}_{\mathcal{X}_{ul}^t \setminus x}|\mathcal{X}_{ul}^t \setminus x, \mathcal{D}_l^t)} [Q_k(\mathcal{A}_\mathcal{X}, [\mathcal{Y}_{\mathcal{X}_l^t}, \hat{\mathcal{Y}}_{\mathcal{X}_{ul}^t \setminus x}, \hat{y}_x])], \quad \forall x \in \mathcal{X}_{ul}^t, \quad (3)$$

where $\hat{\mathcal{Y}}_{\mathcal{X}_{ul}^t \setminus x}$ denotes the r.v. representing the possible labels of $\mathcal{X}_{ul}^t \setminus x$ conditioned on $\mathcal{D}_l^t$. This r.v. is constructed based on Eq. (2), but we set the data point of interest, x, as a variable while fixing other data points, $\mathcal{X}_{ul}^t \setminus x$, as their expected values. Thus, $\tilde{Q}_k(x)$ represents all the possible values of metric Q_k for each data point x when y_x varies.

To solve (ii), we formulate the acquisition function as the mutual information (Houlsby et al., 2011; Gal et al., 2017) between $\tilde{Q}_k(x)$ and the BNN weights ω, i.e.,

$$\mathbb{I}[\tilde{Q}_k(x); \omega|x, \mathcal{D}_l^t] = \mathbb{H}[\tilde{Q}_k(x)|x, \mathcal{D}_l^t] - \mathbb{E}_{\omega \sim p(\omega|\mathcal{D}_l^t)}[\mathbb{H}[\tilde{Q}_k(x)|x, \omega]]. \quad (4)$$

This formulation can be approximated via the BNN posterior distribution $q_\theta(\omega|\mathcal{D}_l^t, \mathcal{S}_l^t)$ using the similar methodology as in Gal et al. (2017). Detailed formulas are in App. B.3.

Finally, the data point to be sampled is the one with the highest value of $\mathbb{I}[\tilde{Q}_k(x); \omega|x, \mathcal{D}_l^t]$, i.e.,

$$x_{t+1}^* = \mathrm{argmax}_{x \in \mathcal{X}_{ul}^t} \mathbb{I}[\tilde{Q}_k(x); \omega|x, \mathcal{D}_l^t]. \quad (5)$$

3.4 SAMPLING METHODOLOGY FOR MULTIPLE METRICS

Given a set of multiple metrics $\{Q_k\}_{k=1}^{K}$, extending the idea in Sec. 3.3 to multiple metrics, we can sample the data point with the highest sum of $\mathbb{I}[\tilde{Q}_k(x); \omega|x, \mathcal{D}_l^t]$, i.e.,

$$x_{t+1}^* = \mathrm{argmax}_{x \in \mathcal{X}_{ul}^t} \sum_{k=1}^{K} \mathbb{I}[\tilde{Q}_k(x); \omega|x, \mathcal{D}_l^t], \quad (6)$$

4 EXPERIMENTAL RESULTS

We evaluate **ALT-MAS** using various models-under-test and metric sets on MNIST dataset. Our experiments aim to answer the following questions: (Q.1) *Does our active testing framework perform better than traditional machine learning testing approaches?* (Q.2) *Does our active testing*
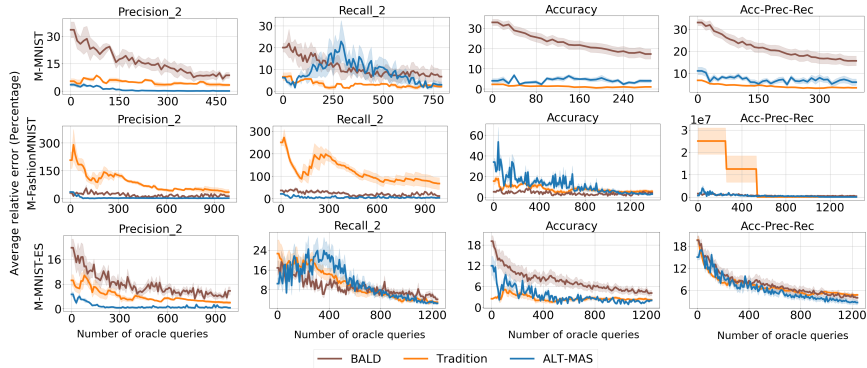
Figure 1: Average relative error of the estimated metrics, for each combination of model-under-test (*M-MNIST*, *M-FashionMNIST*, & *M-MNIST-ES*) and metric set. Plotting mean and standard error over 3 repetitions. Better methods are those that converge faster to zero (Best seen in color).

framework perform better than existing AL approaches when estimating metric values? (Q.3) *Does our proposed sampling method work well for different sets of metrics and models-under-test?* and (Q.4) *How accurate the BNNs predictions are?*

To answer Q.1 & Q.2, we compare **ALT-MAS** with two baselines: (i) **Tradition**: the metric values are computed using their mathematical formula with all the labeled data - that is picked randomly from the whole test dataset - up to the current iteration; (ii) **BALD**: the metric values are estimated using the predicted label of the BNN trained with BALD - the state-of-the-art deep Bayesian AL method (Gal et al., 2017). Note we don't compare with traditional AL methods as it has been shown deep Bayesian AL methods outperform traditional AL methods (Gal et al., 2017). To answer Q.3, for each dataset, we evaluate all methods using various models-under-test and metric sets. In particular, we use three different models-under-test with different levels of accuracy: *M-MNIST* trained on MNIST training set (high accuracy), *M-MNIST-ES* trained on MNIST training set with early stopping (average accuracy), and *M-FashionMNIST* trained on FashionMNIST (Xiao et al., 2017) - a completely different dataset (low accuracy). The metric sets consist of either a single or multiple metrics, and the metrics are of different types: per-class metrics and overall metrics. The BNN we use is a simple MLP model with 2 layers, 256 neurons/layer, and with dropout applied in each layer. For each combination of model-under-test and metric set, we repeat the experiment 3 times. Further details on the experiment set up and the answers to Q.4 are in App. C.

In Fig. 1, we report the average relative errors of the estimated metrics for all combinations of model-under-test and metric set. First, we can see that **Tradition** is not consistent; it performs very well on *M-MNIST* but performs badly on *M-FashionMNIST*. Based on our observations, the reason is that *M-MNIST* has very high accuracy (92.8%), which means its outputs and the true labels are very similar. Thus, randomly picking data points and estimating the metric values via their mathematical formulas results in the metric estimations to be approximately 90-95%, which are similar to the true values of the metrics. However, for other models-under-test with lower accuracies, this technique will not work. Now let us compare **ALT-MAS** and **BALD**. For most cases, **ALT-MAS** outperforms **BALD**, especially on per-class metric and *M-MNIST*. This is understandable as i) *M-MNIST* has high accuracy and thus the BNNs trained with **ALT-MAS** using our data augmentation method have much higher accuracies compared to those with **BALD**, and, ii) the optimal sampling strategy to achieve accurate estimations for per-class metrics is very different to the optimal sampling strategy to obtain a BNN with high accuracy.

5   CONCLUSION

We propose a novel approach to efficiently evaluate the performance of black-box machine learning models. The core idea is to efficiently estimate important metrics of the model being tested based on BNN. We develop a novel data augmentation method for training the BNN to achieve accurate metric estimations. We also devise a novel theoretic information based sampling strategy to sample data points such that the proposed framework can accurately estimate the metrics of interest simultaneously using a minimal number of labeled data. Experimental results show that our proposed approach works efficiently for estimating multiple metrics and diverse models-under-test.

REFERENCES

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning (ICML)*, pp. 1613–1622, 2015.

David A. Cohn, Zoubin Ghahramani, and Michael I. Jordan. Active learning with statistical models. pp. 705–712, 1994.

Juan Gabriel Colonna, Tanel Peet, Carlos Abreu Ferreira, Alípio Mário Jorge, Elsa Ferreira Gomes, and João Gama. Automatic classification of anuran sounds using convolutional neural networks. In *Proceedings of the Ninth International C* Conference on Computer Science & Software Engineering, C3S2E '16, Porto, Portugal, July 20-22, 2016*, pp. 73–78. ACM, 2016.

Charles Corbière, Nicolas Thome, Avner Bar-Hen, Matthieu Cord, and Patrick Pérez. Addressing failure prediction by learning model confidence. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 2898–2909. 2019.

Gregory Druck and Andrew McCallum. Toward interactive training and evaluation. In *Proceedings of the 20th ACM Conference on Information and Knowledge Management, CIKM 2011, Glasgow, United Kingdom, October 24-28, 2011*, pp. 947–956. ACM, 2011.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pp. 1050–1059. JMLR.org, 2016.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1183–1192. PMLR, 2017.

Shivapratap Gopakumar, Sunil Gupta, Santu Rana, Vu Nguyen, and Svetha Venkatesh. Algorithmic assurance: An active approach to algorithmic testing using bayesian optimisation. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 5470–5478, 2018.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1321–1330. PMLR, 2017.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

Neil Houlsby, Ferenc Huszar, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *CoRR*, abs/1112.5745, 2011. URL http://arxiv.org/abs/1112.5745.

Heinrich Jiang, Been Kim, Melody Y. Guan, and Maya R. Gupta. To trust or not to trust A classifier. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 5546–5557. 2018.

Namit Katariya, Arun Shankar Iyer, and Sunita Sarawagi. Active evaluation of classifiers on large datasets. In *12th IEEE International Conference on Data Mining, ICDM 2012, Brussels, Belgium, December 10-13, 2012*, pp. 329–338, 2012.

M. J. Keith, A. Jameson, W. van Straten, M. Bailes, S. Johnston, M. Kramer, A. Possenti, S. D. Bates, N. D. R. Bhat, M. Burgay, S. Burke-Spolaor, N. D'Amico, L. Levin, Peter L. McMahon, S. Milia, and B. W. Stappers. The High Time Resolution Universe Pulsar Survey – I. System configuration and initial discoveries. *Monthly Notices of the Royal Astronomical Society*, 409(2): 619–627, 11 2010.

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 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 7024–7035. 2019.

Anurag Kumar and Bhiksha Raj. Classifier risk estimation under limited labeling resources. In *Advances in Knowledge Discovery and Data Mining - 22nd Pacific-Asia Conference, PAKDD 2018, Melbourne, VIC, Australia, June 3-6, 2018, Proceedings, Part I*, volume 10937, pp. 3–15. Springer, 2018.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 6402–6413. 2017.

Yann LeCun, Corinna Cortes, and Christopher J.C. Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 2, 2010.

David J. C. MacKay. A practical bayesian framework for backpropagation networks. *Neural Computation*, 4(3):448–472, May 1992.

Radford M. Neal. *Bayesian learning for neural networks*. PhD thesis, University of Toronto, 1995.

Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In Luc De Raedt and Stefan Wrobel (eds.), *Machine Learning, Proceedings of the Twenty-Second International Conference (ICML 2005), Bonn, Germany, August 7-11, 2005*, volume 119 of *ACM International Conference Proceeding Series*, pp. 625–632. ACM, 2005.

John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, pp. 61–74. MIT Press, 1999.

Ashish Sabharwal and Hanie Sedghi. How good are my predictions? efficiently approximating precision-recall curves for massive datasets. In *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2017, Sydney, Australia, August 11-15, 2017*. AUAI Press, 2017.

Ashish Sabharwal and Yexiang Xue. Adaptive stratified sampling for precision-recall estimation. In *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, UAI 2018, Monterey, California, USA, August 6-10, 2018*, pp. 825–834. AUAI Press, 2018.

Christoph Sawade, Niels Landwehr, Steffen Bickel, and Tobias Scheffer. Active risk estimation. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, pp. 951–958. Omnipress, 2010.

Sebastian Schelter, Tammo Rukat, and Felix Bießmann. Learning to validate the predictions of black box classifiers on unseen data. In *Proceedings of the 2020 International Conference on Management of Data, SIGMOD Conference 2020, online conference [Portland, OR, USA], June 14-19, 2020*, pp. 1289–1299. ACM, 2020.

Burr Settles. Active learning literature survey. Technical report, Computer Science Department, University of Wisconsin–Madison, 2010.

Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pp. 2960–2968, 2012.

Peter Welinder, Max Welling, and Pietro Perona. A lazy man's approach to benchmarking: Semisupervised classifier evaluation and recalibration. In *2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, OR, USA, June 23-28, 2013*, pp. 3262–3269. IEEE Computer Society, 2013.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017. URL http://arxiv.org/abs/1708.07747.

Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July 23-26, 2002, Edmonton, Alberta, Canada*, pp. 694–699. ACM, 2002.

Jie M. Zhang, Mark Harman, Lei Ma, and Yang Liu. Machine learning testing: Survey, landscapes and horizons. *CoRR*, abs/1906.10742, 2019. URL http://arxiv.org/abs/1906.10742.

A   RELATED WORK

Classifier Performance Evaluation with Limited Labeled Data   There have been various research works tackling the problem of evaluating classifier performance on an unlabeled test dataset using a limited number of labeled data and a labelling oracle. However, most of these works have focused only on some specific metrics such as accuracy and precision-recall curve. For instance, the approach in Sawade et al. (2010) aims to estimate a risk score, which is a function of the model-under-test output and the ground-truth (akin to metric), by deriving a sampling distribution that minimizes the variance of the risk score. The data points to be sent to the oracle for labelling is selected using this sampling distribution. But, to the best of our knowledge, this approach has only been shown to be tractable when the risk score is the accuracy metric. With the same goal of efficiently estimating classifiers accuracies, Katariya et al. (2012) propose a hashing based stratified sampling method whilst Kumar & Raj (2018) derive a stratified sampling strategy that offers optimal variance reduction in the accuracy estimation. To estimate precision-recall or ROC curves of classifiers, Sabharwal & Sedghi (2017) suggested PAULA, a precision approximation method using logarithmic annotations, and then ADASTRAT (Sabharwal & Xue, 2018), an adaptive stratified sampling approach. On the other hand, the methods in Druck & McCallum (2011) and Welinder et al. (2013) can estimate any types of metrics like ours, however, they require information from the classifier internal structure, particularly the softmax outputs. Specifically, Druck & McCallum (2011) propose to use the softmax output for constructing the stratification function whilst Welinder et al. (2013) use this information to compute the sampling distribution. More recently, Gopakumar et al. (2018) suggested to search for the worst case model performance using limited labeled data, however, we posit that using worst case to assess the goodness of a model-under-test is an overkill because the worst case is often just an outlier. The work in Schelter et al. (2020) learns to validate the model without labeled data by generating a synthetic dataset representative of the deployment data. The restrictive assumption is that it requires domain experts to provide a set of data generators, a task usually infeasible in reality.

Trustworthy Machine Learning. This line of works aims to assess the confidence of deep neural networks in making predictions on test data points (Platt, 1999; Zadrozny & Elkan, 2002; Niculescu-Mizil & Caruana, 2005; Guo et al., 2017; Gal & Ghahramani, 2016; Lakshminarayanan et al., 2017; Hendrycks & Gimpel, 2017; Jiang et al., 2018; Corbière et al., 2019). For example, Gal & Ghahramani (2016) and Lakshminarayanan et al. (2017) estimate the uncertainty of deep neural networks via Bayesian methods so as to return a distribution over the predictions. The work in Hendrycks & Gimpel (2017) uses the softmax probability of the network to detect if a test data point is misclassified or out-of-distribution. A new trust score to understand whether a classifier's prediction for a test data point can be trusted or not is proposed in Jiang et al. (2018). Most recently, Corbière et al. (2019) assess the confidence of a model by proposing a new target criterion for model confidence based on the True Class Probability. These methods rely on the training data and/or internal architecture of the network to generate the model confidence score. Our method, in contrast, assumes an already trained and black-box model; our goal is to estimate the performance of the model on various metrics on a new test dataset.

Active Learning. These methods aim to train a machine learning model in a data-efficient way by selecting the most informative data points for which labels should be acquired (Settles, 2010; Blundell et al., 2015; Gal et al., 2017; Kirsch et al., 2019). These AL methods aim to train a model to predict the labels of new data points accurately whilst our method aims to train a model to estimate a specified metric set accurately. We have shown experimentally that for the active testing framework, our proposed method outperforms existing AL methods.

B ALGORITHM DETAILS

B.1 BACKGROUND

Bayesian Neural Networks Bayesian neural networks (BNNs) are special neural networks that maintain a distribution over its parameters (MacKay, 1992; Neal, 1995). Specifically, given the training data $\mathcal{D}_{tr} = \{x_i, y_i\}_{i=1}^{N}$, a BNN can provide the posterior distribution $p^*(\omega|\mathcal{D}_{tr})$ with ω being the neural network weights. In practice, performing exact inference to obtain $p^*(\omega|\mathcal{D}_{tr})$ is generally intractable, hence we use a variational approximation technique to approximate this posterior. In particular, we employ the MC-dropout method (Gal & Ghahramani, 2016) as it is known to be both scalable and theoretically guaranteed in terms of inferring the true model posterior distribution $p^*(\omega|\mathcal{D}_{tr})$. That is, the MC-dropout method is equivalent to performing approximate variational inference to find a distribution in a tractable family that minimizes the Kullback-Leibler divergence to the true model posterior.

Bayesian Active Learning by Disagreement (BALD) BALD is a sampling methodology in Bayesian active learning, which samples the data point that maximizes the mutual information between the model prediction and the model posterior (Houlsby et al., 2011; Gal et al., 2017). To be more specific, let us denote $\mathcal{D}_l^t = \{\mathcal{X}_l^t, \mathcal{Y}_{\mathcal{X}_l^t}\}$ as the labeled set obtained after iteration t and $\mathcal{B}_\omega^t$ as the BNN trained on $\mathcal{D}_l^t$ (with parameter ω), then the acquisition function of BALD is defined as,

$$\mathbb{I}[y; \omega|x, \mathcal{D}_l^t] = \mathbb{H}[y|x, \mathcal{D}_l^t] - \mathbb{E}_{\omega \sim p(\omega|\mathcal{D}_l^t)}[\mathbb{H}[y|x, \omega]], \tag{7}$$

where y is predicted label for the data point x. The data point that maximizes BALD is the data point for which the model has many possible predictions, i.e. the posterior draws have disagreement.

B.2 DATA AUGMENTATION STRATEGY

At iteration t, given the labeled set $\mathcal{D}_l^t$ and the model-under-test outputs $\mathcal{A}_\mathcal{X}$, the goal is to train a BNN $\mathcal{B}_\omega$ such that the corresponding metric estimations are most accurate. Training BNN using solely the labeled set $\mathcal{D}_l^t$ might not result in accurate enough metric estimations. Thus, to improve the metric estimation accuracy, we propose to incorporate the information from the model-under-test outputs $\mathcal{A}_\mathcal{X}$ into the BNN training process. In particular, using the labeled set $\mathcal{D}_l^t$, we also train a binary classifier $\mathcal{C}_\eta$ that aims to predict the data points in the test dataset for which the model-under-test agrees with the ground-truth. Using the predictions by the classifier $\mathcal{C}_\eta$, we then construct an *augmented labeled set* $\mathcal{S}_l^t = \{\mathcal{X}_S^t, \mathcal{Y}_{\mathcal{X}_S^t}\}$ where $\mathcal{X}_S^t$ are all the data points in the test dataset $\mathcal{X}$ that $\mathcal{C}_\eta$ identifies the model-under-test predictions are accurate, and $\mathcal{Y}_{\mathcal{X}_S^t}$ are the corresponding model-under-test outputs of $\mathcal{X}_S^t$. The BNN is then trained using both the labeled set $\mathcal{D}_l^t$ and the augmented labeled set $\mathcal{S}_l^t$.

To train the binary classifier $\mathcal{C}_\eta$, we first split the labeled set $\mathcal{D}_l^t$ into two parts: training and validation, and then train $\mathcal{C}_\eta$ on the training part whilst tuning the softmax probability threshold using the validation part so that $\mathcal{C}_\eta$ achieves *the highest precision* on the validation part. This is because we want $\mathcal{C}_\eta$ to choose a data point only when it is most certain that the ground-truth and the model-under-test output of that data point is the same. Besides, as the precision of $\mathcal{C}_\eta$ is rarely 100%, thus, after obtaining the set of data points provided by $\mathcal{C}_\eta$, we only take N_s data points from this set with the highest softmax probability. The number N_s is computed by multiplying the precision of $\mathcal{C}_\eta$ on the validation part with the cardinality of the original predicted set. For example, if the precision of the classifier $\mathcal{C}_\eta$ is 50% on the validation part and the original predicted set consisting of 100 data points, the final augmented labeled set $\mathcal{S}_l$ only consists of 50 data points with the highest softmax probability. Finally, the binary classification problem can be imbalanced, particularly when

the model-under-test is very accurate or very bad. Hence, when training $\mathcal{C}_\eta$, we employ the over-sampling technique (for the minority class) to ensure the training data of the binary classification problem to be balanced, i.e. the cardinalities of the majority and minority classes are equal.

Remark B.1. With this training methodology, the more accurate the model-under-test, the more accurate the BNN $\mathcal{B}_\omega$. In case when the model-under-test is bad, the augmented labeled set $\mathcal{S}_l^t$ does not consist of many elements, thus, the BNN accuracy does not improve much compared to when training solely using the labeled set $\mathcal{D}_l^t$. However, in this case, the BNN does not need to have high accuracy in order to accurately estimate the metrics. Specifically, for any data point for which the model-under-test disagrees with the ground truth, the BNN does not need to accurately predict its label. That is, even when the BNN predicts other labels, except the model-under-test output label, the metric estimation is still accurate (more detailed examples in App. D.1).

Remark B.2. For simplicity, we suggest to set the architecture of the binary classifier and BNN to be same. For example, if the BNN is a 2-layer MLP, then the binary classifier is also a 2-layer MLP.

Remark B.3. In our implementation, to ensure the augmented labeled set $\mathcal{S}_l^t$ is accurate, we compute N_s to be the square of C_η precision multiplying with the cardinality of the original predicted set. For example, if the precision of the classifier $\mathcal{C}_\eta$ is 50% on the validation part and the original predicted set consisting of 100 data points, the final augmented labeled set $\mathcal{S}_l$ only consists of 25 data points with the highest softmax probability.

B.3   Details of the Proposed Acquisition Function

Our proposed acquisition function is defined as the mutual information (Houlsby et al., 2011; Gal et al., 2017) between $\tilde{Q}_k(x)$ and the BNN parameters ω, i.e.,

$$\mathbb{I}[\tilde{Q}_k(x); \omega | x, \mathcal{D}_l^t] = \mathbb{H}[\tilde{Q}_k(x)|x, \mathcal{D}_l^t] - \mathbb{E}_{\omega \sim p(\omega | \mathcal{D}_l^t)}[\mathbb{H}[\tilde{Q}_k(x)|x, \omega]]. \tag{8}$$

This mutual information can be approximated using the MC-dropout variational distribution $q_\theta(\omega | \mathcal{D}_l^t)$. Next, we show how to approximate each tearm on the right hand side of Eq. (8).

Computing $\mathbb{H}[\tilde{Q}_k(x)|x, \mathcal{D}_l^t]$ For each data point x, $\hat{y}_x$ is a discrete random variable with C distinct values, so $\tilde{Q}_k(x)$ is also a discrete random variable with at most C distinct values. Therefore, $\mathbb{H}[\tilde{Q}_k(x)|x, \mathcal{D}_l^t]$ can be computed as,

$$\mathbb{H}[\tilde{Q}_k(x)|x, \mathcal{D}_l^t] = -\sum\nolimits_{q \in \mathcal{Q}} p(\tilde{Q}_k(x) = q|x, \mathcal{D}_l^t) \log p(\tilde{Q}_k(x) = q|x, \mathcal{D}_l^t), \tag{9}$$

where $\mathcal{Q}$ consists of all the possible values of $\tilde{Q}_k(x)$ when $\hat{y}_x \in \{1, ..., C\}$. By using the union bound, $\mathbb{H}[\tilde{Q}_k(x)|x, \mathcal{D}_l^t]$ can then be expressed as,

$$\mathbb{H}[\tilde{Q}_k(x)|x, \mathcal{D}_l^t] = -\sum\nolimits_{q \in \mathcal{Q}} \left(\sum\nolimits_{h \in \tilde{Q}_k^{-1}(q)} p(\hat{y}_x = h|x, \mathcal{D}_l^t) \right) \log \left(\sum\nolimits_{h \in \tilde{Q}_k^{-1}(q)} p(\hat{y}_x = h|x, \mathcal{D}_l^t) \right),$$

where $\tilde{Q}_k^{-1}(q)$ is the inverse function that maps the value of $\tilde{Q}_k(x)$ to $\hat{y}_x$. Given M stochastic forward passes $\{\hat{\omega}_j\}$ from the MC-dropout posterior distribution $q_\theta(\omega | \mathcal{D}_l^t)$, $\mathbb{H}[\tilde{Q}_k(x)|x, \mathcal{D}_l^t]$ can finally be approximated as,

$$\begin{aligned} \mathbb{H}[\tilde{Q}_k(x)|x, \mathcal{D}_l^t] \approx -\sum\nolimits_{q \in \mathcal{Q}} \Big(&\Big(\sum\nolimits_{h \in \tilde{Q}_k^{-1}(q)} (\sum\nolimits_{j=1}^M p(\hat{y}_x = h|x, \hat{\omega}_j))/M \Big) \\ &\times \log \Big(\sum\nolimits_{h \in \tilde{Q}_k^{-1}(q)} (\sum\nolimits_{j=1}^M p(\hat{y}_x = h|x, \hat{\omega}_j))/M \Big) \Big), \end{aligned} \tag{10}$$

where $\tilde{Q}_k(x)$ can be approximated as $\tilde{Q}_k(x) \approx \mathbb{E}_{\omega \sim q_\theta(\omega | \mathcal{D}_l^t)} [Q_k(\mathcal{A}_\mathcal{X}, [\mathcal{Y}_{\mathcal{X}_l^t}, \hat{\mathcal{Y}}_{\mathcal{X}_{ul}^t \setminus x, \omega}, \hat{y}_x])] \approx (\sum_{j=1}^M Q_k(\mathcal{Y}_\mathcal{A}, [\mathcal{Y}_l^t, \hat{\mathcal{Y}}_{\mathcal{X}_{ul}^t \setminus x, \hat{\omega}_j}, \hat{y}_x]))/M$, with $\hat{\mathcal{Y}}_{\mathcal{X}_{ult}^t \setminus x, \hat{\omega}_j}$ denoting the predicted labels for $\mathcal{X}_{ul}^t \setminus x$ given the parameter $\hat{\omega}_j$.

Computing $\mathbb{E}_{\omega \sim p(\omega | \mathcal{D}_l^t)}[\mathbb{H}[\tilde{Q}_k(x)|x, \omega]]$ Similar as in the above paragraph, given M stochastic forward passes $\{\hat{\omega}_j\}_{j=1}^M$ from the MC-dropout variational distribution $q_\theta(\omega | \mathcal{D}_l^t)$,

$\mathbb{E}_{\omega \sim p(\omega|\mathcal{D}_l^t)}[\mathbb{H}[\tilde{Q}_k(x)|x,\omega]]$ can be approximated as,

$$
\begin{aligned}
\mathbb{E}_{\omega \sim p(\omega|\mathcal{D}_l^t)}[\mathbb{H}[\tilde{Q}_k(x)|x,\omega]] &\approx \frac{1}{M}\sum\nolimits_{j=1}^{M}\mathbb{H}[\tilde{Q}_k(x)|x,\hat{\omega}_j] \\
&\approx -\frac{1}{M}\sum\nolimits_{j=1}^{M}\left(\sum\nolimits_{q\in\mathcal{Q}}p(\tilde{Q}_k(x)=q|x,\hat{\omega}_j)\log p(\tilde{Q}_k(x)=q|x,\hat{\omega}_j)\right) \\
&\approx -\frac{1}{M}\sum\nolimits_{j=1}^{M}\left(\sum\nolimits_{q\in\mathcal{Q}}\left(\sum\nolimits_{h\in\tilde{Q}_k^{-1}(q)}p(\hat{y}_x=h|x,\hat{\omega}_j)\right)\right. \\
&\qquad\qquad\qquad\left. \times \log\left(\sum\nolimits_{h\in\tilde{Q}_k^{-1}(q)}p(\hat{y}_x=h|x,\hat{\omega}_j)\right)\right),
\end{aligned}
$$

(11)

where $\mathcal{Q}$ consists of all the possible values of $\tilde{Q}_k(x)$ when $\hat{y}_x \in \{1,...,C\}$, $\tilde{Q}_k^{-1}(q)$ is the inverse function that maps the value of $\tilde{Q}_k(x)$ to $\hat{y}_x$, and $\tilde{Q}_k(x)$ can be approximated as $\tilde{Q}_k(x) \approx \mathbb{E}_{\omega \sim q_\theta(\omega|\mathcal{D}_l^t)}[Q_k(\mathcal{A}_\mathcal{X},[\mathcal{Y}_{\mathcal{X}_l^t},\hat{\mathcal{Y}}_{\mathcal{X}_{ul}^t\setminus x,\omega},\hat{y}_x])] \approx (\sum_{j=1}^{M}Q_k(\mathcal{Y}_\mathcal{A},[\mathcal{Y}_l^t,\hat{\mathcal{Y}}_{\mathcal{X}_{ul}^t\setminus x,\hat{\omega}_j},\hat{y}_x]))/M$, with $\hat{\mathcal{Y}}_{\mathcal{X}_{ult}^t\setminus x,\hat{\omega}_j}$ denoting the predicted labels for $\mathcal{X}_{ul}^t\setminus x$ given the parameter $\hat{\omega}_j$.

C   EXPERIMENT DETAILS

C.1   MNIST DATASET

Models-under-test   The three models-under-test we use to evaluate our proposed method are:

(i) *M-MNIST*: A CNN trained on the train MNIST dataset (LeCun et al., 2010). We implement the model in tensorflow but use the official network architecture published on keras.com[1] as there is no official source code on tensorflow.org. The accuracy of this model on the test MNIST dataset is 99.06%;

(ii) *M-MNIST-ES*: A CNN trained on the train MNIST dataset. The model architecture is same as M-MNIST, but we use early stopping to decrease the model performance, i.e. we set smaller epochs and higher batch size. The accuracy of this model on the test MNIST dataset is 70.67%;

(iii) *M-FashionMNIST*: A CNN trained on the train FashionMNIST dataset (i.e. a completely different dataset compared to MNIST) (Xiao et al., 2017). The model is implemented using the official code published on tensorflow.org.[2] The accuracy of this model on the test MNIST dataset is 12.39%.

Metric sets   The metric sets we use to evaluate our proposed method are:

(i) Three sets of metrics consisting of only one metric in each set: precision of class 2, recall of class 2, and overall accuracy. These are common metrics used to evaluate performance of classifiers, and these metrics cover both per-class metrics and overall metrics.

(ii) One set of metrics consisting of 21 metrics (accuracy, precision and recall of each class).

The BNN and the binary classifier architecture   The BNN and the binary classifier have the same architecture. The architecture is an MLP with two layers, 256 neurons/layer, and with dropout applied in each layer. The number of MC-dropout samples are 50. The initial labeled set $\mathcal{D}_l^0$ has 100 data points randomly sampled from the test dataset. We tuned 3 hyper-parameters: learning rate, epochs and dropout rate using Bayesian optimization (Snoek et al., 2012). We also reinitialize the BNN after each iteration as in (Gal et al., 2017).

---

[1] https://keras.io/examples/mnist_cnn/
[2] https://www.tensorflow.org/tutorials/keras/classification

D ADDITIONAL EXPERIMENT RESULTS

D.1 THE QUALITY OF THE BNNS

In Fig. 2, we plot the prediction accuracies of the BNNs in **BALD** and **ALT-MAS** for all combinations of models-under-test and metric sets on the MNIST dataset. Below, we evaluate how complex the BNN needs to be for the metric estimation to be accurate and how effective our data augmentation strategy and our proposed sampling methodology are.

Theoretically, we can see that for the BNN to be useful, it does not need to have high classification accuracy; it only needs to accurately predict the data points that contribute to the metric estimation. Being incorrect at other data points does not affect the predicted metric values. See example:

True label	Model-under-test prediction	Surrogate model prediction
0	5	Any prediction except 5 (e.g. $1, 2, 3, ...$) will give accurate metric estimation
5	5	Correct prediction matters for some metrics (e.g. accuracy, precision/recall of class 5), but not for other metrics (e.g. precision/recall of other classes)

In summary, the metric estimation accuracy depends on the model-under-test and the metric set. This is clearly demonstrated by the performance of **BALD**. To be specific, the BNNs trained with **BALD** have accuracies ranging from $70 - 90\%$, but for the models-under-test *M-FashionMNIST* and *M-MNIST-ES* (average & bad models), the metric estimation accuracies range from $90 - 100\%$ - which are much higher than the BNNs' accuracies.

For our proposed method **ALT-MAS**, with the models-under-test *M-FashionMNIST*, *M-MNIST-ES*, the behaviours are similar to those of **BALD**. That is, the metric estimation accuracies are always higher than the BNNs accuracies, especially for per-class metrics. It is worth noting that, for the per-class metrics, even though the BNNs accuracies by **ALT-MAS** are much lower than the BNNs by **BALD**, but the metric estimations by **ALT-MAS** are much higher than by **BALD**. This asserts the motivation of our sampling approach, that is, the BNN only needs to accurately predict the data points that contribute to the metric estimation. On the other hand, with the good model-under-test *M-MNIST*, due to our data augmentation training strategy, the BNN accuracies by **ALT-MAS** are much higher than those of **BALD**, and thus, the metric estimations by **ALT-MAS** are also more accurate than those by **BALD**.
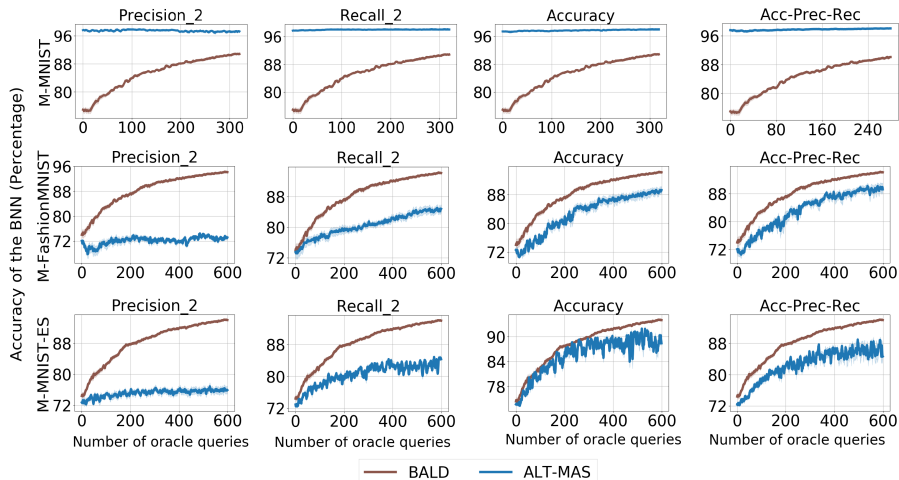

Figure 2: The accuracy of the BNN, for each combination of model-under-test (*M-MNIST*, *M-FashionMNIST*, & *M-MNIST-ES*) and metric set. Plotting mean and standard error over 3 repetitions (Best seen in color).