

MPP: Model Performance Predictor

Sindhu Ghanta, Sriram Subramanian, Lior Khermosh, Harshil Shah, Yakov Goldberg, Swaminathan Sundararaman, Drew Roselli, and Nisha Talagala, *ParallelM*

https://www.usenix.org/conference/opml19/presentation/ghanta

This paper is included in the Proceedings of the 2019 USENIX Conference on Operational Machine Learning (OpML '19).

May 20, 2019 • Santa Clara, CA, USA

ISBN 978-1-939133-00-7

Open access to the Proceedings of the 2019 USENIX Conference on Operational Machine Learning is sponsored by USENIX.

MPP: Model Performance Predictor

Sindhu Ghanta
ParallelM

Sriram Subramanian
ParallelM

Lior Khermosh
ParallelM

Harshil Shah
ParallelM

Yakov Goldberg
ParallelM

Swaminathan Sundararaman
ParallelM

Drew Roselli
ParallelM

Nisha Talagala
ParallelM

Abstract

Operations is a key challenge in the domain of machine learning pipeline deployments involving monitoring and management of real-time prediction quality. Typically, metrics like accuracy, RMSE etc., are used to track the performance of models in deployment. However, these metrics cannot be calculated in production due to the absence of labels. We propose using an ML algorithm, *Model Performance Predictor* (MPP), to track the performance of the models in deployment. We argue that an ensemble of such metrics can be used to create a score representing the prediction quality in production. This in turn facilitates formulation and customization of ML alerts, that can be escalated by an operations team to the data science team. Such a score automates monitoring and enables ML deployments at scale.

1 Introduction

Using machine learning models to extract insights from massive datasets is a widespread industry goal [10]. The training phase typically generates several models and the model with the best predictive performance is deployed to production. However, a model's performance in production depends on both the particular data it receives and the datasets originally used to train the model. Models perform optimally on different data distributions and vary in their capacities for generalization. Production datasets often vary with external factors [8, 14]. Whether rapid or gradual, these variations can require models to be updated or rolled back to maintain good predictive performance. Massive scale in production systems prohibits manual intervention or monitoring of such events, requiring in turn automated methods to detect, diagnose, and improve the quality of predictive performance. However, typical production scenarios do not have real-time labels, so popular metrics that compare predictions with labels cannot be used to assess real-time health.

We present a technique to track the predictive performance of the deployed models called: *Model Performance Predictor* (MPP). It tracks the predictive performance metric of the model. For (a) classification and (b) regression, we present an example that targets (a) accuracy and (b) RMSE respectively as the metric to track.

Detecting the applicability of an activity model to a different domain using another model was proposed in [15] using algorithm-specific information. Similar to our approach, an error dataset is used to train another model, but it is limited to a specific algorithm (random forest) and a unique domain. With a similar goal of detecting the confidence in predictions made by a machine learning algorithm, [3] proposed hedging the predictions using conformal predictors. A hold out set (in contrast to error set) is used to obtain a bound on the error probability. On the other hand, we present an approach that models the errors by using them as the labels. On similar lines, [12] presented a metric that tracks the divergence in data patterns between training and inference. We argue that an ensemble of such approaches can be customized to serve as a score, based on which alerts can be raised.

2 Model Performance Predictor

The goal of Model Performance Predictor (MPP) algorithm is to predict the predictive performance of the deployed algorithm on the test data. This algorithm is trained on the error dataset which consists of prediction errors made by the primary algorithm. In the training phase, the data is divided into training and validation datasets (apart from the test set). The training dataset is used to train the primary algorithm that will be deployed in production. Predictions made by this algorithm on the validation dataset generate the errors that are used as labels to train the MPP algorithm.

Figure 1 describes the structure of this framework. Labels of this error dataset are the errors in predictions made by the primary algorithm, and features could be a range of things depending on the application. They could simply be the primary algorithm features themselves, predictions from the primary algorithm, probability measures from the primary predictions or some algorithm-specific metrics, such as the number of

trees or variation in output from different trees in a Random Forest. Both primary and MPP algorithms make predictions on the test dataset. The primary algorithm focuses on the classification/regression task, while the MPP algorithm focuses on predicting the performance of the primary algorithm. We present MPP as a binary classification algorithm that predicts whether a prediction is correct (1) or incorrect (0).

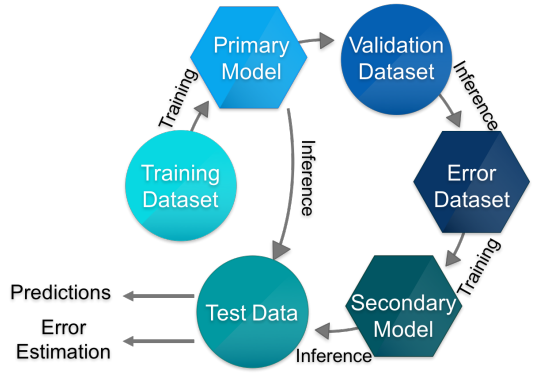


Figure 1: MPP algorithm flow

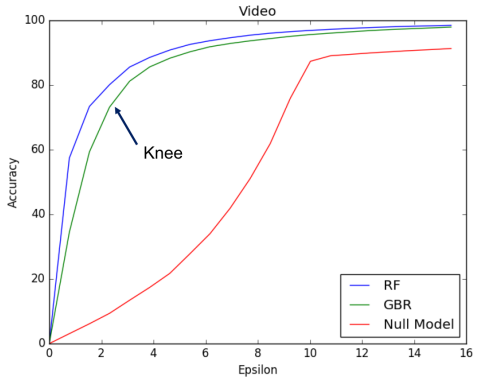


Figure 2: REC curve for the Video dataset . RF represents Random Forest; GBR represents Gradient Boosted Regression Tree

For regression problems, in order to calculate error analogously to how it is done in classification, we use a threshold (ε) on the absolute error of primary predictions to be within tolerable limits. For example, as long as error is within ±ε of the true value, it is considered an acceptable prediction (1). When the prediction of an algorithm is outside these bounds, it's considered an unacceptable (or incorrect) prediction (0). However, this threshold value is application specific and there is a need to detect a default value. To provide a default value, we use the null model concept introduced by [5]. Every regression problem has a null model and hence an REC curve associated with it. We detect the knee of this curve using the first convex dip in its double differential and choose the corresponding ε to be the default threshold value. An REC plot for the video dataset [11] is shown in Figure 2. We calculate this default threshold for all the regression experiments reported in Section 3.

Dataset	Primary Algorithm Error	MPP predicted accuracy	Absolute difference
Samsung [2]	0.92	0.92	0.00
Yelp [1]	0.95	0.95	0.00
Census [13]	0.78	0.63	0.15
Forest [6]	0.65	0.64	0.01
Letter [17]	0.71	0.6	0.11

Table 1: MPP's performance on classification datasets. Ideally, the primary algorithm accuracy and MPP's prediction should match.

Dataset	Primary Algorithm Error	MPP predicted accuracy	Absolute difference
Facebook [16]	0.56	0.56	0.00
Songs [4]	0.58	0.61	0.03
Blog [7]	0.73	0.71	0.02
Turbine [9]	0.51	0.85	0.34
Video [11]	0.59	0.72	0.13

Table 2: MPP's performance on regression datasets with default epsilon value. Ideally, the primary algorithm accuracy (generated by thresholding with default epsilon) and the MPP's prediction should match

3 Illustration

We illustrate the performance of this algorithm using 5 classification and regression datasets, listed in Table 1 and 2 respectively. Features used by the MPP algorithm for the purpose of these experiments are same as the features used by the primary algorithm. Ideally, the score presented by *MPP* algorithm should match the predictive performance of the primary algorithm. It can be seen from the tables that the MPP algorithm is able to track the performance of primary algorithm in most of the datasets.

4 Conclusion

We presented an approach *MPP* to track the predictive performance of a ML model in deployment. Such a score helps operations teams to create automated ML alerts and data scientists to get insights about the efficacy of deployed models in production. This helps both, the operations teams to monitor and manage the deployed ml model potentially preventing catastrophic predictions and the data scientists to get the information they need for further analysis of the production system.

References

[1] Yelp Dataset. https://www.yelp.com/dataset_challenge/, 2013.

[2] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge L. Reyes-Ortiz. A public domain dataset for human activity recognition using smartphones. *21th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN*, pages 1–15, 2013.

[3] Vineeth Balasubramanian, Shen-Shyang Ho, and Vladimir Vovk. *Conformal Prediction for Reliable Machine Learning: Theory, Adaptations and Applications*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st edition, 2014.

[4] T. Bertin-Mahieux. UCI machine learning repository. http://archive.ics.uci.edu/ml/datasets/YearPredictionMSD, 2011.

[5] Jinbo Bi and Kristin P. Bennett. Regression error characteristic curves. In *Proceedings of the 20th International Conference on Machine Learning*, pages 43–50, 2003.

[6] Jock A. Blackard, Denis J. Dean, and Charles W. Anderson. UCI machine learning repository. https://archive.ics.uci.edu/ml/datasets/Covertype, 1998.

[7] K. Buza. Feedback prediction for blogs. in data analysis, machine learning and knowledge discovery. *Springer International Publishing*, pages 145–152, 2014.

[8] Piotr Cal and Michał Woźniak. Drift detection and model selection algorithms: Concept and experimental evaluation. In Emilio Corchado, Václav Snášel, Ajith Abraham, Michał Woźniak, Manuel Graña, and Sung-Bae Cho, editors, *Hybrid Artificial Intelligent Systems*, pages 558–568, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[9] Andrea Coraddu, Luca Oneto, Alessandro Ghio, Stefano Savio, Davide Anguita, and Massimo Figari. Machine learning approaches for improving condition?based maintenance of naval propulsion plants. *Journal of Engineering for the Maritime Environment*, –(–):–, 2014.

[10] A. Daecher. Internet of things: From sensing to doing. Wall Street Journal, 2016.

[11] T. Deneke, H. Haile, S. Lafond, and J. Lilius. Video transcoding time prediction for proactive load balancing. In *Multimedia and Expo (ICME), 2014 IEEE International Conference on*, pages 1–6, July 2014.

[12] Sindhu Ghanta, Sriram Subramanian, Lior Khermosh, Swaminathan Sundararaman, Harshil Shah, Yakov Goldberg, Drew Roselli, and Nisha Talagala. Ml health: Fitness tracking for production models. *arXiv:1902.02808*, 2019.

[13] Ronny Kohavi and Barry Becker. UCI machine learning repository. "https://archive.ics.uci.edu/ml/datasets/Census+Income, 1996.

[14] Osama A. Mahdi, Eric Pardede, and Jinli Cao. Combination of information entropy and ensemble classification for detecting concept drift in data stream. In *Proceedings of the Australasian Computer Science Week Multiconference*, ACSW '18, pages 13:1–13:5, New York, NY, USA, 2018. ACM.

[15] Robert P. Sheridan. Using random forest to model the domain applicability of another random forest model. *Journal of Chemical Information and Modeling*, 53(11):2837–2850, 2013.

[16] Kamaljot Singh. Facebook comment volume prediction. *International Journal of Simulation- Systems, Science and Technology- IJSSST V16*, January 2016.

[17] David J. Slate. UCI machine learning repository. https://archive.ics.uci.edu/ml/datasets/Letter+Recognition, 1991.